

Федеральное государственное образовательное бюджетное учреждение
высшего образования
«Финансовый университет при Правительстве Российской Федерации»

На правах рукописи

Кораблев Юрий Александрович

ПРОГНОЗИРОВАНИЕ РЕДКИХ
ЭКОНОМИЧЕСКИХ СОБЫТИЙ С ПОМОЩЬЮ
ВОССТАНОВЛЕНИЯ МЕХАНИЗМОВ ИХ
ОБРАЗОВАНИЯ

5.2.2. Математические, статистические и инструментальные методы
в экономике

ДИССЕРТАЦИЯ
на соискание ученой степени
доктора экономических наук

Научный консультант

Бывшев Виктор Алексеевич,
доктор технических наук, профессор

Судаков Владимир Анатольевич,
доктор технических наук, доцент

Москва - 2026

Оглавление

Введение.....	4
Глава 1 Анализ и адаптация существующих методов прогнозирования редких экономических событий.....	25
1.1 Проблематика прогнозирования редких экономических событий и терминология.....	25
1.2 Методы классификации в прогнозировании событий	40
1.3 Границы применимости и недостатки методов классификации при прогнозировании событий	67
1.4 Методы регрессии в прогнозировании событий	73
1.5 Границы применимости и недостатки регрессионных методов при прогнозировании событий	80
1.6 Методы на основе потоков событий в прогнозировании событий.....	83
1.7 Границы применимости и недостатки метода, основанного на потоках событий.....	91
1.8 Другие задачи и методы, связанные с прогнозированием событий.....	92
Глава 2 Модели механизмов образования событий и математические методы восстановления их параметров	104
2.1 Общая схема предлагаемого подхода прогнозирования событий.....	104
2.2 Выбор и описание модели механизма образования событий	109
2.3 Восстановление параметров механизма образования событий регрессионными методами.....	119
2.4 Восстановление параметров механизма образования событий методами машинного обучения.....	127
2.5 Восстановление параметров механизма образования событий методами сплайновой коллокации	133
2.6 Монотонная/положительная сплайновая коллокация	148
2.7 Экстраполяция параметров и получение прогноза будущего события ..	169
2.8 Определение гиперпараметров.....	175

2.9 Проверка адекватности модели механизма образования событий.....	178
Глава 3 Восстановление алгоритмических моделей механизмов образования событий с помощью численных методов.....	191
3.1 Особенности представления модели механизма образования событий в алгоритмической форме.....	191
3.2 Определение параметров алгоритмических моделей механизмов образования событий.....	198
3.3 Экстраполяция восстановленных параметров.....	206
3.4 Методы автоматического подбора структуры алгоритмических моделей механизмов образования событий.....	208
Глава 4 Разработка информационных технологий для прогнозирования экономических событий.....	217
4.1 Разработка библиотеки eventometrics для прогнозирования событий на языке Python.....	218
4.2 Создание компонента в Loginom для прогнозирования событий.....	247
Глава 5 Апробация и сравнение нового подхода с существующими методами.....	257
5.1 Апробация разработанных методик.....	257
5.2 Сравнение с методами классификации.....	261
5.3 Сравнение с методами регрессии.....	297
5.4 Сравнение с другими методами.....	310
5.5 Экономический эффект.....	323
Заключение.....	331
Список литературы.....	335
Приложение А Формальное определение механизма образования событий через аксиомы Колмогорова.....	356

Введение

Актуальность темы исследования. Прогнозирование событий даёт возможность к ним подготовиться, позволяет извлекать выгоду или уменьшать убытки, определяет успех или спасает от неудач. Прогноз событий востребован как в государственном управлении, так и в управлении частным бизнесом. Своевременный прогноз может позволить правительству заранее подготавливать меры поддержки, центральному банку – заранее корректировать монетарную политику. Умение прогнозировать события позволит бизнесу предвидеть изменения спроса, оптимизировать издержки на хранение запасов и логистику, избегать дефицита и затоваривания, оптимизировать работу своего персонала, управлять как подконтрольными, так и неподконтрольными клиентами, инвестировать в перспективные проекты и избегать рискованные инвестиции. Сложно переоценить способность предсказывать будущие события. В особенности это касается редких экономических событий.

Первое, с чем ассоциируются «редкие события», – это финансовые кризисы, дефолт того или иного банка, землетрясения, выигрыш в лотерее, прорывное научное открытие, внезапный скачок продаж или цен и так далее. То есть что-то экстраординарное по своей величине и значению. Однако будет ли неожиданное изменение продаж одной компании редким событием для другой компании, которая получает и без того огромные прибыли на регулярной основе? Будет ли скачок цен на товар или на акции редким событием, если существуют другие товары и акции, которые и без того имеют крайне высокие цены? Будет ли финансовый кризис в мире хоть чем-то значимым для изолированного сообщества людей? Что для одного может являться редким событием, может таковым не являться для другого. Не величина играет главную роль, она может быть большой или маленькой, положительной или отрицательной, а внезапность. В этом понимании «редкие события» обозначают моменты времени, когда нарушаются ожидания людей.

Это те моменты, когда мысленные или математические модели, которые разные люди используют для прогнозирования своего будущего дня, разбиваются об объективную реальность. Такие редкие события Нассим Николас Талеб называет «чёрными лебедями» [1]. Однако Талеб на протяжении всей своей знаменитой работы говорит о невозможности прогнозирования «чёрных лебедей», момента времени, когда ваша модель сломается.

Ожидания нарушаются лишь по нашему ошибочному представлению. Нужно лучше изучать происходящие в мире процессы и разбираться в причинах возникновения событий. Можно выразиться так: те «чёрные лебеди», которые привлекли внимание, заставляют строить модели процессов, в результате чего события перестают быть чёрными лебедями. Далее можно вообще уйти от понятия чёрных лебедей. Мы занимаемся прогнозированием тех событий, модели образования которых можно построить. Талеб называет их «серыми лебедями». Конечно же, наши модели, так же, как и любые другие модели, могут давать ошибочный прогноз, но это будет лишь свидетельствовать о неполном понимании реальности. Это несоответствие будет говорить о том, что надо скорректировать модель.

В этой работе под редкими событиями понимается не нарушение ожиданий. Редкие события относятся всё к тем же дискретным событиям, на которые по той или иной причине обращается внимание и которые могли бы вовсе не обладать смыслом редких или значимых событий. Но в результате моделирования и предсказания событий некоторые из них могут получиться такими, которые будут обладать тем самым уникальным свойством, которое позволило бы назвать такие события редкими или значимыми. Предсказание редких событий базируется на способности предсказывать обычные события. С этого момента можно вообще опустить само понятие «редкие», ведь, занимаясь предсказанием событий, в том числе будем получать и редкие. Вместо значимости под редкостью понимается «не частота», подчёркивая дискретный характер данных. Но даже с этим понятием не всё так просто и

однозначно, так как оно является качественным. Формальные определения будут даны в начале первой главы.

В области моделирования редких событий царит терминологическая неразбериха. Одинаковые понятия часто обозначаются разными терминами, а под одним и тем же названием могут скрываться принципиально разные концепции и решаемые задачи. Среди названий можно встретить такие, как аномалии, новизна, выбросы, инциденты. Среди задач, использующих термин «редкие события», встречаются такие, как задача обнаружения событий во временных рядах, классификация по наблюдаемым признакам, определение вероятности возникновения события в заданном периоде времени, определение вероятности выхода за границы допустимых условий и др. Исследование существующих методов показало, что большинство из них основано на выявлении статистических связей между наблюдаемыми признаками и интервалами времени до следующего события или самим фактом появления события на интервале времени. Но когда в наблюдаемых признаках не содержится информация, способная объяснить интервалы времени до следующего события, или таких признаков вовсе нет, то такие статистические методы могут работать плохо. Другие методы ограничиваются лишь определением закона распределения для интервалов времени между событиями, что неспособно объяснить, почему получающиеся интервалы оказываются то короче, то длиннее. Модели, основанные на применении машинного обучения, в особенности нейросетевые модели или модели, использующие градиентный бустинг, применяющиеся в прогнозировании событий, имеют характер чёрного ящика. Это создаёт проблемы, связанные с объяснимостью и пониманием устройства самих этих моделей, что делает результат прогнозирования неубедительным и ненадёжным.

Примечание – В последнее время развиваются идеи создания так называемого объяснимого искусственного интеллекта (eXplainable AI, XAI) [2, 3], т.е. набора методов, чтобы понять/проанализировать результаты решения, полученные с помощью методов «искусственного интеллекта». Это позволяет частично прощупать и проинтерпретировать полученный в результате обучения «чёрный ящик». Другой подход может основываться на конструировании «физически-информированных нейронных сетей» (Physics-informed neural networks, PINNs), в которых отклонения от законов природы, например законов сохранения, заложены в функцию потерь [4]. Результаты в этой области ещё далеки от идеала.

Анализ множества работ показал, что задача прогнозирования редких событий изучена недостаточно подробно. Отсутствует комплексная методология прогнозирования редких экономических событий. Существующим методам необходим большой объём обучающей выборки, что недоступно при анализе редких событий.

Отсутствие комплексной методологии прогнозирования редких экономических событий составляет научную проблему исследования. Задача прогнозирования событий является крайне важной. Знание момента появления событий позволит грамотно к нему подготовиться, извлечь определённую выгоду или уменьшить возможные убытки. Сложно переоценить способность предсказывать будущие события. Основываясь на вышесказанном, можно констатировать, что разработка новой методологии прогнозирования редких экономических событий является важной и актуальной темой исследования.

Как показано в работе, задача прогнозирования редких экономических событий может решаться в рамках новой методологии, в которой предлагается по небольшой зашумлённой выборке событий восстанавливать параметры алгоритмических моделей механизмов их образования. Такой подход по точности является конкурентным передовым существующим методам машинного обучения, зачастую превосходя их. Разрабатываемая методология закладывает основы новому научному направлению, которое назовем «эвентометрика».

В первой главе рассматриваются существующие статистические методы, описывается, как следует их правильно применять для задач прогнозирования событий, разбираются основные ошибки при применении этих методов. Во второй главе разрабатывается методология прогнозирования событий, которая основывается на механистическом представлении причин возникновения событий. Формулируется новый подход, который предлагает строить алгоритмические модели образования событий. Рассматриваются и разрабатываются способы восстановления параметров таких алгоритмических моделей математическими и статистическими методами, способы

экстраполяции таких параметров, способ определения гиперпараметров, описывается способ получения самого прогноза событий, разрабатываются способы проверки адекватности построенных алгоритмических моделей прогнозирования событий. В третьей главе описывается способ восстановления параметров механизма образования событий численными методами и описываются способы автоматического подбора самих алгоритмических моделей. В четвёртой главе описывается разработка собственной библиотеки `eventometrics` на языке Python, а также компонента в системе бизнес-аналитики `Loginom` для прогнозирования экономических событий. В пятой главе представлены результаты апробации описанного подхода, а также происходит его сравнение с существующими методами по точности получаемого прогноза. Также в пятой главе описывается экономический эффект от использования нового подхода для прогнозирования событий.

Степень разработанности темы исследования. Вопросами математического моделирования экономических процессов занимались Кантарович Л.В., Юдин Д.Б., Данциг Г., Беллман Р., Леонтьев В.В., Нейман Дж., Morgenstern О., Слуцкий Е.Е., Фельдман Г.А., Немчинов В.С., Поспелов И.Г., Моисеев Н.Н., Иванилов Ю.П., Макаров В.Л., Бахтизин А.Р. и др.

Вероятностное моделирование редких событий опирается на фундамент, который заложили Байес Т., Бернулли Д., Парето В., Пирсон К., Пуассон С.Д., Фишер Р., Госсет В., Марков А.А., Колмогоров А.Н., Ширяев А.Н. и др. Однако чисто вероятностное моделирование неспособно объяснить отклонения в большую или меньшую сторону от своего ожидаемого значения, что делает прогноз событий менее точным и обоснованным. Прогнозирование интервалов времени до следующего события на основе наблюдаемых признаков возможно осуществить проверенными регрессионными методами, основы которых заложили Лежандр А.М., Гаусс К.Ф., Адраин Р., Нейман Дж., Эйткен А.С., Фишер И., Фриш Р.,

Тинберген Я., Шумпетер Й., Андерсон О., Айвазян С.А., Бывшев В.А., Бабешко Л.О. и многие другие.

Появление событий часто прогнозируют методами бинарной или множественной классификации, где по наблюдаемым сопутствующим признакам определяют класс того или иного наблюдения. Для этого используются хорошо известные методы, которые последнее время выделяют в группу методов машинного обучения, такие как логистическая и пробит-регрессия, метод ближайших соседей, метод опорных векторов, деревья решений, градиентный бустинг, нейронные сети и другие методы, основы которых были заложены в работах Берксона Дж., Блисса Ч., Вапника В.Н., Червоненкиса А., Босера Б., Бравермана Э., Cramer J.S., Cortes C., Fix E., Cover T., Hart P., Bentley J.L., Omohundro S.M., Singh S., Morgan, J.N., Sonquist, J.A., Belson W., Breiman, L., Friedman J., Olshen R., Stone C.J., Landwehr N., Hall M., Frank E., Kearns M., Valiant L.G., Schapire R.E., Freund Y., Rosenblatt F., Choe W., Ersoy O.K., McCullagh P., Nelder J., Hosmer D.W., Lemeshow S., Goodfellow I., Hastie T., Tibshirani R., Каширина И.Л., Демченко М.В., Судакова В.А., Соловьева В.И. и др. Редкие события зачастую связаны с проблемой несбалансированности выборки, решение которой представлено в работах Vannucci M., Colla V. и др. Методы классификации имеют очень широкое применение в работах отечественных и зарубежных исследователей: для определения устойчивости (дефолта/банкротств) компаний или банков – Гусятников П.В., Могилат А.Н., Туктарова П.А., Заиченко Е.М., Рыгин В.Е., Биджоян Д.С., Kumar K., Tan C., Jardin P., Johnsen T. и многих других (где дефолты/банкротства определяются как редкие события); в информационной безопасности для определения атак или эксплойтов – Лифанов К.А, Зайцев К.С.; в прогнозировании ненулевого спроса – Пивкин К.С.; при определении потенциальных покупателей – Заказчикова Н.А.; и во многих других областях экономики.

Очень часто редкие события рассматривают с точки зрения потоков событий или точечных процессов, развитие которых осуществлено в работах

Пуассона С.Д., Пальма К., Вейбулла В., Эрланга А., Феллера В., Кингмена Дж., Кокса Д., Башарина Г.П., Наумова В.А., Вентцель Е.С., Овчаров Л.А. и др. В экономике теория потоков событий находит очень широкое применение: при прогнозировании спроса и моделировании собственных объёмов страховых запасов – Истомина А.А., Ян Л.А., Замалетдинова Д.А., Syntetos A.A., Babai M.Z., Luo S., Shale E.A.; в сельском хозяйстве – Белякова А.Ю., Петрова С.А.; определении надёжности технических систем гражданской инфраструктуры – Постников И.В., Скопинцев В.А.; при оценке стоимости акций – Кожевников А.С.; при моделировании экологических катастроф – Борисов В.В.; при расчёте рисков негативных событий – Саченко Л.А. и многие другие. В теории экстремальных событий также оперируют термином «редкие события», хотя в ней не делается предсказание событий. События моделируются пуассоновскими потоками, а ущерб от событий – с помощью распределений с тяжёлыми хвостами. В этой области работали Embrechts P., Klüppelberg C., Mikosch T., Paul E. и др.

Прогнозирование событий перекликается с задачами создания систем раннего предупреждения, например кризисов, которые развивали зарубежные учёные Berg A., Pattillo C. и др. Современные системы раннего предупреждения, основанные на методах машинного обучения, рассматривали в своих работах Xi L., Yining W., Qianqian F., Qinyun L., Tian W. и др. Также редкие события встречаются в работах по теории случайного блуждания, таких как случайные процессы Леви и теории больших прыжков. Но в этой теории не предсказываются отдельные события, а определяются законы распределения для суммарного движения. В этом направлении работали Wiener N., Lévy P., Burioni R., Vezzani A., Holl M., Barkai E.

Отдельно для задачи прогнозирования ненулевого спроса (прерывистого спроса, *intermittent demand forecast*), в которой также используют понятие редких событий, существуют специфические статистические методы прогнозирования, позволяющие рассчитать объёмы

собственных страховых запасов. В этом направлении работали исследователи Croston J.F., Syntetos A.A., Boylan J.E., Leven E., Segerstedt A., Vinh D.Q., Efron B., Willemain T.R., Smart C.N., Hua Z.S., Kaya G.O., Pince C. и др.

К редким событиям иногда относят отказы технических устройств и информационных систем в теории надёжности систем, развитие которой связано с работами Гнеденко Б.В., Беляева Ю.К., Соловьева А.Д., Ушакова И.А., Кокса Д., Барлоу Р., Проскан Ф. и др.

Полученные в данной работе результаты опираются на методы непараметрической регрессии, такие как ядерные функции, теория приближения функций сплайнами, сплайновая коллокация, интегро-дифференциальные сплайны, методы решения некорректно поставленных задач, определение оптимального параметра сглаживания, основы которых были заложены в работах Nadaraya E.A., Watson G.S., Henderson D.J., Бубнова И.Г., Whittaker E.T., Schoenberg I.J., Curry H.B., Reinsch Ch.H., Ahlberg J.H., Nilson E.N., Walsh J.L., Boor C.D., Schumaker L.L., Kimeldorf G., Craven P., Wahba G., Ramsay J.O., Green P.J., Silverman B.W., Тихонова А.Н., Арсенина В.Я., Wright I.W., Morozov A., Hansen, P.C., Киреева В.И., Бирюковой Т.К., Самойловой Э.Н., Федоровой О.П., Zemlyanova A.Y., Machina A., Dagnino C. Обеспечение положительности восстанавливаемой функции основывается на работах Elfving T., Andersson L., Vlachkova K., Greiner H., Ramsay J.O., Nagahara M., Green P.J., Silverman B.W., Chan V., Tsui K.W., Wei Y., Смоляка С.А. Решение задач квадратичного программирования опирается на работы Лагранжа Ж.Л., Лемке К.Э., Katta G.M., Wright M.H., Писарука Н.Н.

Цель исследования – разработка методологии прогнозирования экономических событий, в том числе редких, с помощью восстановления механизмов их образования для повышения точности и обоснованности прогноза.

Задачи:

- Анализ и адаптация существующих методов для прогнозирования экономических событий (методы классификации, регрессионные методы, методы машинного обучения, методы теории случайных процессов и др.), анализ границ их применимости и недостатков.
- Разработка методологии прогнозирования экономических событий, основанной на составлении алгоритмических моделей и восстановлении параметров механизмов их образования.
- Анализ применимости существующих математических методов для восстановления параметров механизмов образования событий.
- Разработка метода восстановления параметров механизма образования событий, основанного на методах монотонной сплайновой коллокации.
- Разработка способов экстраполяции восстановленных параметров на будущее, чтобы механизм образования событий функционировал в условиях, соответствующих будущему периоду времени.
- Разработка метода подбора гиперпараметров для применяемых алгоритмов восстановления и экстраполяции параметров механизма образования событий.
- Разработка способов проверки адекватности построенных алгоритмических моделей прогнозирования событий.
- Разработка метода восстановления параметров механизмов образования событий, заданных в алгоритмической форме, на основе численной оптимизации.
- Разработка методов автоматического подбора самих алгоритмических моделей образования событий.
- Сравнение нового подхода прогнозирования событий с существующими методами на практических примерах.

– Реализация разработанных математических методов и алгоритмов с использованием современных информационных технологий на современных языках программирования.

Объект исследования – экономические события, в том числе редкие, и методы их прогнозирования.

Предмет исследования – механизмы образования экономических событий, в том числе редких, и методы восстановления их параметров.

Область исследования. Диссертация подготовлена в соответствии с п. 11. «Компьютерные методы и программы моделирования экономических процессов», п. 14. «Эконометрические и статистические методы анализа данных, формирования и тестирования гипотез в экономических исследованиях. Эконометрическое и экономико-статистическое моделирование» Паспорта научной специальности: 5.2.2. Математические, статистические и инструментальные методы в экономике (экономические науки).

Научная новизна исследования заключается в разработке новой методологии прогнозирования экономических событий, основанной на восстановлении механизмов их образования. В отличие от существующих статистических методов в новом подходе строятся алгоритмические модели механизмов образования событий. По выборке событий восстанавливаются параметры моделей. Разработаны аналитические средства восстановления параметров, основанные на монотонной сплайновой коллокации, а также средства, основанные на численных методах. Восстановленные параметры экстраполируются на будущее и используются при моделировании самих механизмов для получения прогноза будущих событий.

Положения, выносимые на защиту:

1) Произведена адаптация существующих методов анализа данных и машинного обучения, таких как методы классификации, регрессии и методы на основе потоков событий, для задачи прогнозирования событий. Адаптация заключается в способах: построения матриц признаков из последовательности

событий как для обучения, так и для прогнозирования; формирования объясняемой переменной; интерпретации результата как прогноза событий. Рассмотрены недостатки и границы применимости существующих методов, даны рекомендации по их применению. Полученные результаты конкретизируют существующие подходы и методы прогнозирования в области экономических событий (С. 40–101).

2) Разработан новый подход к прогнозированию событий, в котором строятся алгоритмические модели механизмов образования событий. Под механизмом понимается детерминированная последовательность операторов над переменными и параметрами. Переменные инициализируются и изменяются внутри механизма, параметры изменяются вовне механизма. Параметры могут быть как статичными, так и динамичными. По выборке событий происходит восстановление этих параметров аналитическими методами, в случае если модель механизма это позволяет, или численными методами. После экстраполяции параметров прогноз будущих событий получается в результате моделирования механизмов образования событий. Полученные результаты являются принципиально новыми положениями по отношению к существующим методам прогнозирования экономических событий (С. 104–177).

3) Разработан новый метод восстановления динамических параметров по выборке событий, основанный на монотонной сплайновой коллокации восстановления функции по разным функционалам, таким как сами значения искомой функции, её первые и вторые производные, определённые интегралы на заданных, возможно, пересекающихся интервалах. Данный метод обеспечивает неотрицательность восстанавливаемой функции, что востребовано содержанием многих экономических задач. Ранее аналитическими методами параметры механизмов образования событий не восстанавливались. Полученные результаты дополняют и расширяют комплекс существующих статистических и математических методов на область прогнозирования экономических событий (С. 138–168).

4) Предложены способы экстраполяции параметров механизма образования событий: с помощью переноса тенденций с прошлого периода времени, с помощью регрессионных методов и с помощью разложения на сумму ограниченного количества гармонических функций. Данные практические предложения позволяют моделировать механизм образования событий в условиях, соответствующих будущему периоду времени, а не прошлому, тем самым повышая адекватность получаемых прогнозов (С. 169–173; 227–237).

5) Разработан способ определения оптимальных гиперпараметров для применяемых методов и алгоритмов восстановления и экстраполяции параметров механизма образования событий. В отличие от популярных методов анализа данных и машинного обучения, где гиперпараметры определяются только поиском на сетке, в разработанном способе гиперпараметры определяются на основе комбинирования поиска на сетке и локальной оптимизации в границах ячейки сетки алгоритмом Нелдера-Мида. Функция потерь минимизирует погрешность прогноза нескольких последних событий. Составлены принципиально новые метрики оценки качества прогноза событий на основе средней и среднеквадратичной, абсолютной или относительной погрешности прогноза их дат наступления и их характеристик. Подобных метрик для дискретных событий ранее не было (С. 175–178; 232–246).

б) Разработаны способы построения доверительных интервалов для проверки адекватности получающихся моделей прогнозирования экономических событий: а) на основе оценки дисперсии параметров сплайна; б) на основе имитационного моделирования; в) на основе моделирования механизма образования событий с параметрами, отклонёнными от их среднего значения. При промахе фактического события мимо рассчитанного доверительного интервала принимается гипотеза о неадекватности модели. Положения являются новыми, так как ранее доверительные интервалы для прогноза событий таким образом не рассчитывались (С. 178–186).

7) Разработан новый метод восстановления параметров модели механизма образования событий, основанный на численной оптимизации. Динамические параметры задаются кубическим сплайном, который кодируется через начальные значения и значения третьей производной в узлах сплайна. Функция потерь минимизирует квадрат относительных отклонений (моментов времени и значений) событий, полученных в результате моделирования, от фактических событий, плюс штраф на нелинейность динамических параметров. Используется специальный приём, ограничивающий объём вычислений. Полученные положения дают возможность восстанавливать параметры сложных алгоритмических моделей (С. 191–206).

8) Предложен метод подбора самих алгоритмических моделей механизмов образования событий, основанный либо на полном переборе с возможностью пропуска повторяющихся комбинаций операторов, либо на генетическом программировании. В этом способе функция потерь минимизируется не только по параметрам механизма, но и по структурной форме модели механизма образования событий. Благодаря полученным положениям появляется принципиальная возможность исследовать совершенно неизвестные события с помощью подбора алгоритмической модели механизма образования событий. Отличительной особенностью является алгоритм перебора схем операторов, а также их операндов (С. 208–215).

9) Разработан программный инструментарий на языке R и Python, реализующий комплекс методов, необходимых для прогнозирования будущих экономических событий. Создана программа на языке Python, реализующая метод восстановления функции по разным функционалам с помощью монотонной сплайновой коллокации. Разработана на языке Python и выложена в репозиторий PyPi библиотека eventometrics, которая реализует основные методы и алгоритмы прогнозирования экономических событий с помощью восстановления механизмов их образования. Библиотека имеет привычный

для аналитиков данных интерфейс, реализует эффективные вычисления, ключевые алгоритмы оптимизированы с помощью Cython, используются параллельные вычисления для поиска оптимальных гиперпараметров. Библиотека отличается высоким быстродействием. Запрограммирован на языке R способ восстановления параметров механизма образования событий с помощью численных методов. На языке R реализован способ автоматического перебора самих моделей механизма образования событий. Разработан компонент для платформы бизнес-аналитики Loginom, реализующий предложенный новый подход прогнозирования событий с помощью восстановления механизмов их образования. Полученные результаты открывают возможность широкого применения разработанной методологии в практике прогнозирования экономических событий (С. 159; 206; 211; 217–255).

Теоретическая и практическая значимость работы. Формирование комплексной методологии и разработка соответствующих технологий для прогнозирования экономических событий может создать критически важный научно-практический прорыв, закладывающий новое направление исследований (эвентометрика) в области анализа данных. Построение алгоритмических моделей и оценка их параметров является дальнейшим развитием научной мысли в экономических исследованиях, переводящим фокус с непрерывности изменения изучаемых закономерностей значений и показателей на дискретность явлений, возникающих в результате функционирования сложных процессов или механизмов. В дальнейшем это может привести к образованию направления обучения студентов, которые специализировались бы на построении моделей механизмов формирования экономических событий и оценке их параметров.

Полученные результаты позволяют поставить на новые научно-технологические рельсы и повысить уровень процесса анализа и прогнозирования экономических событий, в том числе редких, способствуют

получению новых знаний и закономерностей в области происходящих экономических процессов, приводящих к возникновению событий.

На основе созданного фундамента можно сконструировать целое множество моделей, предсказывающих появление событий, в том числе редких, в различных прикладных областях. Разработанная методология способствует лучшему пониманию причин возникновения отдельных событий в экономике, что позволяет получать более точные и обоснованные прогнозы экономических событий. В бизнесе это позволит к ним подготовиться и извлечь определённую выгоду, например, более эффективно сформировать план производства, план закупки сырья, план пополнения запаса, оптимизировать логистические расходы, уменьшить объёмы продуктовых запасов товаров, лежащих на складах, повысить уровень сервиса, добиться большего удовлетворения клиентов, применять новые маркетинговые приёмы и т.д. Также это даёт возможность органам государственного управления разрабатывать планы, отвечающие вызовам современного времени. Разработанные технологии могут способствовать быстрому обучению государственных служащих прогнозированию экономических событий для последующей выработки управленческих воздействий на протекающие в экономике и социуме процессы. Результаты исследования могут способствовать достижению национальных целей развития: комфортная и безопасная среда для жизни; достойный, эффективный труд и успешное предпринимательство; цифровая трансформация.

Методология и методы исследования опираются на проверенные математические, статистические и информационные средства. При адаптации существующих методов машинного обучения к прогнозированию событий рассматриваются методы опорных векторов, логистической регрессии, ближайших соседей, деревьев решений, градиентного бустинга, нейронных сетей, классические методы регрессии, теория потоков событий, методы прогнозирования прерывистого спроса.

Формирование моделей механизмов образования событий опирается на экономическую теорию, математический анализ, теорию управления запасами, системный анализ.

Восстановление параметров механизмов образования событий аналитическими методами опирается на теорию вероятностей, методы регрессии, математический анализ, сплайновую коллокацию (восстановление функции по функционалам), методы регуляризации, такие как кросс-валидация, метод L-кривой и невязки Морозова. Обеспечение положительности восстанавливаемой функции опирается на квадратичное программирование, метод множителей Лагранжа, алгоритм Лемке, метод внутренней точки.

Восстановление параметров механизмов образования событий численными методами базируется на имитационном моделировании, методах глобальной и локальной оптимизации. Автоматический подбор алгоритмических моделей опирается на комбинаторные методы и метод генетического программирования.

Экстраполяция восстановленных параметров основывается на регрессионных методах или анализе временных рядов, на алгоритме Куина-Фернандеса, который в свою очередь базируется на гармоническом анализе с помощью разложения в ряд Фурье.

Исследование вопросов точности опирается на математический анализ, теорию вероятностей, имитационное моделирование, метод Монте-Карло.

Разработка программного инструментария базируется на языках программирования R и Python. При разработке программ на Python задействуются средства оптимизации Cython и распараллеливания вычислений.

Степень достоверности, апробация и внедрение результатов исследования. Достоверность исследования подтверждается корректным использованием математического аппарата, моделей и методов. Достоверность полученных в исследовании математических выражений

демонстрируется на представленном в работе сквозном примере. Справедливость предложенных методик обосновывается компьютерным моделированием и примерами, основанными как на синтетических, так и на реальных данных.

Результаты исследования были доложены и обсуждены на ряде международных/всероссийских научных конференций / круглых столах / научных семинарах: на III Всероссийской научной конференции «Россия 2030 глазами молодых учёных» (Москва, ИНИОН РАН, 26 апреля 2012 г.); на VI Международной научно-практической Интернет-конференции «Современные проблемы моделирования социально-экономических систем» (г. Харьков, Харьковский национальный экономический университет, 3–12 апреля 2014 г.); на VII Международной научно-практической Интернет-конференции «Современные проблемы моделирования социально-экономических систем» (г. Харьков, Харьковский национальный экономический университет, 2–10 апреля 2015 г.); на IV Международной научно-практической конференции-биеннале «Системный анализ в экономике-2016» (Москва, Финансовый университет, 9–11 ноября 2016 года); на Международном круглом столе «Системная экономика, социально-экономическая кибернетика, мягкие измерения в экономике» (Москва, Финансовый университет, 8 июня 2017 г.); на Ежегодном международном круглом столе «Системная экономика, социально-экономическая кибернетика, мягкие измерения в экономике» (Москва, Финансовый университет, 6 июня 2018 г.); на Всероссийском межвузовском круглом столе «Мягкие измерения в научной и учебной деятельности» (Москва, Финансовый университет, 17 сентября 2018 г.); на V Международной научно-практической конференции-биеннале «Системный анализ в экономике» (Москва, Финансовый университет, 21–23 ноября 2018 г.); на Международной научно-практической конференции «Анализ данных, принятие решений и финансовые технологии», посвященной 100-летию Финансового университета (Москва, Финансовый университет, 23 мая 2019 г.); на

Ежегодной международной конференции «Системная экономика, социально-экономическая кибернетика, мягкие измерения в экономике» (Москва, Финансовый университет, 7 июня 2019 г.); на Международной научно-практической конференции «Системная экономика, социально-экономическая кибернетика и мягкие измерения в экономике» (Москва, Финансовый университет, 20 мая 2020 г.); на VII Международной научно-практической конференции «Современная математика и концепции инновационного математического образования» (Москва, Финансовый университет, 25 июня 2020 г.); на VI Международной научно-практической конференции-биеннале «Системный анализ в экономике» (Москва, Финансовый университет, 9–11 декабря 2020 г.); на Научном семинаре в Департаменте анализа данных и машинного обучения на Факультете информационных технологий и анализа больших данных (Москва, Финансовый университет, 20 января 2021 г.); на VIII Международной научно-практической конференции «Современная математика и концепции инновационного математического образования» (Москва, Финансовый университет, 28 мая 2021 г.); на Международной научно-практической конференции «Системная экономика, социально-экономическая кибернетика и мягкие измерения в экономике» (Москва, Финансовый университет, 24–25 июня 2021 г.); на Научном семинаре в ЦЭМИ «Прикладная статистика и моделирование реальных процессов» (Москва, ЦЭМИ, 8 декабря 2021 г.); на Научном семинаре на секции «Эконометрика» в Департаменте математики (Москва, Финансовый университет, 26 января 2022 г.); на IX Международной научно-практической конференции «Современная математика и концепции инновационного математического образования» (Москва, Финансовый университет, 9 июня 2022 г.); на Международной научно-практической конференции «Системная экономика, социально-экономическая кибернетика и мягкие измерения в экономике» (Москва, Финансовый университет, 1 июля 2022 г.); на VII Международной научно-практической конференции-биеннале «Системный анализ в экономике» (Москва, Финансовый университет, 7–8 декабря 2022 г.); на

Международной научно-практической конференции «Системная экономика, социально-экономическая кибернетика и мягкие измерения в экономике» (Москва, Финансовый университет, 15–16 июня 2023 г.); на X Международной научно-практической конференции «Современная математика и концепции инновационного математического образования» (Москва, Финансовый университет, 21 июня 2023 г.); на XI Международной научно-практической конференции «Современная математика и концепции инновационного математического образования» (Москва, Финансовый университет, 19 июня 2024 г.); на Ежегодной всероссийской научно-практической конференции с международным участием «Системная экономика, социально-экономическая кибернетика, мягкие измерения в экономике» (Москва, Финансовый университет, 20–21 июня 2024 г.); на Круглом столе «Экономико-математическое моделирование как инструмент исследования динамических процессов» в рамках Всероссийского фестиваля науки «NAUKA 0+» (Москва, Финансовый университет, 12 октября 2024 г.); на VI Международной научной конференции «Развитие современной экономической науки: проблемы, тенденции, перспективы» (Москва, МГИМО, 27–29 ноября 2024 г.); на VIII Международной научно-практической конференции-биеннале «Системный анализ в экономике» (Москва, Финансовый университет, 12–14 декабря 2024 г.); на XI Международной научно-практической конференции «Современная математика и концепции инновационного математического образования» (Москва, Финансовый университет, 11 июня 2025 г.); на Всероссийской научно-практической конференции с международным участием «Системное моделирование в экономике, финансах и управлении» (Москва, Финансовый университет, 24–25 июня 2025 г.); на XIV Научно-практической конференции «Математическое и компьютерное моделирование в экономике, страховании и управлении рисками» (г. Саратов, СГУ, 20–22 ноября 2025 г.).

Результаты исследования были использованы при выполнении НИР по теме проекта исследований гранта РФФИ «Ёмкостный метод анализа редких

событий в экономике» в 2019–2021 гг., № 19-010-00154, в котором автор являлся руководителем проекта.

Результаты исследования были использованы при выполнении НИР Финуниверситета (Государственное задание ПИ-20, приказ от 26.12.2024 № 3268/о) по теме «Разработка методологии моделирования и прогнозирования динамики пространственного развития Российской Федерации на основе data-driven management».

Получено Свидетельство о государственной регистрации программы для ЭВМ № 2025688461 «Программный продукт, реализующий метод восстановления функции одновременно по значениям, значениям первой и второй производной, определённым интегралам, наблюдаемым с погрешностью, с условием обеспечения положительности или монотонности».

Материалы диссертационной работы внедрены в практическую деятельность компании ООО «АУМЕД» в целях улучшения планирования логистических процессов. Разработанные методики и программный инструментарий расширяют набор возможностей для анализа и прогнозирования таких событий, как будущие заказы отдельных корпоративных клиентов, появление ненулевого спроса или же определение величины запасов, что способствует определённости в принятии решений и в результате улучшению уровня сервиса.

Материалы диссертационной работы внедрены в практическую деятельность компании ООО «Квайссер Фарма» в целях совершенствования процессов планирования реализации фармацевтической продукции. Разработанная методология прогнозирования редких событий в экономике позволяет точнее определять будущие потребности в поставках фармацевтической продукции. Отдельные положения нашли своё применение при обосновании спада или повышения спроса на биологически активные добавки в период пандемии COVID-19.

Материалы диссертации используются Кафедрой бизнес-информатики Факультета информационных технологий и анализа больших данных Финансового университета в преподавании учебных дисциплин «Платформы бизнес-аналитики», «Технологии продвинутой аналитики».

Апробация и внедрение результатов исследования подтверждены соответствующими документами.

Публикации. Материалы исследования опубликованы в 32 научных публикациях общим объёмом 61,61 п.л. (авторский объём – 58,07 п.л.), в том числе одной авторской монографии объёмом 18,5 п.л., 18 статьях общим объёмом 25,09 п.л. (авторский объём – 21,55 п.л.), опубликованных в рецензируемых научных изданиях, определённых ВАК при Минобрнауки России, из которых 4 статьи общим объёмом 8,25 п.л. (авторский объём – 7,65 п.л.) опубликованы в изданиях, входящих в цитатно-аналитическую базу RSCI, и 2 статьи авторским объёмом 4,1 п.л. опубликованы в изданиях, индексируемых в цитатно-аналитической базе «Scopus» (Q1 и Q2), а также 4 статьи авторским объёмом 7,22 п.л. опубликованы в рецензируемом научном издании «Известия Кабардино-Балкарского научного центра РАН», включённом на момент публикации статей в перечень ВАК при Минобрнауки России по научной специальности 08.00.13 – Математические и инструментальные методы экономики (физико-математические науки). Все публикации по теме диссертации.

Структура и объём диссертации обусловлены целью и задачами исследования. Диссертация состоит из введения, 5 глав, заключения, списка литературы, состоящего из 219 наименований, и одного приложения. Текст диссертации изложен на 358 страницах, содержит 31 таблицу и 124 рисунка.

Глава 1

Анализ и адаптация существующих методов прогнозирования редких экономических событий

1.1 Проблематика прогнозирования редких экономических событий и терминология

На протяжении всей истории люди пытаются предсказать будущее, чтобы в дальнейшем принимать обоснованные решения. В современное время это по-прежнему актуально. В микроэкономике компании, анализируя потребности покупателей, пытаются предсказать спрос на продукты или услуги, чтобы установить цены, объём производства или разработать и вывести на рынок новый продукт. Производственные компании стараются предсказать спрос, чтобы управлять собственными запасами сырья или готовой продукции. Страховые организации всё время следят за изменениями и анализируют различные риски, чтобы определить размер страховой премии и страховых выплат. На финансовых рынках инвесторы прогнозируют движение цен на акции или финансовые инструменты, чтобы получать прибыль во время самих торгов. В макроэкономике государственные органы, анализируя различные показатели, пытаются предсказать будущее состояние экономики, чтобы разработать такую политику, которая приводила бы к развитию и укреплению страны. Для всего этого люди тем или иным способом вынуждены получать прогноз.

В давние времена люди были вынуждены обращаться к различным гадателям или оракулам, из-за чего прогноз был случайным или субъективным. С развитием науки в области понимания тех или иных физических или социальных процессов начали обращаться к математическим или статистическим моделям и методам. В таких моделях и методах содержится опыт и интуиция разработчиков, их понимание, их выражение действительности, прошедшее через множество этапов проверки и подтверждения. Развитие компьютерных технологий и доступность огромных

объёмов данных привели к созданию более точных алгоритмов и моделей прогнозирования. Но вместе с этим появились проблемы, связанные с возросшей сложностью и трудностью понимания устройства самих получающихся моделей, а также объяснимостью получаемых результатов, что делает результат прогнозирования неубедительным и ненадёжным.

Особенно важной является задача прогнозирования событий. Знание момента появления событий позволит грамотно к ним подготовиться, извлечь определённую выгоду или уменьшить возможные убытки. Сложно переоценить способность предсказывать будущие события. Грамотный прогноз может определить успех или спасти от неудач как в государственном управлении, так и в управлении частным бизнесом. Правительство может заблаговременно начать разработку стимулирующих мер поддержки, центральный банк – заранее корректировать свою денежно-кредитную политику. Частные компании могут спрогнозировать сезонный спрос, оптимизировать запасы и логистику, сократить издержки, избегая дефицита и затоваривания, управлять персоналом, приостановить рискованные инвестиции или, наоборот, инвестировать капитал в наиболее перспективные стартапы.

Отдельный интерес составляет способность прогнозирования редких экономических событий. Эта тема крайне сложная хотя бы из-за неоднозначности трактовки и понимания самого термина редких событий. Из-за средств массовой информации большинству людей, когда они сталкиваются со словосочетанием «редкие события», первое, что приходит в голову, это что-то экстраординарное по своей величине и значению, что превосходит мыслимые границы. В этот момент в сознании людей всплывают такие события, как крах на финансовых рынках, дефолт крупного банка, землетрясение, падение метеорита или, наоборот, выигрыш в лотерее, внезапное появление новой перспективной научной технологии, неожиданный скачок продаж некоторого товара, появление вирусного аудио- или видеоклипа и т.д. Тем самым большинство людей в первую очередь

представляют себе значимость или величину события. Но будет ли редким событием то, что ежедневная выручка небольшого предприятия внезапно увеличилась в 100 или более раз? Для этого малого предприятия, скорее всего, будет. Однако есть компании, которые получают такие же, если не большие, прибыли на регулярной основе. То есть то, что для одного может являться редким событием, может таковым не являться для другого. С другой стороны, возможно обрушение привычных больших значений выручки, вплоть до нулевых значений, и кто-то также будет называть это редким событием. Играть роль не величина, она может быть большой или маленькой, положительной или отрицательной, а внезапность. То есть те люди, которые под выражением «редкие события» имеют в виду кризисы, дефолты и т.д., на самом деле понимают те моменты времени, когда нарушаются их ожидания, когда появляется необходимость корректировать свою повседневную жизнь. В эти моменты разбиваются об объективную реальность мысленные или математические модели, применяемые для планирования следующего дня. Такие редкие события Нассим Николас Талеб называет «чёрными лебедями» [1]. Концепции чёрных лебедей Талеба надо уделить отдельное внимание.

Чёрные лебеди Н.Н. Талеба

В своей знаменитой книге [1] Талеб называет события чёрными лебедями, если они обладают тремя характеристиками: 1) аномальность (ничто их не предвещало); 2) огромная сила воздействия; 3) желание искать им объяснения после того, как они произошли. Сам термин «чёрные лебеди» является лишь метафорой, описывающей ситуацию, когда европейцы были абсолютно уверены, что чёрных лебедей не существует. На протяжении всей книги Талеб говорит о невозможности прогнозирования «чёрных лебедей». Книга в большей степени посвящена описанию возможных причин человеческих заблуждений, основанных на когнитивных искажениях. Среди них Талеб выделяет: ошибку выживших (о неудачных случаях история умалчивает), ошибку подтверждения (склонность интерпретировать информацию в пользу уже сложившихся убеждений), искажение нарратива

(потребность уложить историю в ряд связанных или не связанных фактов для лучшего запоминания), игровую ошибку (объяснение случайности через упрощённые примеры из азартных игр), ретроспективное искажение (очевидность произошедшего при неспособности его предвидения), проблему «индюшки»/индукции (экстраполяция прошлого без учёта возможности кардинальных изменений), эпистемическую самонадеянность (переоценка своих знаний и недооценка неопределённости), Платонизм (концентрация на чётких идеалистических понятиях в ущерб менее определённым структурам), склонность к «туннелированию» (фокус на известных рисках и игнорирование неизвестных), недооценку роли случайности (приписывание успеха личным качествам и игнорирование роли случайности), иллюзию контроля (вера в способность всё контролировать), огрехи-перевёртыши (принятие отсутствия свидетельств за свидетельство отсутствия), проблему переключения между эмпирической (быстрой, интуитивной, бессознательной) и рациональной (медленной, требующей усилий) системами мышления [1].

У Талеба редкое событие, называемое им «чёрным лебедем», есть следствие проявления когнитивных искажений. Такие события появляются в месте, которое он называет «платонической складкой», – месте, где идеализированные модели входят в контакт с реальностью, в результате чего проявляются огрехи этих моделей [1]. Другими словами, такие редкие события являются сломом наших ожиданий, когда модели не справляются с объяснением происходящего.

В своей книге Талеб утверждает, что чёрных лебедей предсказать невозможно. Вместо этого он даёт рекомендации, как избежать когнитивных искажений. Даёт совет извлекать выгоду из проблемы прогнозирования и эпистемической самонадеянности. Описывает несколько хитростей: уметь отличать «хорошие» случайности от «плохих» (где непредсказуемость может обернуться крупным выигрышем, а где провалом); не гнаться за точностью и конкретикой, а мыслить шире; хвататься за любую возможность; быть начеку и остерегаться разработанных государственных планов (у чиновников нет

цели докопаться до истины); не тратить время на борьбу с прогнозистами, дающими «дурацкие прогнозы»; стараться выбирать ситуации, в которых благоприятные последствия значительно масштабнее неблагоприятных [1]. Книга Талеба, безусловно, полезна для людей, собирающихся заниматься бизнесом.

Вместе с тем на страницах книги Талеб пишет: «событию недостаточно быть редким или каким-то из ряда вон выходящим, чтобы обрести статус чёрного лебедя, – оно должно быть неожиданным», «многие редкие события обладают неким набором составляющих, всё же доступных для изучения», «мы можем, так сказать, превратить этих чёрных лебедей в серых лебедей». Однако во всей книге термин «серые лебеди» встречается всего три раза (и один раз в глоссарии). Математических методов прогноза «серых лебедей» Талеб не даёт (только упоминает, что, возможно, пригодится фрактальный принцип (самоподобия) Мандельброта) [1].

В этой диссертационной работе не прогнозируются «чёрные лебеди» в смысле моментов времени, когда происходит поломка моделей, моментов, когда ожидания не соответствуют действительности. Можно согласиться с утверждением Талеба, что такие события нельзя предсказать. Даже если бы была возможность предсказать момент поломки одной модели, находящейся в чьей-то одной голове, это невозможно сделать для каждой модели в голове каждого человека. Под редкими событиями в этой работе понимается, скорее, то, что Талеб называет серыми лебедями.

«Чёрные лебеди» могут заставить нас обратить внимание на определённые процессы, заставить разбираться в причинах возникновения редких событий, строить соответствующие модели, моделировать процессы и получать прогнозы событий. В этом смысле получаемые прогнозы редких событий вовсе не будут прогнозами «чёрных лебедей» Талеба, но среди них могут получиться такие удивительные прогнозы событий, которые будут обладать тем самым уникальным свойством неожиданности, которое позволило бы назвать их «чёрными лебедями». Научившись предсказывать

события, будем уметь предсказывать в том числе и редкие события. Другого пути решения задачи прогнозирования редких событий на данный момент не видится.

В итоге в этой работе занимаемся прогнозированием тех событий, которые можно прогнозировать. Конечно же, как бы основательно мы ни изучали происходящие в мире процессы, наши модели могут быть несовершенны. Помимо прогнозируемых событий («серых лебедей»), среди которых могут быть удивительные (которые можно назвать спрогнозированными «чёрными лебедями»), могут всё же быть случаи проявления несовершенства наших моделей, когда прогноз сильно расходится с реальностью (истинные «чёрные лебеди»). Такое несоответствие будет сигналом, чтобы заняться корректировкой модели.

Далее дадим более формальные определения используемым терминам.

Термин «событие»

Из-за доминирования статистических методов на протяжении нескольких веков сложилось некоторая двойственность в понимании термина «событие». В теории вероятностей событие понимается как некоторое подмножество из заранее известного множества элементарных исходов (при проведении контролируемого испытания, опыта) [5; 6]. Тем самым достаточно часто можно встретить трактовку, что событие есть результат опыта со случайным исходом. Но термин «событие» существовал ещё до развития статистических методов. В книге «История слов» историка-лингвиста и академика РАН Виноградова В.В. [7] говорится, что термин «событие» встречается в 1096 году в церковнославянской письменности и имеет значение «исполнение» и «осуществление». В словаре Даля [8] слово «событие» определяется как «происшествие, что сбылось». Эти трактовки не содержат никакого упоминания о случайной природе события.

В данной работе термин «событие» понимается именно в широком смысле, как что-то произошло или случилось, не обязательно связанное со статистикой, хотя речь ниже и будет вестись о различных статистических

методах. Заметим, что Колмогоровское определение события (подмножество из множества элементарных исходов), например событие выпадения единицы или двойки на шестигранном кубике, также можно перефразировать как наступление определённой комбинации условий (само выполнение условий и есть событие).

Обычно события появляются вне рамок каких-либо испытаний и независимо от воли исследователя этих событий. В данной работе особое внимание уделяется механизмам (процессам) образования событий. Предполагается, что события образуются не сами по себе, а в результате некоторого физического, политического, социального или любого другого процесса. Механизм (процесс) может задействовать человека, а может не задействовать. Однако события возникают независимо от того, проводил исследователь какие-либо испытания или нет.

В результате вышесказанного в этой работе под термином «событие» понимается следующее:

Событие – факт или явление, которое происходит в определённый момент времени как результат проявления экономических, политических, социальных, физических и других процессов, протекающих в той или иной сфере.

Именно такое общее понимание события, не связанное с испытаниями со случайным исходом, используется в работе. Под прилагательным «экономическое» (в словосочетании «экономическое событие») понимаем область влияния события на мир. Таким образом:

Экономическое событие – событие, которое влияет на экономические процессы (и которыми может быть порождено), на экономических агентов или на общее экономическое состояние в целом.

Далее по тексту часто будем писать просто «событие», подразумевая экономическое событие.

Редкость событий

«Редкость» или «частота» показывают, насколько количество или вероятность одних событий отличается от количества или вероятности других событий. Однако понятия «редкость» или «частота» являются такими же диалектическими понятиями, как понятия «много» или «мало». Где заканчиваются частые события и начинаются редкие события, зависит от контекста.

Существуют объективные проблемы проведения чёткой количественной границы между качественными понятиями. Количественное выражение фиксирует лишь избранные аспекты качества, не обеспечивает полноту описания исходного качества. Сведение сложного, многомерного качества к одному или нескольким количественным показателям может искажать реальность. Также сам выбор количественных показателей может являться субъективным. У Николаса Талеба акцент делается на неожиданности (аномальности) и на огромное воздействие событий на мир («чёрные лебеди») [1]. В теории информации из-за своей редкости события уже обладают большей информацией [9]. Другой проблемой является интерпретируемость. Для одного человека событие может представлять значительный интерес, когда для другого нет. Для одних систем даже тысячи событий в секунду может быть недостаточно, чтобы такие события назывались частыми, а для других систем события, которые возникают раз в несколько месяцев, по-прежнему могут являться частыми событиями.

Во втором законе диалектики о переходе количества в качество (Энгельс Ф.) мерой перехода накопленных количественных изменений в другое качество может являться интервал. Причём границы интервала зависят от разных условий, и их может быть невозможно чётко установить для различающихся систем, объектов и задач. В итоге вопрос о полной сводимости количества к качеству остаётся дискуссионным философским вопросом.

С оговоркой на вышесказанное дадим количественную оценку для качественного понятия «редкие события». В исследовании [215, с. 291–295]

показано, что только при трёх и более событиях за период требуемой гранулярности (интервал времени) восстановление функции скорости расхода ресурса при потере части данных было более точным с помощью агрегирования событий во временной ряд по сравнению с индивидуальным рассмотрением событий. Таким образом, индивидуальное рассмотрение событий более выгодно при количестве событий меньше или равным двум. Число 2 может быть той границей, когда количество событий позволяет говорить о редкости. В итоге:

События, которые появляются в среднем в количестве два и менее за период требуемой гранулярности, будем называть редкими.

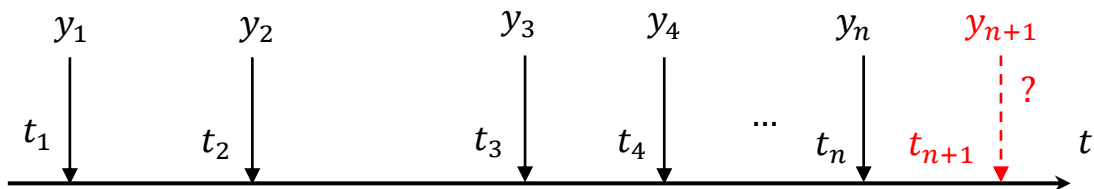
Однако теперь надо пояснить понятие «гранулярность». Гранулярность данных – степень детализации данных, которая может существенно повлиять на точность, надёжность и удобство использования прогнозов [10]. Детализация зависит от потребителя в компании, которому нужен прогноз, и напрямую зависит от циклов планирования. Для краткосрочных планов это может быть неделя, для долгосрочных – год [10]. Другими словами, исследователь/аналитик вынужден рассматривать ограниченные интервалы времени, за которые происходит расчёт каких-нибудь характеристик. Сама задача может диктовать исследователю, какая ширина интервала является чрезмерно маленькой, какая чрезмерно большой, а какая удовлетворительной. Задача определяет требования к гранулярности данных (ширине рассматриваемых интервалов времени, за которые надо наблюдать или подсчитывать события).

Таким образом, события называем редкими, когда за интересующие исследователя периоды времени появляется в среднем два и менее события.

В этой работе делается акцент на дискретности событий. Вынесенный в заглавие термин «редкость» событий дополнительно подчёркивает эту дискретность. В то же время изучаемые в работе методы могут быть применены для прогнозирования любых дискретных событий, необязательно редких (в том смысле, что они особенные). Изучаемые методы могут быть

применены как для нередких, так и для редких событий. Дело в том, что многие существующие методы работы с частыми событиями не подходят или малоэффективны для работы с редкими событиями, в основном по причине редкости наблюдений. Например, нет смысла по выборке редких событий строить и анализировать временные ряды – такие ряды будут содержать огромное количество нулевых значений, так как интервалы времени между событиями очень большие. При этом укрупнение интервалов времени при агрегировании, например, когда вместо агрегирования по месяцам происходит агрегирование по кварталам или годам, теряет какой-либо смысл, так как исследователя интересуют более мелкие интервалы времени.

Самое близкое к тому, что понимается под событиями в данной работе, является представление событий в виде потока событий из теории случайных процессов [11] (иногда потоки событий также называют точечным процессом [12]). Поток событий представляет собой последовательность моментов времени возникновения событий t_i , графически это выглядит как нанесение точек на ось времени, рисунок 1.1. Помимо этого события иногда сопровождаются некоторыми характеристиками или воздействиями на мир y_i . Например, в страховании [13, с. 22] событиями являются страховые случаи, а характеристиками являются выплаты по этим страховым случаям. В торговле величинами y_i могут быть объёмы покупок. В любом другом случае воздействием y_i может быть что угодно. Подробнее о потоках событий будет рассказано ниже. Предлагаемое решение не использует потоки событий.



Источник: составлено автором.

Рисунок 1.1 – Основная задача

В данной работе внимание концентрируется на том случае, когда агрегирование нескольких потоков событий в один поток невозможно по той причине, что у нас либо нет этих нескольких потоков, либо такое

агрегирование теряет смысл (например, когда бессмысленно агрегировать продажи от клиентов, находящихся в разных городах, когда требуется прогноз продаж по каждому городу в отдельности), а также когда нет положительного эффекта от агрегирования потоков. Если объединить вместе множество потоков событий, то никакой новой зависимости не проявляется, наоборот – может потеряться имеющаяся информация. Таким образом, под «редкими событиями» понимается прежде всего форма в виде потока событий, причём такие потоки нельзя объединять. Требуется извлечь некоторую закономерность именно из таких одиночных (не агрегированных) потоков.

Отметим, что редкие события зачастую называют по-разному, подразумевая одно и то же, а также, наоборот, называя одинаково, подразумевают совсем разные понятия [14; 15]. Такими терминами являются: аномалии, новизна, выбросы, инциденты. Задачи, в которых используется эта терминология, также отличаются. Это классификация событий во временных рядах, классификация по наблюдаемым признакам, определение вероятности возникновения события в заданном периоде времени, определение вероятности выхода за границы допустимых условий. Но в большинстве таких задач не происходит прогнозирования события в будущем, а происходит детектирование событий в настоящем.

Основная задача, на решение которой направлено данное исследование, заключается в том, чтобы научиться предсказывать будущие события $(t_{n+1}; y_{n+1})$, рисунок 1.1. Интересует прежде всего определение самого момента возникновения будущего события t_{n+1} , а также его характеристики y_{n+1} (его воздействие на мир).

Задача прогнозирования событий

Задача прогнозирования будущих событий может встречаться во многих областях народного хозяйства. В первую очередь (с чем связано большинство примеров данной работы) прогнозирование событий может быть востребовано в торговле или производстве, когда необходимо прогнозирование возникающей дискретно потребности в товарах, сырье или

услугах со стороны неподконтрольных клиентов или организаций. Например, определение моментов времени наступления событий дискретного спроса поможет заранее спланировать план производства, план поставки и хранения готовой продукции, оптимизировать логистические расходы. Прогнозирование потребности в услугах поможет оптимизировать работу персонала, заранее подготовить все расходные материалы в нужном количестве. В добывающей и обрабатывающей отрасли может пригодиться прогноз исчерпания ресурса оборудования, что позволит заблаговременно организовать его обслуживание. В сельском хозяйстве, помимо прогноза погоды, может пригодиться прогноз изменения плодородности почвы и прогноз вспышек популяции вредителей. В сфере транспорта и логистики прогноз событий позволит оптимизировать маршруты и цепочки поставок. В энергетике прогноз событий позволит подготовиться к скачкам потребления, требующим перераспределить нагрузку энергии. В финансовом секторе позволит предвидеть изменения цен на определённые инструменты. В социальной сфере поможет предвидеть вспышки возмущений или протесты, что позволит скорректировать внутреннюю политику и направить усилия на решение или разъяснение ситуации для её принятия. Прогноз событий позволяет перейти от управления «по факту» к управлению «по прогнозу» во многих областях народного хозяйства, во многих задачах и на многих уровнях. Прогнозирование событий востребовано столь же широко, как применение методов оптимизации, (линейного, нелинейного, динамического программирования) в задачах народного хозяйства.

В настоящее время во всех отраслях народного хозяйства существует ряд актуальных задач, требующих применения методов прогнозирования дискретных событий. Новые задачи продолжают появляться постоянно. Объекты народного хозяйства могут быть разнородны, алгоритмы решения задач отличаются разноплановостью. Необходимость прогнозирования событий в каждой из них приводит к необходимости создания отдельной методологии, ориентированной на решение задачи прогнозирования событий.

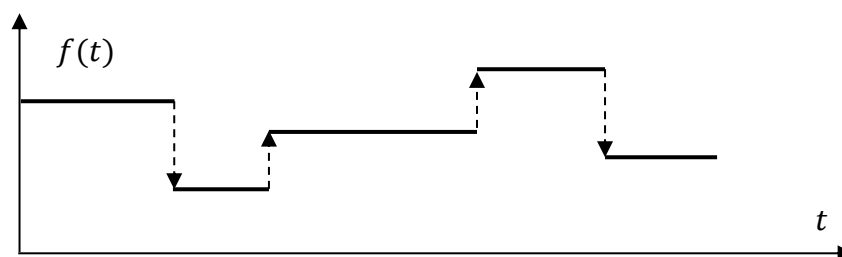
Несмотря на многоплановость задач народного хозяйства, можно выделить общие черты проблемы прогнозирования событий:

- дискретный характер данных – события появляются в определённые моменты времени, между которыми ни событий, ни значений не наблюдается (набор данных не является временным рядом);
- возможно наличие характеристик событий, например, объём поставок, прибыль, расходы и т.д., или же сам факт появления является единственной характеристикой;
- возможно наличие нескольких разных источников событий, которые могут объединяться в один общий поток событий, что усложняет его изучение;
- разнородность шкал оси времени – в одних задачах время может измеряться секундами, в других – годами;
- отсутствие метрик и критериев, как в классических методах машинного обучения, пригодных для оценивания качества полученного прогноза событий;
- каждое событие имеет существенное значение, и его нельзя выбрасывать просто так, как иногда поступают в машинном обучении;
- некачественный прогноз событий может приводить к большим потерям;
- анализ данных дискретных событий может требовать больших трудозатрат и привлечение специалистов.

В существующей науке нет проработанных унифицированных методов прогнозирования событий, впрочем, сам класс задач прогнозирования событий отсутствует среди современных разделов анализа данных, что является научной проблемой данного диссертационного исследования. В свою очередь, отсутствие соответствующих методологических и информационных инструментов прогнозирования событий затрудняет функционирование и развитие экономики страны, что определяет народно-хозяйственную проблему.

Есть ещё аспекты, на которые следует обратить внимание.

Одно из существенных влияний может заключаться в самом сдвиге парадигмы от непрерывных функций, лежащих в основе практически всего математического и экономического анализа, к дискретным событиям и последующим изменениям. Например, в статье [16] утверждается, что на самом деле в экономике почти все реальные траектории показателей в основном разрывны и состоят из интервалов постоянства, рисунок 1.2. Каждая транзакция (бухгалтерская) вызывает скачок в показателях, которые остаются постоянными до следующей транзакции. Большинство, если не все реальные экономические показатели, не могут быть непрерывными функциями. Приводится пример изменения численности рабочих, естественно, такой показатель может изменяться только дискретно на целую величину. В статье [16] утверждается о бессмысленности составления каких-либо моделей прогнозов, основанных на непрерывных функциях, без попытки изучения дискретных изменений. Цитата: «В экономике реальный ряд – это не аппроксимация. Он реален, точен и дискретен. Ряд есть весь, он реальность, он точен. Изучайте его. Так зачем его портить непрерывным приближением? Это просто инерция мышления» [16].



Источник: составлено автором по материалам [16].

Рисунок 1.2 – Кусочно-постоянный вид реальных экономических показателей

К сожалению, в указанной статье не даётся никаких альтернативных методов решения проблемы. В данной же работе как раз восполняется этот пробел, создаётся методология прогнозирования событий, которая впоследствии может быть использована для качественного перехода от непрерывности к дискретности в анализе экономических показателей. Конечно же, не стоит скатываться в крайность – непрерывные функции

используются там, где нет возможности детализировать изменения до уровня отдельных событий из-за отсутствия информации.

Заметим: идеи дискретного подхода уже начинают распространяться. Например, в ИТ-секторе, помимо мышления относительно сущностей, появляется мышление относительно событий. В том смысле, что в базах данных хранится не просто информация об объектах, а хранится информация об изменениях, связанных с объектами [17]. Ведение лога изменений является чуть ли не обязательным требованием к современным хранилищам данных.

Безусловно, сам прогноз экономических событий важен. В предиктивной аналитике [18] отмечается возможность управления рисками в производстве, продажах и финансах. Например: заблаговременное обслуживание и замена деталей дорогостоящего оборудования, чтобы избежать поломки; прогнозирование поведения покупателя по приобретению товара или услуги; возможность дефолта заёмщика в ближайшую перспективу; прогнозирование страховых выплат. Также прогноз событий позволяет применять маркетинговые методы, например реклама или скидки в определённые дни, что может способствовать повышению прибыли и повышению лояльности клиентов [18]. В работе [19, с. 254] отмечается, что наибольшее влияние на бизнес оказывало внедрение BI-систем (Business Intelligence, бизнес-аналитики) с функциями предиктивной аналитики. Добавление новых возможностей по прогнозированию событий к набору инструментов предиктивной аналитики может оказать значительный экономический эффект.

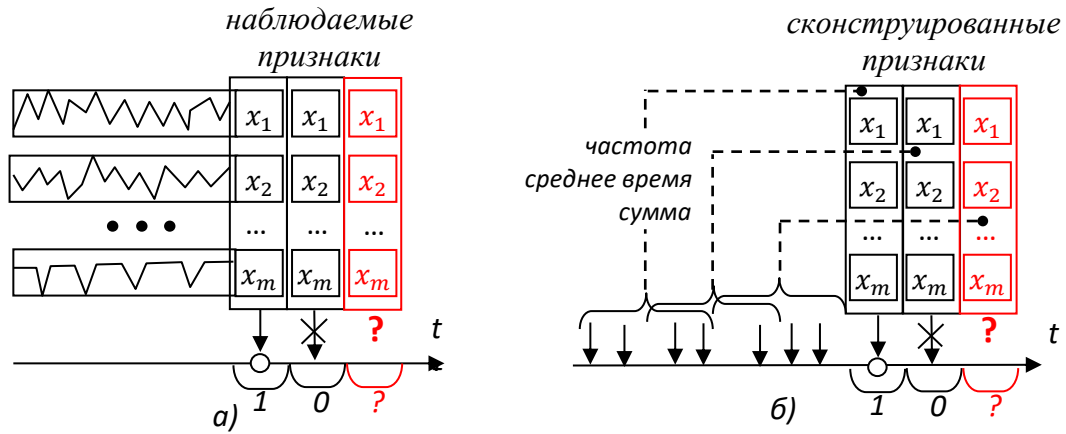
Прогнозирование событий может стать важной частью систем раннего предупреждения (Early Warning Systems), например возникновения финансовых кризисов на валютных рынках. В таких системах практически в реальном времени отслеживаются разные индикаторы, определённые комбинации значений которых могут свидетельствовать о начале кризиса. Однако авторы работы [20] пишут, что к результатам следует относиться с осторожностью, их интерпретация неоднозначна, так как ранние модели

предсказывали лишь треть фактически случившихся кризисов, а половина предсказаний была ложными срабатываниями. Другой проблемой является то, что индикаторы может быть трудно измерить и они могут быть неинформативными [20]. Системы раннего предупреждения в основном строятся на машинном обучении, чтобы распознавать определённые образы [21]. Если же в качестве индикаторов использовать лаговые значения или прогнозы индикаторов (экстраполяции), тоже полученные методами машинного обучения, но другими моделями, то получится уже способ прогнозирования событий. Это далеко не весь перечень способов использования возможности прогнозирования событий в экономике.

Далее рассматриваются существующие методы и их адаптация для задачи прогнозирования событий. Разбираются условия и границы их применимости, обсуждаются недостатки. Следующие подразделы этой главы задумывались как «поваренная книга» или справочник по применению существующих методов для прогнозирования событий.

1.2 Методы классификации в прогнозировании событий

В задачах классификации обычно требуется по значениям наблюдаемых признаков $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ отнести некоторый объект к одному из классов, чаще всего к двум классам, $Y_i \in \{0,1\}$ (или $\{-1,1\}$). Для сведения задачи прогнозирования событий к такой задаче бинарной классификации задают интервал времени определённой величины, и появление события на этом интервале соответствует наблюдению $Y_i = 1$, а отсутствие события на этом интервале соответствует $Y_i = 0$. Причём признаками X_i могут являться как действительно наблюдаемые некоторые значения, например температура за окном или курс валют, так и искусственно сконструированные признаки, например средняя частота или время между событиями, рисунок 1.3.



Источник: составлено автором.

Рисунок 1.3 – Методы классификации при прогнозировании событий

Конечно же, можно одновременно использовать как действительно наблюдаемые признаки, так и искусственно сконструированные.

В таком подходе по выборке событий, предшествующей интересующему исследователя периоду, определяются параметры классифицирующей модели. Однако и при оценке параметров модели, и при валидации модели необходимо на что-то опираться. Поэтому последнее наблюдение необходимо приберечь для валидации, а обучение проводить, опираясь на предпоследнее наблюдение. Само же прогнозирование будущего события следует выполнять по всей доступной выборке, включающей последнее событие. Достаточно часто валидация моделей в некоторых встречающихся работах осуществляется ошибочно, для проверки используется не последнее наблюдение, а случайно отобранные наблюдения, подробнее будет показано ниже.

Заметим, что методы классификации не используются для предсказания числовых значений, поэтому предсказать характеристики/воздействия событий y_{n+1} будет затруднительно.

Примечание – Если разбить числовые значения на группы, то можно использовать методы классификации для предсказания группы и затем вернуть значение, соответствующее этой группе. Так происходит, например, в методе DecisionTreeRegressor, о котором будет сказано ниже.

Далее рассмотрим несколько популярных и известных математических методов классификации.

Метод опорных векторов

Метод опорных векторов (Support Vector Machine, SVM) остаётся одним из самых применяемых методов классификации.

Примечание – Считается, что метод опорных векторов впервые был предложен Владимиром Наумовичем Вапником в 1964 году [22]. Однако из этой ранней работы тяжело разглядеть привычную установившуюся форму метода, угадываются только некоторые фрагменты.

Метод опорных векторов очень часто входит в группу методов классификации, среди которых выбирается наиболее подходящий. Метод опорных векторов обширно освещён в научной литературе [23] и входит во многие учебники и курсы по машинному обучению [24].

Случай линейно неразделимых классов оказывается более интуитивно понятным, чем случай с линейно разделимыми классами (между точками наблюдений можно провести разделяющую гиперплоскость). В классическом виде требуется найти m (где m – размерность задачи) неизвестных параметров $w = (w_1, \dots, w_m)$ и 1 параметр b уравнения гиперплоскости $h(X) = wx + b = w_1x_1 + w_2x_2 + \dots + w_mx_m + b = 0$, а также n неизвестных переменных $\xi = (\xi_1, \dots, \xi_n)$, решая следующую оптимизационную задачу:

$$\|w\|^2 + C \sum_{i=1}^n \xi_i \rightarrow \min, \quad (1.1)$$

$$\begin{cases} Y_i \cdot (wX_i + b) \geq 1 - \xi_i, & i = 1, \dots, n \\ \xi_i \geq 0, & i = 1, \dots, n \end{cases}, \quad (1.2)$$

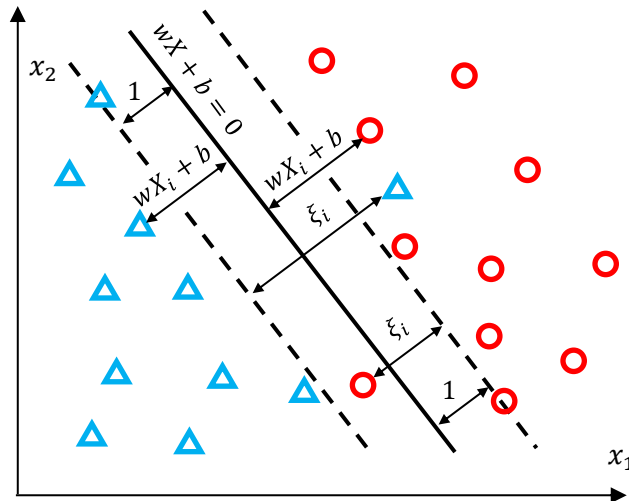
где $\|w\|^2$ – квадрат второй нормы, $\|w\|^2 = \sum_{i=1}^m w_i^2$;

Y_i – класс наблюдения, $Y_i \in \{+1, -1\}$;

C – параметр регуляризации (параметр алгоритма), определяющий, какой вес имеет вторая сумма в оптимизационной задаче.

Можно дать такую простую интерпретацию данной оптимизационной задаче. Здесь $(wX_i + b)$ есть расстояние i -го наблюдения до гиперплоскости, которое перемасштабировано так, что минимальное расстояние среди всех наблюдений становится равным 1. Умножение на Y_i нужно для того, чтобы для отрицательного класса, лежащего по другую сторону гиперплоскости, вновь сделать расстояния положительными. Неотрицательные переменные ξ_i служат для ослабления ограничений, так как для линейно неразделимых классов некоторые наблюдения оказываются с неправильной стороны

гиперплоскости. Тем самым переменные ξ_i показывают, насколько наблюдения отделились от границы своего класса и вылезли на чужую сторону гиперплоскости. На рисунке 1.4 представлена наглядная визуализация для случая двух переменных.



Источник: составлено автором как обобщение сути метода, подчерпнутой из разных открытых источников.

Рисунок 1.4 – Интерпретация метода опорных векторов с нестрогими границами

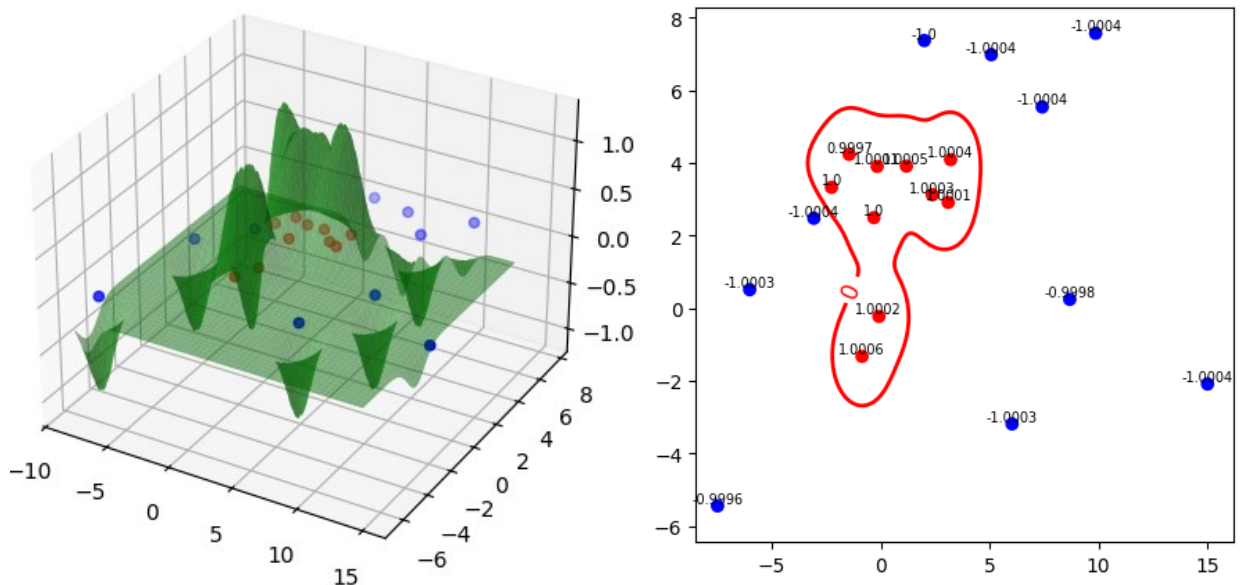
Примечание – Целевую функцию логично было бы даже переписать как $\sum_{i=1}^n \xi_i + C \|w\|^2 \rightarrow \min$, чтобы сделать акцент на том, что надо минимизировать это залезание за границу гиперплоскости, дополнительно штрафую параметры гиперплоскости, чтобы они не были слишком большие. В классическом виде $C \sum_{i=1}^n \xi_i$ называют регуляризацией, а основным минимизируемым выражением является $\|w\|^2$, что усложняет восприятие.

Полученная задача является задачей квадратичного программирования [25], которая после применения метода множителей Лагранжа сводится к задаче о линейной дополнителности [26] и может быть решена либо аналитически методом Лемке, либо итерационно методом внутренней точки [27]. Чаще всего строят двойственную задачу оптимизации и решают её, так как она оказывается легче.

После определения параметров гиперплоскости новые наблюдения X_i классифицируются в зависимости от того, с какой стороны они лежат относительно этой гиперплоскости (достаточно вернуть знак рассчитанного расстояния $h(X_i) = wX_i + b$ для получения прогноза $\{-1, +1\}$).

Для построения нелинейной границы можно использовать нелинейную модель признаков, например полиномиальную $h(X_i) = b + w_1 x_{i,1} + w_2 x_{i,2} +$

$+w_3x_{i,1}^2 + w_4x_{i,1}x_{i,2} + w_5x_{i,2}^2$ или любую другую. Но чаще в методе опорных векторов используют так называемую кернелизацию с помощью ядерных функций $K(X_i, X_j)$, которые показывают, как быстро убывает влияние одной точки на другую (аналог весов во взвешенном скользящем среднем). Чаще всего используют гауссово ядро (радиальные базисные функции) $K(X_i, X_j) = e^{-\gamma\|X_i - X_j\|^2}$, где $\|X_i - X_j\|$ – расстояние между точками, а γ – дополнительный гиперпараметр, влияющий на ширину колокола (разброс). Уравнение границы становится $h(X_i) = \sum_{j=1}^n w_j y_j K(X_i, X_j) + b = 0$, отклонение от которой даёт расстояние $h(X_i)$, по которому можно проводить классификацию. Заметим, что суммирование происходит не по количеству признаков m , а по всем наблюдениям n . Признаки напрямую не используются, а используются лишь для определения расстояния между точками наблюдений X_j и проверяемой точкой X_i . В такой модели $h(X)$ по-прежнему необходимо подобрать значения неизвестных коэффициентов w_j , которых уже будет n штук, и сдвига b . Графическая интерпретация расстояния от границы $h(X)$ (функции решения, сама граница располагается там, где $h(X) = 0$) для двумерного случая при $C = 10$ и $\gamma = 1$ показана на рисунке 1.5.



Источник: составлено автором.

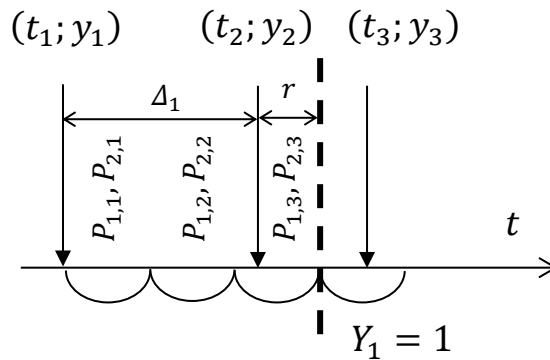
Рисунок 1.5 – Графическая интерпретация расстояния от границы (функции решения) $h(X)$ и сама граница решения $h(X) = 0$

Прогнозирование событий. Пусть имеется выборка событий $(t_i; y_i)$, где t_i – момент времени возникновения события, а y_i – некоторые числовые характеристики этого события. Эти значения дискретны, в промежутках между t_i и t_{i+1} никаких промежуточных значений y_i не наблюдается. Например, это может быть поток событий, как на рисунке 1.1 выше. Дополнительно могут быть известны какие-то сопутствующие индикаторы, которые наблюдаются на протяжении всего времени и представлены в виде временного ряда P_t (индикаторов может быть несколько $P_{1,t}, P_{2,t}, P_{3,t}, \dots$).

Чтобы методами классификации можно было предсказывать события t_{n+1} , необходимо сделать следующую адаптацию. Необходимо разбить ось времени на интервалы постоянной ширины. Выбор ширины интервала (гранулярность) зависит от решаемой задачи и имеющихся данных. Ширина может значительно влиять на качество модели классификации появления событий. Например, можно выбрать неделю.

Далее надо разобраться с тем, что же будет наблюдениями. Каждое наблюдение будет соответствовать своему интервалу с вектором признаков $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ и значением $Y_i \in \{0,1\}$ (которые в методе опорных векторов сами преобразуются в $\{-1,1\}$). В качестве $Y_i = 1$ будет сам факт появления события на интервале, если событие не появилось $Y_i = 0$. Однако, так как модель должна предсказывать появление событий, то значение $Y_i = 1$ должно быть фактом появления события в будущем интервале, а не в том, в котором наблюдаются признаки. Другими словами, признаки, входящие в вектор X_i , должны относиться к предшествующим интервалам (быть лаговыми), по сравнению с интервалом, на который выпало событие, обозначаемое $Y_i = 1$. В итоге самым первым наблюдением не может быть первое событие. Какое-то количество интервалов, как минимум один интервал, должны выступать в качестве лаговых переменных. В качестве признаков можно выбирать значения индикаторов $P_{1,t}, P_{2,t}, \dots$ (если такие есть) на начало интервала или на конец интервала или среднее значение индикатора

за интервал. Можно задействовать признак разницы времени между предыдущими двумя событиями Δ_k , прошедшее время от предыдущего события до начала следующего интервала r . Задействовать характеристики предшествующих событий y_i или $\frac{y_i}{t_{i+1}-t_i}$. Например, в случае использования трёх интервалов для формирования лаговых переменных, рисунок 1.6, вектор признаков может быть $X_1 = (P_{1,1}, P_{2,1}, P_{1,2}, P_{2,2}, P_{1,3}, P_{2,3}, \Delta_1, r, y_2, \frac{y_1}{t_2-t_1})$.



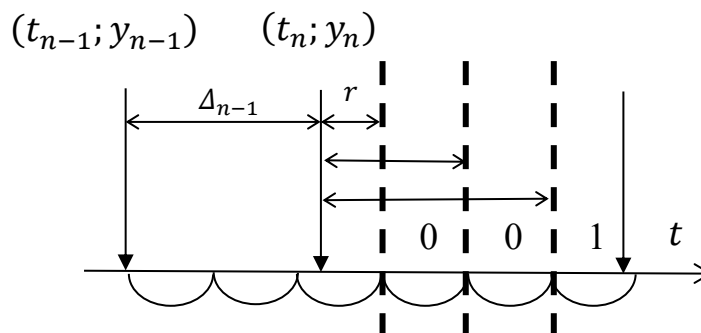
Источник: составлено автором.

Рисунок 1.6 – Пример составления первого наблюдения

Для формирования второго наблюдения X_2 рассматриваем уже следующий интервал и несколько предшествующих ему интервалов, если на следующий интервал не выпало событие, то $Y_2 = 0$. И так далее, пока не дойдём до интервала, на который выпало последнее событие. Все наблюдения X_i формируют матрицу признаков X . Значения Y_i формируют столбец из нулей и единиц. Вообще отбор и составление признаков – это отдельное искусство. В случае задачи прогнозирования событий надо задействовать не только наблюдаемые индикаторы, но и саму выборку событий $(t_i; y_i)$. Конечно же, для способности прогнозирования будущих событий надо использовать именно лаговые значения признаков. Прошлым пытаемся объяснить будущее.

После обучения модели получаем прогноз будущего события следующим образом. От самого последнего интервала откладываем очередной интервал и формируем очередное наблюдение X_i таким же образом, как при формировании обучающего набора. Заметим: часть признаков меняться уже

не будет. В соответствии с обученной моделью получаем прогноз появления события в виде 0 или 1 (или в виде $\{-1, +1\}$). Если получен 0, то события модель не ожидает и надо сдвинуться на очередной интервал, сформировать очередное наблюдение и опять получить прогноз. Если получена 1, то это будет символизировать появление события на соответствующем интервале, что и будет прогнозом будущего события, рисунок 1.7.



Источник: составлено автором.

Рисунок 1.7 – Пример получения прогноза будущего события

Если продолжать делать прогноз события для следующих интервалов, то модель будет выдавать последовательность нулей, за которой следует последовательность единиц, например $(0, 0, 0, 0, 1, 1, 1, 1, \dots)$. Конечно же, последовательность единиц не означает, что на каждом интервале подряд надо ждать событие. Это следует интерпретировать так, что если ранее не было события, то его следует ждать на очередном интервале. В качестве прогноза события следует брать самый первый интервал, на котором модель предсказывает 1. Кстати, модель может дать нулевой прогноз для всех рассматриваемых интервалов, т.е. последовательность только из нулей. Это будет означать, что модель не прогнозирует событие в ближайшую перспективу.

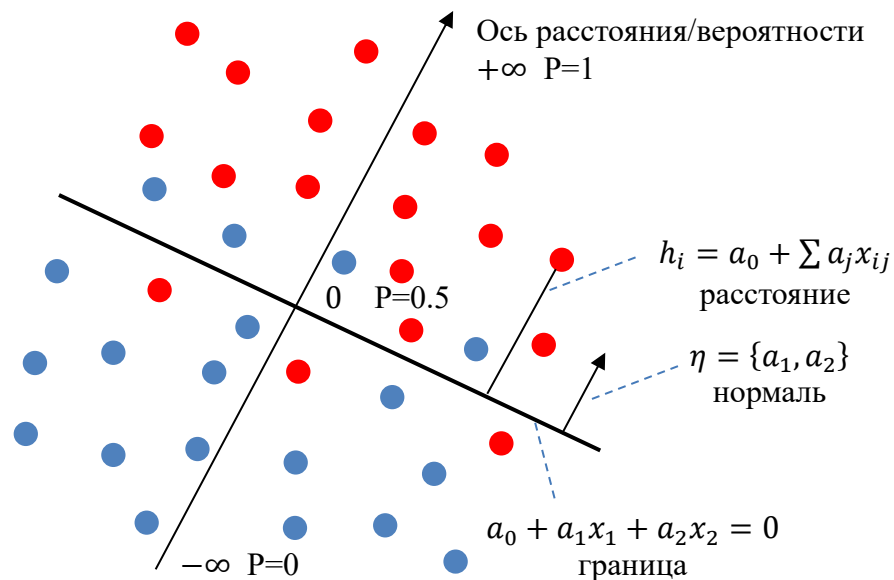
Подробнее получение прогноза методами классификации, в том числе методом опорных векторов, будет рассматриваться в примерах в главе 5.

Заметим, что метод опорных векторов часто критикуют за то, что в исходной реализации он даёт однозначный прогноз (0 и 1), а не рассчитывает вероятности отнесения наблюдений к тому или другому классу. В этом случае

нужны дополнительные манипуляции по конвертации расстояния от границы в вероятность. В следующем методе вероятность принадлежности наблюдения к классу заложена в саму модель метода.

Метод логистической регрессии

Идея логистической регрессии заключается в том, чтобы в многомерном пространстве признаков провести плоскость, которая бы отделяла наблюдения одного класса от наблюдений другого класса. Графическая интерпретация метода логистической регрессии в двумерном пространстве представлена на рисунке 1.8.



Источник: составлено автором как обобщение сути метода, подчерпнутой из разных открытых источников.

Рисунок 1.8 – Графическая интерпретация метода логистической регрессии на плоскости

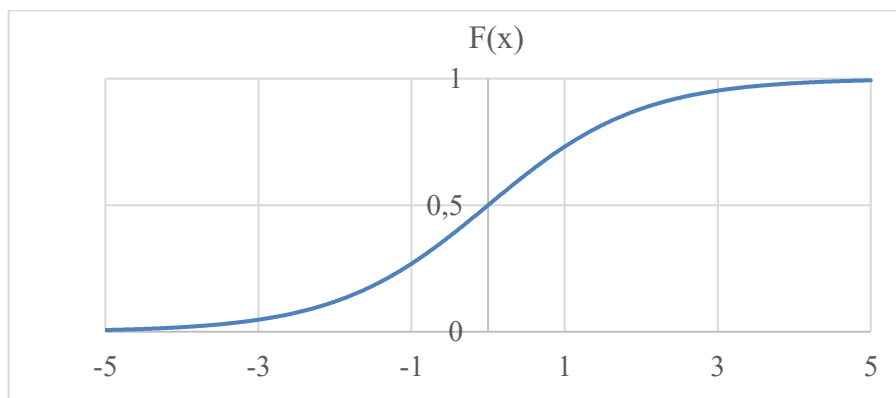
Примечание – Согласно исследованию истории возникновения логистической регрессии [28], осуществленной датским экономистом Жаном Соломоном Крамером, логистическая регрессия (логит-модель) была предложена американским математиком, занимающимся статистикой, Джоозефом Берксоном [29] в 1944 году как более эффективная с точки зрения вычислений модификация пробит-регрессии (пробит-модели), предложенной американским биологом, занимающимся статистическими исследованиями, Чарльзом Иттнером Блиссом [30; 31] в 1934 году.

Для того чтобы провести плоскость (прямую в двумерном пространстве), необходимо определить неизвестные коэффициенты $a_0, a_1, a_2, \dots, a_m$ уравнения $a_0 + a_1x_1 + \dots + a_mx_m = 0$. Нормаль к этой плоскости определяется как $\eta = \{a_1, a_2, \dots, a_m\}$ и показывает, на сколько растёт «расстояние» от границы (т.е. на сколько нарушается уравнение

прямой). В двумерном случае, подставляя координаты x_1 и x_2 конкретного наблюдения в уравнение границы, получаем «расстояние» до границы $h_i = a_0 + \sum a_j x_{ij}$ (на сколько нарушилось ограничение, чтобы получить геометрическое расстояние, надо h_i поделить на $\|a\| = \sqrt{a_1^2 + a_2^2 + \dots + a_m^2}$). Далее это «расстояние» до границы конвертируется в вероятность принадлежности текущего наблюдения к одному из двух классов. Чем больше расстояние от границы, тем больше вероятность принадлежности к заданному классу. Наблюдения, лежащие на самой границе, с вероятностью 0.5 могут принадлежать как к одному классу, так и к другому.

Примечание – Для упрощения формул в вектор признаков добавляют ещё один признак, который для каждого наблюдения всегда принимает значение 1, например для многомерного случая $X_i = (1, x_{i1}, x_{i2}, \dots, x_{im})^T$, тогда расстояние до границы h_i можно рассчитывать как скалярное произведение вектора неизвестных параметров $A = (a_0, a_1, \dots, a_m)^T$ и вектора X_i , т.е. $h_i = A^T X_i = \sum a_j x_{ij}$

Для конвертации этого «расстояния» h_i , которое может быть от $-\infty$ до $+\infty$, в вероятность от 0 до 1 хорошо подходят функции распределения $F(x)$ случайных величин с бесконечной областью определения. В логистической регрессии используется логистическое распределение $F(x) = \frac{1}{1+e^{-(x-m)/s}}$, рисунок 1.9, с нулевым математическим ожиданием $m = 0$ и масштабом s , который связан с коэффициентами a_0, a_1, \dots, a_m уравнения границы (например, уравнения $1x_1 + 1x_2 = 1$ и $100x_1 + 100x_2 = 100$ задают одну и ту же линию, но у второго расстояние h_i получается в 100 раз больше, то есть коэффициенты a_j могут содержать масштабирующий множитель).



Источник: составлено автором по формуле логистического распределения.
Рисунок 1.9 – Функция распределения логистического закона распределения с нулевым математическим ожиданием m и единичным масштабом s

Далее получают и решают оптимизационную задачу

$$\sum_i \ln(1 + e^{-I_i \sum_j a_j x_{ij}}) \rightarrow \min, \quad (1.3)$$

где
$$I_i = \begin{cases} -1, & Y_i = 0 \\ 1, & Y_i = 1 \end{cases}$$

Примечание – Вероятность принадлежности наблюдения к одному из классов можно рассчитать как

$$P(X_i) = P(h_i) = F(h_i) = \frac{1}{1 + e^{-h_i}} = \frac{1}{1 + e^{-\sum_j a_j x_{ij}}}. \quad (1.4)$$

По обучающей выборке наблюдений $X_i = (1, x_{i1}, x_{i2}, \dots, x_{im})^T$, $Y_i \in \{0,1\}$ определяем вероятность принадлежности к классу $p \equiv P(Y = 1|X_i)$, а для наблюдений, не принадлежащих к заданному классу, определяем вероятность $1 - p \equiv P(Y = 0|X_i)$. Выражение для вероятности правильной классификации наблюдения одновременно для двух классов (где $F(x)$ – симметричное распределение)

$$P(Y = Y_i|X_i) = p^{Y_i}(1 - p)^{1 - Y_i} = F(h_i)^{Y_i} (1 - F(h_i))^{1 - Y_i} = F(h_i)^{Y_i} F(-h_i)^{1 - Y_i}. \quad (1.5)$$

Максимизируя по методу максимального правдоподобия, получаем

$$\prod_i F(h_i)^{Y_i} F(-h_i)^{1 - Y_i} \rightarrow \max, \quad (1.6)$$

$$\sum_i Y_i \ln(F(h_i)) + (1 - Y_i) \ln(F(-h_i)) \rightarrow \max. \quad (1.7)$$

Подставляя в полученное выражение формулу логистического распределения $F(x) = (1 + e^{-x})^{-1}$, имеем

$$\sum_i -Y_i \ln(1 + e^{-h_i}) - (1 - Y_i) \ln(1 + e^{h_i}) \rightarrow \max, \quad (1.8)$$

$$\sum_i Y_i \ln(1 + e^{-h_i}) + (1 - Y_i) \ln(1 + e^{h_i}) \rightarrow \min. \quad (1.9)$$

Первое слагаемое обращается в ноль при $Y_i = 0$, а второе при $Y_i = 1$, для каждого наблюдения активно только одно из двух слагаемых. Причём для наблюдений $y_i = 1$ степень у экспоненты отрицательная, а для наблюдений $y_i = 0$ – положительная.

$$L = \sum_i \ln(1 + e^{-I_i h_i}) = \sum_i \ln(1 + e^{-I_i \sum_j a_j x_{ij}}) \rightarrow \min, \quad (1.10)$$

где $I_i = \begin{cases} -1, & Y_i = 0 \\ 1, & Y_i = 1 \end{cases}$, если бы исходный класс наблюдений обозначался как $Y_i \in \{-1,1\}$, то имели бы (1.3).

Оптимизационная задача из-за своей структуры является нелинейной, нахождение искоемых параметров $A = (a_0, a_1, \dots, a_m)$ (обучение модели) производится средствами нелинейного программирования, чаще всего с помощью градиентных методов.

Примечание – Обновление параметров $a_j^{k+1} = a_j^k - \eta \frac{\partial L}{\partial a_j}$, где η – скорость обучения, например

$$0.1. \text{ Производная от функции потерь по параметрам } \frac{\partial L}{\partial a_j} = \sum_i \frac{e^{-I_i \sum_j a_j x_{ij}} \cdot (-I_i x_{ij})}{1 + e^{-I_i \sum_j a_j x_{ij}}} = - \sum_i \frac{I_i x_{ij}}{1 + e^{-I_i \sum_j a_j x_{ij}}}$$

Вероятность принадлежности к одному из двух классов для новых наблюдений (прогноз) можно определить по имеющимся признакам X_i

$$P(X_i) = (1 + e^{-A^T X_i})^{-1}. \quad (1.11)$$

Для задач классификации достаточно проверять, больше ли эта вероятность, чем 0.5, или нет. Но если принимаемые на основе классификации

решения дорогостоящие, то классифицировать можно при больших вероятностях, например 0.8 или 0.9.

Есть другая интерпретация логистической регрессии, которая носит название модели скрытой переменной (латентной переменной). Функция $F(x)$, служащая для преобразования расстояния в вероятность, является функцией распределения. Плотность вероятности логистического распределения имеет симметричную колоколообразную форму с центром около нуля. То есть у нас относительно границы разбросана случайная величина. Но что это за случайная величина? Этой случайной величине можно дать интерпретацию. При отклонении от границы в одну сторону у нас растёт вероятность принадлежности наблюдения к одному классу, следовательно, уменьшается вероятность непринадлежности этого наблюдения к этому же классу. Аналогично, если отклоняться в другую сторону от границы, то уменьшается вероятность непринадлежности к другому классу. Таким образом, относительно границы разбросана случайная величина, символизирующая непринадлежность к любому классу, то есть ошибка классификации. Эта ошибка классификации разбросана в соответствии с выбранным законом распределения: логистическое распределение в логистической регрессии или нормальное распределение в пробит-регрессии.

В том случае, если разброс ошибки классификации уменьшается, то функция сигмоиды $F(x)$ сужается, а при нулевом разбросе сигмоида вырождается в ступенчатую линию (плотность распределения становится импульсной). Такое наблюдается в случае, когда наблюдения можно линейно разделить так, чтобы все наблюдения одного класса гарантированно лежали с одной стороны от границы, а другого класса – с другой стороны (как в методе опорных векторов). Это приводит к тому, что параметры распределения становятся очень большими (коэффициент масштаба s очень маленький), а при нулевом разбросе коэффициенты уходят в бесконечность. Градиентный метод никогда не сойдётся, возникнет ошибка вычислений. В этом случае прибегают к регуляризации, когда в оптимизационную задачу добавляют

слагаемое $\sum_j |a_j|$ или $\|a\|^2 = \sum_j a_j^2$, что заставляет алгоритм минимизации уменьшать также значения параметров

$$\sum_i \ln(1 + e^{-I_i \sum_j a_j x_{ij}}) + \lambda \sum_j a_j^2 \rightarrow \min, \quad (1.12)$$

где λ – коэффициент, показывающий, насколько много внимания надо уделять растущим параметрам по сравнению с основной задачей.

Гиперпараметр λ задаётся исследователем вручную ещё до запуска алгоритма, но существуют и автоматические методы его подбора.

Из-за того, что граница имеет линейный вид (прямая или гиперплоскость), логистическую регрессию относят к методам линейной классификации. Однако граница, разделяющая наблюдения на два класса, может быть нелинейной, например, когда в двумерном пространстве граница принимает вид круга, эллипса или ещё более сложной формы. Например, для эллипса нужно определить параметры уравнения $(x_1 - x_1^0)^2/a + (x_2 - x_2^0)^2/b = 1$. Расстояние до границы уже не будет линейной функцией от входных признаков, но можно прибегнуть к линейаризации. Раскрыв скобки уравнения, будем иметь уравнение вида $a_0 + a_{11}x_1 + a_{12}x_1^2 + a_{21}x_2 + a_{22}x_2^2 = 0$. Если теперь вместо входных признаков $X_i = (1, x_{i,1}, x_{i,2})^T$ использовать $X_i = (1, x_{i,1}, x_{i,1}^2, x_{i,2}, x_{i,2}^2)^T$, то получится по-прежнему использовать линейную форму модели для определения неизвестных коэффициентов. Однако такая линейаризация приводит к мультиколлинеарности, когда объясняющие переменные (столбцы) становятся линейно зависимыми, что сильно влияет на погрешность определения искомых параметров. Если это влияние становится очень сильным, то логистическая регрессия не подходит, должны использоваться более сложные методы нелинейной классификации. Для нелинейного случая также надо отметить то, что расстояние от границы не обязательно будет всегда изменяться от $-\infty$ до $+\infty$. Например, для границы в виде эллипса единичная вероятность может соответствовать точке, лежащей в середине

этой области, т.е. на некотором конечном расстоянии от границы. Тогда вместо логистического распределения, имеющего бесконечную область определения, следует выбирать другие распределения с фиксированной с одной стороны областью определения.

В случае, если в обучающей выборке, по которой подбираются неизвестные параметры уравнения границы, наблюдений одного класса значительно больше, чем наблюдений другого класса, т.е. имеет место несбалансированность выборки, то определяемые коэффициенты будут смещены. В этом случае граница будет расширяться в сторону от подавляющего класса к меньшему классу. Например, когда граница имеет замкнутую область, если увеличить объём обучающей выборки дублированием наблюдений подавляющего класса, то визуально это приведёт к тому, что область с редкими событиями, отделённая границей, сузится. Изменение положения границы, в свою очередь, приведёт к тому, что вероятность классификации наблюдений из меньшего класса будет занижена. А это может иметь серьёзные последствия, так как эти редкие события могут иметь очень высокое влияние, и ошибка их классификации может стоить очень дорого.

Существуют способы для работы с несбалансированными выборками. В работе [32] произведён достаточно обширный обзор существующих методов классификации в условиях несбалансированной выборки. Все методы авторы делят на две группы подходов – внутренние и внешние методы. В группе внутренних методов происходит изменение самого алгоритма классификации. Одним из таких подходов является введение весов для ошибок классификации пропорционально стоимости этой ошибки (редким событиям ставится больший вес). Группа внешних методов исправляет несбалансированность самой обучающей выборки. Такие методы изменяют количество наблюдений в выборке либо дублированием наблюдений, принадлежащих к меньшему классу, либо удалением части наблюдений подавляющего класса наблюдений. Какие из наблюдений удалять или дублировать, можно выбирать либо

случайно, либо по какому-то правилу. Конечно же, можно комбинировать внутренние и внешние методы при работе с несбалансированными выборками, подробнее об этом в [32].

Метод логистической регрессии один из самых популярных, но не единственный метод классификации [33; 34; 35; 36]. Существуют и другие методы.

Прогнозирование событий. Так же как было описано выше (для метода опорных векторов), надо формировать наблюдения в виде векторов X_i и значений $Y_i \in \{0,1\}$, и признаки, входящие в вектор X_i , должны относиться к предшествующим интервалам (быть лаговыми). Отличие будет в том, что метод логистической регрессии способен выдавать вероятности прогноза. Таким образом, последовательность прогноза событий на интервалах может выглядеть как последовательность вероятностей наступления события на соответствующих интервалах (0, 0.05, 0.17, 0.38, 0.51, 0.65, 0.79, 0.9, 0.99). В качестве прогноза события тогда можно брать первый интервал, для которого вероятность появления события будет больше 0,5. Однако, как было сказано выше, из-за несбалансированности выборки граница, разделяющая два класса, может смещаться, что приводит к уменьшению вероятности классификации редких событий. Если не устранять несбалансированность, то следует выбирать меньший порог классификации события.

Метод ближайших соседей

Метод ближайших соседей является простейшим методом классификации. Для задачи классификации метод относит исследуемый объект к тому классу, к которому относится большинство его ближайших соседей (объект, лежащий в корзине с яблоками, скорее всего, тоже яблоко).

Примечание – В привычном виде метод ближайших соседей появился в работе [37], но, по заявлению авторов, правило «ближайшего соседа» появилось в работе [38]. По заверению авторов сайта [39], которые пытались найти создателя метода ближайших соседей, концепция минимального расстояния схожа с концепцией «бритвы Окама» («не следует множить сущее без необходимости») философа-номиналиста Уильяма из Оккама (1285–1349), жившего в XIV веке, однако с этим можно не согласиться. Марселло Пелилло из университета Ка Фоскари в Венеции в работе [40] в специальном исследовании установил, что идеи метода ближайших соседей встречались у египетского математика Ибн аль-Хайсама (латинизированное имя Альхазен), жившего в период с 965-го по 1039 г., в работах по оптике (перевод [42]) – в идеях по визуальному распознаванию предметов.

Под близостью чаще всего понимают евклидово расстояние, но можно использовать другие метрики. Согласно документации самой распространённой библиотеки машинного обучения `scikit-learn` [41], основные метрики могут быть следующими (здесь $X_0 = (x_{0,1}, \dots, x_{0,m})$ – наблюдение, для которого ищутся ближайшие соседи):

- евклидово расстояние: $d_i = \sqrt{\sum_{j=1}^m (x_{i,j} - x_{0,j})^2}$;
- Манхэттенское расстояние: $d_i = \sum_{j=1}^m |x_{i,j} - x_{0,j}|$;
- Чебышево расстояние: $d_i = \max_j |x_{i,j} - x_{0,j}|$;
- расстояние Минковского: $d_i = (\sum_{j=1}^m w_j (x_{i,j} - x_{0,j})^p)^{1/p}$, где p – положительный параметр, w_j – вес признака;
- расстояние Хэмминга: $d_i = \sum_{j=1}^m \mathbb{1}(x_{i,j}, x_{0,j}) / m$, для признаков, принимающих целочисленные значения, где $\mathbb{1}(a, b) = \begin{cases} 1, & a = b \\ 0, & a \neq b \end{cases}$;
- расстояние Джаккарда: $d_i = NNEQ / NNZ$ для логических признаков, где $NNEQ = \sum_{j=1}^m (1 - \mathbb{1}(x_{i,j}, x_{0,j}))$ – количество несовпадающих признаков, NNZ – количество ненулевых признаков, т.е. без признаков, принимающих одновременно ложное значение;
- и др. (в том числе пользовательские метрики).

При использовании одновременно нескольких типов признаков появляется дополнительная трудность с оценкой «близости» наблюдений. Например, при расчёте евклидова расстояния, когда значения одних признаков измеряются миллионами, а другие признаки лишь принимают дискретные значения 0 или 1, нужны соответствующие масштабирующие коэффициенты (выше в метрике Минковского есть веса w_j).

Отбор соседей ограничивается либо максимальным расстоянием, когда находятся все наблюдения ближе, чем заданное расстояние $d_{\text{макс}}$, либо количеством уже найденных соседей.

В методе ближайших соседей, по существу, не обучаются никакие модели. Однако просмотр всей выборки наблюдений может быть трудоёмким. Чтобы не проверять наблюдения, которые лежат слишком далеко, существуют специальные процедуры (KD Trees [43] и Ball Tree [44]) разбиения всего множества наблюдений на подмножества меньшего объёма, в результате чего образуется дерево множеств. Эти приготовления для более быстрого счёта по своему можно назвать «обучением».

Классификация нового наблюдения по методу ближайших соседей выполняется очень просто, достаточно вернуть наиболее встречающуюся метку класса Y_i у найденных соседей (для получения вероятностей достаточно посчитать долю соседей с указанными метками). Метод ближайших соседей также может использоваться для регрессии, когда Y_i являются эндогенными переменными, принимающими произвольные числовые значения. Тогда прогнозное значение определяется как среднее или взвешенное по расстоянию среднее эндогенных переменных Y_i , по множеству K ближайших соседей.

$$Y_0 = \frac{1}{|K|} \sum_{i \in K} Y_i \quad \text{или} \quad Y_0 = \frac{\sum_{i \in K} Y_i / d_i}{\sum_{i \in K} 1 / d_i}. \quad (1.13)$$

Можно изменять способ расчёта расстояния до соседа, как это расстояние потом сравнивается, изменять способ получения прогноза. Иногда адаптируют технологии нечётких множеств [45]. Метод ближайших соседей относят к непараметрическим методам, так как в нём заранее не надо предполагать исходный вид модели, возвращающей прогнозное значение. Однако параметры у метода имеются, это количество соседей. Оптимальное значение количества соседей, как правило, подбирается на практике эмпирическим путем.

Прогнозирование событий. В отличие от всех других методов, этот метод единственный, который при прогнозе событий для последовательности интервалов будет выдавать не упорядоченную последовательность вида $(0, 0, 0, 0, 1, 1, 1, 1, \dots)$, а последовательность, где 0 и 1 могут чередоваться, например

(0, 0, 0, 1, 0, 0, 1, 1, 1). Прогнозом события будем считать тот интервал, на котором появилась первая 1.

Деревья решений

Идея деревьев решений достаточно проста. Ограничимся случаем бинарной классификации одной из самых популярных реализаций CART (Classification And Regression Trees) Лео Бреимана [46]. Имея набор наблюдений $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m}), Y_i \in \{0,1\}, i = 1, \dots, n$, находится простое правило, которое разделяет этот набор на две части, в которых оказывается больше доля наблюдений одного и того же класса, чем до деления.

Примечание – Корни метода деревьев решений непросто проследить. В обзоре 2014 года [47], посвящённом 50 годам деревьев классификации и регрессии, автор утверждает, что первый алгоритм регрессионных деревьев появился в 1963 году в публикации [48] Джеймса Моргана и Джона Сонквиста. Однако, согласно публичному обсуждению среди исследователей, увлекающихся статистикой и машинным обучением [49], первая статья, развивающая подход деревьев решения, за авторством Вильяма Белсона датирована 1959 годом [50]. В том же обсуждении утверждается, что и обычное разбиение на когорты по возрасту и полу, используемое в актуарных таблицах смертности, встречающихся более двух столетий назад, можно отнести к подходу деревьев решений. Очевидно, можно ещё дальше углубляться в историю, так как само мышление часто предполагает простейшее разделение некоторых объектов простыми логическими правилами.

Для деления набора на две части надо определить, какой из признаков j и по какому его значению x_j^* лучше всего разделяет наблюдения. С этой целью используется функция примеси (impurity function) φ , в качестве которой выбирают один из критериев:

- информационная энтропия: $\varphi = -p_0 \text{Log}(p_0) - p_1 \text{Log}(p_1)$;
- джини: $\varphi = p_0(1 - p_0) + p_1(1 - p_1) = 2p_0p_1$;
- ошибки классификации: $\varphi = 1 - \max(p_0, p_1) = \min(p_0, p_1) = \min(p_0, 1 - p_0)$,

где p_0 и p_1 – доли наблюдений заданного класса в полученном делении.

Для каждой части разбиения рассчитывают эту функцию примеси, φ^L и φ^R (левая и правая). Задача заключается в минимизации взвешенной по количеству наблюдений суммы этих функций примеси

$$\frac{n^L}{n} \varphi^L + \frac{n^R}{n} \varphi^R \rightarrow \min, \quad (1.14)$$

где n^L и n^R – количество наблюдений в левой и правой части разбиения.

Примечание – В методе CART на первых страницах [46, с. 27] предполагается искать правила с использованием сразу нескольких переменных, например правила $x_3 + x_5 \leq -2$ и $x_3 + x_5 > -2$. Ниже, на с. 34, предлагается использовать разбиение только с участием одной переменной. На с. 44, 141–143, 181, 182 Бреиман опять возвращается к этой идее, но отмечает, что алгоритм получается сложным в вычислениях, так как целевая функция содержит много локальных оптимумов, и само дерево теряет главное свойство, а именно лёгкость в интерпретации.

Совершив оптимальное разбиение исходной выборки, рекурсивным способом разбивают каждую получившуюся часть на очередные две части тем же способом. Тем самым получается структура дерева. Алгоритм заканчивается, когда в листьях дерева останутся наблюдения только одного класса либо выполняются другие условия остановки. Такими условиями могут быть ограничение на максимальную глубину дерева, на минимальное количество элементов в листьях или на значение функции примеси φ . Другой подход заключается в построении полного дерева, а затем обрезания ветвей, которые очень слабо улучшали качество классификации. Построение неполного дерева препятствует переобучению.

Примечание – Чтобы найти наилучшее разбиение (признак j и его значение x_j^*), количественные признаки сортируются по возрастанию. Середина между двумя соседними значениями становится кандидатом на оптимальное разбиение. Для каждого количественного признака имеется n способов разбиения. Среди возможных разбиений по каждому признаку определяется наилучший, а потом определяется наилучший из наилучших по всем признакам. Для категориальных признаков, принимающих дискретные значения, например $x_6 \in \{a_1, a_2, \dots, a_L\}$, кандидатами в разбиения являются различные неповторяющиеся подмножества, получая условия вида $x_6 \in \{a_2, a_4\}$ и $x_6 \notin \{a_2, a_4\}$, всего $2^{L-1} - 1$ разбиений [46, с. 34]. Но если категорий L слишком много, то предлагается [46, с. 106] проверять принадлежность только к одной категории, а не к подмножеству, тем самым сокращая количество возможных разбиений до L .

Само прогнозирование очень просто, достаточно для нового наблюдения X_i проверить все условия, начиная с корневого узла, и посмотреть, в каком листе окажется данное наблюдение. В качестве прогноза можно вернуть наиболее встречающийся класс наблюдений этого листа либо вернуть доли классов в этом листе, что будет служить вероятностями принадлежности к классу (в регрессионных деревьях возвращается среднее значение эндогенных переменных Y_i наблюдений листа).

Дерево решений сильно зависит от обучающей выборки наблюдений. Если в обучающей выборке наблюдений изменятся или добавятся несколько новых наблюдений, то уже в самом первом узле разделяющее правило может выбраться другим, из-за чего дерево может полностью измениться. Развитием

является использование леса решений [52; 53], когда из исходного набора наблюдений формируется случайная подвыборка, причём некоторые наблюдения могут отобраться повторно. По каждой подвыборке строится своё дерево решений. Сама классификация новых наблюдений происходит агрегированием прогнозов от каждого дерева решений. Однако при использовании леса решений теряется интерпретируемость построенной модели.

Модификаций деревьев решений достаточно много. Интересен подход логистических деревьев [53], когда с помощью логистической регрессии, в которой разделяющая гиперплоскость не обязана быть перпендикулярна одной из осей, строят дерево решений (после каждого деления вновь применяют логистическую регрессию к получающимся частям, получая дерево решений). Правда, такой подход всё же затрудняет интерпретацию модели, так как в каждом узле дерева будет более сложное правило.

Прогнозирование событий происходит так же, как было описано выше. Формируются наблюдения X_i , содержащие лаговые значения признаков, и значения $Y_i \in \{0,1\}$. Для всех последующих интервалов получается прогноз появления событий в виде $(0, 0, 0, 1, 1, 1, \dots)$. Прогнозом будущего события будет тот интервал, на котором появилась первая 1.

Градиентный бустинг

Градиентный бустинг можно описать как алгоритм, который на входных данных применяет один из уже известных методов машинного обучения (например деревья решений), затем рассчитывает остатки для полученной модели и вновь применяет тот же метод, но уже на остатках, при этом модель корректируется (агрегируется аддитивно) как взвешенная сумма всех построенных моделей.

Примечание – Согласно обзору [54] ещё в 1988-м и 1994 годах в работах [55; 56] авторы Кернс и Валиант задаются вопросом: «Можно ли усилить слабый алгоритм обучения, который работает лишь немного лучше, чем случайное угадывание, в сколь угодно точный сильный алгоритм обучения»? На что Шапир в 1989 году в работе [57] и Фроинд в 1995 году в работе [58] дали положительные ответы в виде полиномиальных алгоритмов, которые страдали от определённых недостатков. Выпущенный совместно теми же авторами в 1995 году в работе [59] алгоритм Adaptive Boosting (AdaBoost) решил некоторые недостатки. Джереми Фридмен сперва в виде технического отчёта [60] Станфордского университета в 1999 году и чуть

позже, в 2001 году, в научном журнале [61] показал, что алгоритм AdaBoost является лишь частным случаем более общего подхода, названного градиентным бустингом. Работа [61] считается основополагающей по градиентному бустингу, хотя уже достаточно крепкий фундамент был заложен ранее. Сам Фридмен в своей работе [61] особенно отмечает подход, используемый в лесе, составленном из деревьев решений Бреймана [52]. Сама идея метода очень похожа на то, как по максимуму выжимается информация из «случайных» остатков в авторегрессионных моделях, которыми занимались многие учёные и статистики на протяжении, наверное, большей части прошлого столетия.

В общем виде алгоритм записывается следующим образом:

- задать функцию потерь $L(Y, f(X))$;
- задать начальное приближение модели $f_0(X)$;
- начиная с шага $m = 1$ посчитать остатки (псевдоостатки, псевдоответы, pseudoresponse) как производную от функции потерь при применении модели, полученной на предыдущем шаге

$$\tilde{Y}_i = - \left[\frac{\partial L(Y_i, f(X_i))}{\partial f(X_i)} \right]_{f(X)=f_{m-1}(X)} ; \quad (1.15)$$

- применить какой-нибудь известный метод машинного обучения на этих остатках \tilde{Y}_i , в результате чего будет получена модель $h_m(X)$, объясняющая эти остатки;
- найти оптимальный множитель ρ_m , который минимизирует выбранную функцию потерь от добавления к результатам прогноза по предыдущей модели результатов новой модели

$$\rho_m = \operatorname{argmin}_{\rho} \sum_i L(Y_i, f_{m-1}(X_i) + \rho h_m(X_i)) ; \quad (1.16)$$

- скорректировать модель

$$f_m(X) = f_{m-1}(X) + \rho_m h_m(X); \quad (1.17)$$

- продолжать рассчитывать остатки и корректировать модель, пока не будет достигнуто необходимое требование по качеству.

Тут под корректировкой модели понимается не изменение самой модели, а лишь объединение выходных значений всех промежуточных моделей. Тем самым каждую промежуточную модель $h_m(X)$ необходимо запоминать.

В случае использования обычной квадратичной функции потерь $L(Y, f(X)) = (Y - f(X))^2$ остатки рассчитываются как $\tilde{Y}_i = Y_i - f_{m-1}(X_i)$. В качестве

начального приближения обычно берут константу, минимизирующую функцию потерь $f_0(X) = \gamma = \operatorname{argmin}_\gamma \sum_i L(Y_i, \gamma)$, которая зависит только от значений Y_i и не зависит от X_i .

Для целей бинарной классификации $Y_i \in \{-1, +1\}$ Фридмен предложил использовать лог-правдоподобие

$$f(X) = \frac{1}{2} \log \left[\frac{P(Y = +1|X)}{P(Y = -1|X)} \right] = \frac{1}{2} \log \frac{p}{1-p} \quad (1.18)$$

и функцию потерь (negative binomial log-likelihood, [47, с. 12])

$$L(Y, f(X)) = \log(1 + \exp(-2Yf(X))). \quad (1.19)$$

Примечание – Функция лог-правдоподобия $f(X)$ будет положительной, если модель рассчитывает вероятность положительного класса p больше, чем вероятность отрицательного класса $1-p$. Функция потерь $L(Y, f(X))$ принимает маленькие значения, когда наблюдение Y_i и лог-правдоподобие $f(X_i)$ одного знака, так как экспонента возводится в отрицательную степень. Если наблюдение Y_i и лог-правдоподобие $f(X_i)$ разного знака, то экспонента возводится в положительную степень и функция потерь становится большой.

Расчёт псевдоостатков следующий

$$\tilde{Y}_i = - \left[\frac{\partial L(Y_i, f(X_i))}{\partial f(X_i)} \right]_{f(X)=f_{m-1}(X)} = \frac{2Y_i}{1 + \exp(-2Y_i f_{m-1}(X_i))}. \quad (1.20)$$

В качестве начального приближения $f_0(X)$ берётся лог-правдоподобие

$$f_0(X) = \frac{1}{2} \log \left(\frac{1 + \bar{Y}}{1 - \bar{Y}} \right), \quad (1.21)$$

где \bar{Y} – среднее значение наблюдений Y_i .

Далее предполагается обучить модель дерева решений $h_m(X_i)$ определённой глубины.

После заданного количества M шагов получается хорошее приближение лог-правдоподобия $f_M(X) = \frac{1}{2} \log \frac{p}{1-p}$. Откуда вероятность p принадлежности к положительному классу и вероятность $1-p$ принадлежности к отрицательному классу определяется как

$$p = 1/(1 + e^{-2f_M(X)}), \quad (1.22)$$

$$p = 1/(1 + e^{+2f_M(X)}). \quad (1.23)$$

Примечание – Фридмен предложил дополнительное улучшение. В деревьях решений в качестве значений возвращается среднее арифметическое наблюдений, которые попали в терминальный узел (лист) дерева в результате проверки всех разделяющих правил. То есть, если $X_i \in R_{jm}$, где R_{jm} область, определяемая всеми условиями, располагающимися в узлах выше заданного терминального узла (листа) j , то $h_m(X_i) = b_{jm} = \text{avg}_{X_i \in R_{jm}} \tilde{Y}_i$. Так как на следующем шаге предполагается определять множитель ρ_m , с которым учитывается результат новой модели $f_m(X) = f_{m-1}(X) + \rho_m h_m(X)$, то модель корректируется на $\rho_m b_{jm}$. Тогда вместо того, чтобы искать один коэффициент ρ_m , применяемый к значениям во всех листах, лучше искать отдельные значения для каждого листа γ_{jm} (вместо $\rho_m b_{jm}$)

$$\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{X_i \in R_{jm}} L(\tilde{Y}_i, f_{m-1}(X_i) + \gamma) = \underset{\gamma}{\operatorname{argmin}} \sum_{X_i \in R_{jm}} \log \left(1 + \exp \left(-2\tilde{Y}_i (f_{m-1}(X_i) + \gamma) \right) \right). \quad (1.24)$$

Значение γ_{jm} является константой, которая прибавляется, если наблюдение попадает в лист j . Решение этой оптимизационной задачи Фридмен предлагает искать численным методом Ньютона-Рафсона¹⁾, причём ограничиваясь лишь одной итерацией. В результате приходит к формуле

$$\gamma_{jm} = \frac{\sum_{X_i \in R_{jm}} \tilde{Y}_i}{\sum_{X_i \in R_{jm}} |\tilde{Y}_i| (2 - |\tilde{Y}_i|)}. \quad (1.25)$$

Тем самым дерево строится только для того, чтобы определить области R_{jm} терминальных листов, а значения, которые каждый терминальный лист возвращает, определяются по формуле (1.21). Множитель ρ_m не рассчитывается, целый шаг с оптимизацией пропускается. Обновление модели модифицируется как

$$f_m(X) = f_{m-1}(X) + \sum_j \gamma_{jm} \mathbb{1}(X \in R_{jm}), \quad (1.26)$$

где $\mathbb{1}(X \in R_{jm})$ – функция, возвращающая 1, если наблюдение попало в терминальный узел j , или возвращающая 0 иначе.

В общем виде градиентный бустинг не ограничивается лишь одними деревьями решений, в качестве модели $h_m(X_i)$ может использоваться практически любой известный метод регрессии или классификации.

Прогнозирование событий происходит так же, как было описано выше для других методов.

Классификация искусственными нейронными сетями

Классический алгоритм классификации с помощью обычной многослойной архитектуры нейросети¹⁾ прямого распространения (feed-forward) выглядит следующим образом. Входным слоем сети являются непосредственно m признаков $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ наблюдения. Каждый элемент одного слоя соединён с каждым элементом следующего слоя связью. Связям ставятся веса $w = \{w_{kj}\}$, которые определяются в результате обучения модели. В нейронах следующего слоя происходит суммирование сигналов

¹⁾ Создание нейросетей приписывают Франку Розенблату, разработавшему в 1958 году модель перцептрона [62], хотя к этому времени уже были попытки проводить вычисления на «неорганизованных машинах», основанных на биологических принципах. За подробной историей нейросетей можно обратиться к той же Википедии [63; 64], где опубликован достаточно подробный исторический обзор, содержащий более чем 100 источников.

предыдущего слоя, умноженных на соответствующие веса связей с добавлением соответствующих постоянных значений $b = \{b_j\}$ (сдвигов), которые также определяются во время обучения. В результате на вход нейрона j приходит значение $z_j = [w^T X_i + b]_j = \sum_{k=1}^m w_{kj} x_{i,k} + b_j$. В каждом нейроне j к полученному входному значению z_j применяется нелинейная функция активации $f(x)$. Функций активации $f(x)$ большое количество, например:

- логистическая функция (сигмоида)

$$f(z) = \sigma(z) = \frac{1}{1 + e^{-z}}; \quad (1.27)$$

- линейный выпрямитель (Rectified linear unit, ReLU)

$$f(z) = \max(0, z); \quad (1.28)$$

- софтплюс (гладкий вариант ReLU)

$$f(z) = \log(1 + e^z); \quad (1.29)$$

- гиперболический тангенс

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{2}{1 + e^{-2z}} - 1; \quad (1.30)$$

- экспоненциальная линейная функция (Exponential linear unit, ELU)

$$f(z) = \begin{cases} a(e^z - 1), & z < 0 \\ z, & z \geq 0 \end{cases}. \quad (1.31)$$

Для неглубоких нейронных сетей часто применяют сигмоиду и гиперболический тангенс, для глубоких сетей с большим количеством слоёв чаще выбирают ReLU и ELU [65]. В результате выходом нейрона становится значение $f(\sum_{k=1}^m w_{kj} x_{i,k} + b_j)$, которое подаётся на вход нейронов следующего слоя, домножаясь на соответствующие веса связей. Так происходит до тех пор, пока не будет достигнут последний выходной слой.

На последнем выходном слое N для задачи бинарной классификации может находиться либо один нейрон, либо два нейрона. Если нейрон всего один, то только этот нейрон показывает одновременно вероятность принадлежности к первому классу $p_1 = p$ и вероятность принадлежности к

другому классу $p_2 = 1 - p$. Выходное значение этого единственного нейрона конвертируется в вероятность p от 0 до 1 по логистической функции $p = \sigma(z)$.

Если нейронов два, то каждый нейрон показывает принадлежность только к своему классу (в задачах с большим количеством классов используется больше выходных нейронов). Для конвертации выходных значений этих нейронов в вероятности применяется функция *softmax*, которая возвращает сразу несколько значений

$$\text{softmax}(z_1, z_2) = (p_1, p_2) = \left(\frac{e^{z_1}}{e^{z_1} + e^{z_2}}, \frac{e^{z_2}}{e^{z_1} + e^{z_2}} \right). \quad (1.32)$$

Полученные вероятности p_1 и p_2 можно использовать в качестве прогноза класса нового наблюдения.

Зная истинное значение класса $Y_i \in \{0,1\}$, рассчитывается функция потерь. В качестве функции потерь часто используется функция отрицательного лог-правдоподобия

$$L(p_1, p_2) = -Y_i \log p_1 - (1 - Y_i) \log p_2. \quad (1.33)$$

Примечание – В этой функции потерь одна из частей всегда обращается в ноль, если $Y_i = 1$, то активна первая часть, иначе вторая. Если вероятности p_1 или p_2 малы, то логарифм даёт большие отрицательные значения, которые потом становятся положительными значениями функции потерь. Чтобы предотвратить возможную ошибку расчёта логарифма от нуля, к вероятностям p_1 и p_2 прибавляют фиксированную очень маленькую константу, например $tol = 10^{-6}$.

Нахождение параметров нейросети (весов $w^{(l)}$ и приращений $b^{(l)}$) на каждом слое l осуществляется с помощью градиентного спуска. Для этого от функции потерь $L(p_1, p_2)$ ищутся частные производные по каждому параметру, применяя правило дифференцирования сложной функции (цепное правило в англоязычной литературе). Расчёт начинается с конца и продвигается к началу, способ расчёта называют обратным распространением ошибки.

Примечание – Для случая с одним нейроном в выходном слое с номером N (когда используется функции активации f сигмоида $(z^{(N)}) = \sigma(z^{(N)})$) производная от функции потерь по значению $z^{(L)}$ (суммарного входа) будет

$$\begin{aligned} \frac{\partial L(p_1, p_2)}{\partial z^{(N)}} &= \frac{\partial L(p_1, p_2)}{\partial p_1} \frac{\partial p_1}{\partial \sigma(z^{(N)})} \frac{\partial \sigma(z^{(N)})}{\partial z^{(N)}} + \frac{\partial L(p_1, p_2)}{\partial p_2} \frac{\partial p_2}{\partial \sigma(z^{(N)})} \frac{\partial \sigma(z^{(N)})}{\partial z^{(N)}} = \\ &= -\frac{Y_i}{p_1} \frac{\partial \sigma(z^{(N)})}{\partial z^{(N)}} - \frac{1 - Y_i}{p_2} (-1) \frac{\partial \sigma(z^{(N)})}{\partial z^{(N)}} = \left(-\frac{Y_i}{p} + \frac{1 - Y_i}{1 - p} \right) \sigma(z^{(N)}) = \frac{p - Y_i}{p(1 - p)} \frac{\partial \sigma(z^{(N)})}{\partial z^{(N)}} = \end{aligned}$$

$$= \frac{p - Y_i}{p(1-p)} \frac{\partial \left(\frac{1}{1 + e^{-z^{(N)}}} \right)}{\partial z^{(N)}} = \frac{p - Y_i}{p(1-p)} (p(1-p)) = p - Y_i, \quad (1.34)$$

где $p = \sigma(z) = 1/(1 + e^{-z^{(N)}})$. Производная от сигмоиды $\partial \sigma(z^{(N)})/\partial z^{(N)}$ посчитана отдельно

$$\frac{\partial \left(\frac{1}{1 + e^{-z}} \right)}{\partial z} = -\frac{1}{(1 + e^{-z})^2} \frac{\partial(1 + e^{-z})}{\partial z} = -\frac{1}{(1 + e^{-z})^2} (e^{-z})(-1) = \frac{e^{-z} + 1 - 1}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} \left(1 - \frac{1}{1 + e^{-z}} \right).$$

Для случая использования двух нейронов в выходном слое (когда используется функция $\text{softmax}(z_1, z_2)$) ищутся частные производные по входам $z_1^{(N)}$ и $z_2^{(N)}$, выражения получаются точно такими же, например производная по z_1

$$\begin{aligned} \frac{\partial L(p_1, p_2)}{\partial z_1} &= \frac{\partial L(p_1, p_2)}{\partial p_1} \frac{\partial p_1}{\partial z_1} + \frac{\partial L(p_1, p_2)}{\partial p_2} \frac{\partial p_2}{\partial z_1} = -\frac{Y_i}{p_1} \frac{\partial \left(\frac{e^{z_1}}{e^{z_1} + e^{z_2}} \right)}{\partial z_1} - \frac{1 - Y_i}{p_2} \frac{\partial \left(\frac{e^{z_2}}{e^{z_1} + e^{z_2}} \right)}{\partial z_1} = \\ &= -\frac{Y_i}{p_1} \frac{\partial \left(\frac{e^{z_1}}{e^{z_1} + e^{z_2}} \right)}{\partial z_1} - \frac{1 - Y_i}{p_2} \frac{\partial \left(1 - \frac{e^{z_1}}{e^{z_1} + e^{z_2}} \right)}{\partial z_1} = \left(-\frac{Y_i}{p_1} - \frac{1 - Y_i}{p_2} (-1) \right) \frac{\partial \left(\frac{e^{z_1}}{e^{z_1} + e^{z_2}} \right)}{\partial z_1} = \\ &= \frac{p_1 - Y_i}{p_1(1-p_1)} (p_1(1-p_1)) = p_1 - Y_i, \end{aligned} \quad (1.35)$$

где производная от функции softmax по z_1 посчитана отдельно

$$\begin{aligned} \frac{\partial \left(\frac{e^{z_1}}{e^{z_1} + e^{z_2}} \right)}{\partial z_1} &= \frac{e^{z_1}(e^{z_1} + e^{z_2}) - e^{z_1}e^{z_1}}{(e^{z_1} + e^{z_2})^2} = \frac{e^{z_1}e^{z_2}}{(e^{z_1} + e^{z_2})^2} = \frac{e^{z_1}}{e^{z_1} + e^{z_2}} \left(\frac{e^{z_2}}{e^{z_1} + e^{z_2}} \right) = \\ &= \frac{e^{z_1}}{e^{z_1} + e^{z_2}} \left(1 - \frac{e^{z_1}}{e^{z_1} + e^{z_2}} \right). \end{aligned}$$

Для корректировки весов $w^{(N-1)}$ и приращений $b^{(N-1)}$ предпоследнего $N - 1$ слоя надо рассчитать частные производные от функции потерь, для этого достаточно умножить рассчитанные ранее значения на частную производную от линейной функции $z_k^{(N)} = \sum_{k=1}^m w_{kj}^{(N-1)} f(z_k^{(N-1)}) + b_j^{(N-1)}$:

$$\frac{\partial L(p_1, p_2)}{\partial w_{kj}^{(N-1)}} = \frac{\partial L(p_1, p_2)}{\partial z^{(N)}} \frac{\partial z^{(N)}}{\partial w_{kj}^{(N-1)}} = (p - Y_i) f(z_k^{(N-1)}), \quad (1.36)$$

$$\frac{\partial L(p_1, p_2)}{\partial b_j^{(N-1)}} = \frac{\partial L(p_1, p_2)}{\partial z^{(N)}} \frac{\partial z^{(N)}}{\partial b_j^{(N-1)}} = (p - Y_i), \quad (1.37)$$

где $f(z_k^{(N-1)})$ – выход с нейрона предыдущего слоя (для самого первого слоя входное значение признака $x_{i,k}$).

Для корректировки весов $w^{(N-2)}$ и приращений $b^{(N-2)}$ очередного слоя $N - 2$ нужно рассчитать частные производные функции потерь по выходам слоя $f(z_k^{(N-1)})$ (по функциям активации), потом частную производную от функции активации по входам $z_k^{(N-1)} = \sum_{k=1}^m w_{kj}^{(N-2)} f(z_k^{(N-2)}) + b_j^{(N-2)}$, которая в случае ReLU равна 1 для положительного входа. И опять найти частную производную по весам $w_{kj}^{(N-2)}$ и приращениям этого нового слоя $b_j^{(N-2)}$.

$$\frac{\partial L(p_1, p_2)}{\partial f(z_k^{(N-1)})} = \frac{\partial L(p_1, p_2)}{\partial z^{(N)}} \frac{\partial z^{(N)}}{\partial f(z_k^{(N-1)})} = (p - Y_i) \frac{\partial (\sum_{k=1}^m w_{kj}^{(N-1)} f(z_k^{(N-1)}) + b_j^{(N-1)})}{\partial f(z_k^{(N-1)})} = (p - Y_i) w_{kj}^{(N-1)} \quad (1.38)$$

$$\frac{\partial L(p_1, p_2)}{\partial z_k^{(N-1)}} = \frac{\partial L(p_1, p_2)}{\partial z^{(N)}} \frac{\partial z^{(N)}}{\partial f(z_k^{(N-1)})} \frac{\partial f(z_k^{(N-1)})}{\partial z_k^{(N-1)}} = (p - Y_i) w_{kj}^{(N-1)} \begin{pmatrix} 0, z_k^{(N-1)} < 0 \\ 1, z_k^{(N-1)} \geq 0 \end{pmatrix} \quad (1.39)$$

$$\frac{\partial L(p_1, p_2)}{\partial w_{kj}^{(N-2)}} = \frac{\partial L(p_1, p_2)}{\partial z^{(N)}} \frac{\partial z^{(N)}}{\partial f(z_k^{(N-1)})} \frac{\partial f(z_k^{(N-1)})}{\partial z_k^{(N-1)}} \frac{\partial z_k^{(N-1)}}{\partial w_{kj}^{(N-2)}} = (p - Y_i) w_{kj}^{(N-1)} \begin{pmatrix} 0, z_k^{(N-1)} < 0 \\ 1, z_k^{(N-1)} \geq 0 \end{pmatrix} f(z_k^{(N-2)}) \quad (1.40)$$

$$\frac{\partial L(p_1, p_2)}{\partial b_j^{(N-2)}} = \frac{\partial L(p_1, p_2)}{\partial z^{(N)}} \frac{\partial z^{(N)}}{\partial f(z_k^{(N-1)})} \frac{\partial f(z_k^{(N-1)})}{\partial z_k^{(N-1)}} \frac{\partial z_k^{(N-1)}}{\partial b_j^{(N-2)}} = (p - Y_i) w_{kj}^{(N-1)} \begin{pmatrix} 0, z_k^{(N-1)} < 0 \\ 1, z_k^{(N-1)} \geq 0 \end{pmatrix} \quad (1.41)$$

Аналогичным образом последовательно рассчитываются частные производные для весов и приращений на всех предыдущих слоях.

Градиент функции потерь рассчитывается либо по всей выборке наблюдений как сумма частных производных для каждого наблюдения, либо

только по её части (стохастический градиентный спуск) как сумма частных производных по подвыборке наблюдений. В классическом виде новые значения параметров w_{t+1} получаются добавлением антиградиента $-\left[\partial L / \partial w\right]_t$, умноженного на коэффициент η (эта), определяющий скорость обучения и являющийся гиперпараметром алгоритма

$$w_{t+1} = w_t - \eta \left[\frac{\partial L}{\partial w} \right]_t. \quad (1.42)$$

Существует достаточно большое количество вариаций градиентного спуска [66; 67], где либо скорость обучения η изменяется от шага к шагу, либо накапливается «импульс» как бы при движении в поле тяготения, равному антиградиенту, либо и то и другое одновременно. Это очень объёмный материал, который выходит за рамки этого краткого рассмотрения.

Только что была описана простейшая архитектура нейросети (называемая сетью прямого распространения). Но архитектур нейросетей существует большое количество. Например, рекуррентные сети, когда появляются связи к тем же нейронам, откуда эти связи вышли, или к нейронам на том же слое или на предыдущих слоях. В таких сетях появляется некоторый эффект памяти. Или свёрточные сети, когда по нейронам слоя двигается некоторое небольшое скользящее окно (ядро свертки). Для задачи классификации событий (появления / неоявления) чаще всего применяют обычные сети прямого распространения.

Отдельно стоит сказать, что нейросети иногда применяют для детекции событий [68; 69], когда среди множества изменяющихся непрерывных признаков (временных рядов) находится определённый шаблон, свидетельствующий о наступлении события в настоящем. Это совсем не та задача, которая решается в данной работе. Нас интересует задача научиться по произошедшим событиям (дискретным, имеющим характер потока событий) предсказывать будущие события.

Прогнозирование событий. Так же как у других методов, формируются наблюдения X_i (с использованием лаговых признаков) и

значения $Y_i \in \{0,1\}$. По ним обучается модель. Для будущих интервалов также формируются векторы X_i , которые подаются на вход обученной нейросети. Если нейросеть имеет один выходной нейрон, то этот выходной нейрон сразу даёт вероятность появления события на интервале. Если в выходном слое два нейрона, то надо взять значение только с одного из них. В результате получается последовательность вероятностей, например (0; 0,05; 0,17; 0,38; 0,51; 0,65; 0,79; 0,9; 0,99). Прогнозом наступления события будет первый интервал, где вероятность больше 0,5. Примеры прогнозирования событий методами классификации показаны в главе 5.

1.3 Границы применимости и недостатки методов классификации при прогнозировании событий

Абстрактные признаки вместо событий

Основное ограничение связано с тем, что признаки $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$, которые наблюдаются в настоящем, должны объяснять появление или непоявление события в будущем.

Признаки могут быть сконструированы искусственно. Например, имея поток событий $(t_i; y_i)$ (где y_i – характеристики или воздействие на мир произошедшего события, не путать с фактом возникновения события Y_i на некотором интервале времени), рисунок 1.1 в начале главы, можно посчитать среднюю частоту, среднее время между событиями, сумму событий, суммарное воздействие от событий $\sum_i y_i$. Так, средняя частота может объяснять появление/непоявление события в будущем интервале времени заданной длины.

Однако при конструировании таких искусственных признаков, как, впрочем, и самих объясняемых переменных $Y_i \in \{0,1\}$ (факт появления события), теряются сами события, теряется информация. Событие могло выпасть в первый день интервала времени, а может и в последний день этого интервала. Но факт появления события в обоих случаях обозначается единицей $Y_i = 1$. В то же время событие могло выпасть на следующий день

после границы интервала времени (чуть опоздало/промахнулось), а факт появления события всё равно будет показывать ноль $Y_i = 0$. Потеря информации очевидна.

Сам выбор ширины интервала ничем не регламентируется. Обычно ширина интервала может быть определена через предметную область, когда слишком большой интервал брать бессмысленно, а выбор слишком узкого интервала приводит к тому, что применяемые методы начинают работать крайне плохо, так как будет слишком много нулевых значений. В итоге останавливаются на некотором среднем варианте. В то же время, если задать широкий интервал времени, может получиться так, что в будущем на этот интервал выпадет два события. Но это никак не отразится, прогноз, да и сам факт, будет по-прежнему показывать единицу $Y_i = 1$.

Искусственно сконструированные признаки, такие как частота, среднее время, являются абстрактными индикаторами, описывающими предшествующий поток событий. Индивидуально каждое дискретное событие не получится использовать в виде признаков, так как самих событий может быть разное количество, причём изменяющееся со временем. Тем самым самих предшествующих событий в обучающей выборке быть не может, только сконструированные по ним признаки. При использовании методов классификации размер матрицы признаков остаётся, как правило, одинаковым. Использование абстрактных агрегированных признаков, таких как частота или количество, вместо самих событий – это тоже потеря информации.

Сами идущие вместе с событиями $(t_i; y_i)$ характеристики (значения воздействия на мир) y_i в методах классификации никак не прогнозируются. Чтобы предсказывать числовые значения, нужны другие методы, например регрессионные. О них будет рассказано чуть позже.

Усреднение по наблюдениям перемешивает информацию

Следующая проблема относится не только к методам классификации, но и к другим методам, таким как регрессионные методы.

Практически во всех методах машинного обучения так или иначе заложено усреднение по выборке наблюдений. В методе опорных векторов минимизируется суммарный выход за границу классификации всех наблюдений. В методе логистической регрессии минимизируется суммарное расстояние до разделяющей гиперплоскости (максимизируются вероятности классификации) всех наблюдений. В деревьях решений при формировании разделяющего правила в каждом узле подсчитывается частота наблюдений (вероятность) заданных классов до и после деления, опять же, задействуя все наблюдения в узле. В итоге так или иначе происходит некоторое усреднение определённых качеств по наблюдениям.

Но что является этими наблюдениями? Очень часто в методах классификации (да и регрессии) наблюдениями являются записи о клиентах. То есть по строкам в обучающих матрице X и столбце Y откладываются клиенты, совсем разные клиенты, каждый со своими индивидуальными вкусами и предпочтениями, рисунок 1.10.



Источник: составлено автором по общеизвестной информации, опубликовано в совместной публикации [219].

Рисунок 1.10 – Матрицы, используемые при обучении в методах классификации (и регрессии тоже)

Получается, что усреднение происходит по разным клиентам. Наблюдаемые признаки должны объяснять возникновение/невозникновение события что для одного клиента, что для другого. Например, разделяющая гиперплоскость (в логистической регрессии) строится по отклонениям от этой гиперплоскости точек признаков разных клиентов. Тем самым информация извлекается из совокупности разных клиентов. Построенная таким образом

модель классификации в итоге будет объяснять появление/непоявление событий либо у тех же, либо у совсем других клиентов. В реальности клиенты с близкими по значению признаками могут иметь противоположные предпочтения. Усреднение может перемешать всю информацию от этих разных клиентов. В итоге модель может быть неспособна объяснять, почему для одних клиентов событие происходит, а для других не происходит.

Одним из подходов по устранению этой проблемы является выявление и группировка в отдельные наборы данных схожих наблюдений (клиентов). Далее обучение по каждой группе проводится отдельно. Таким образом якобы решается проблема смешивания информации. Но лишь частично. В каждой группе всё равно наблюдениями являются хоть и схожие, но разные клиенты. Более того, сокращение размера выборки приводит к уменьшению репрезентативности и проблемам оценивания моделей с большим количеством параметров. Само разбиение выборки на группы также является непростой задачей, ошибки на этом этапе будут влиять на дальнейшее исследование групп наблюдений.

Неправильная валидация и метрики качества моделей

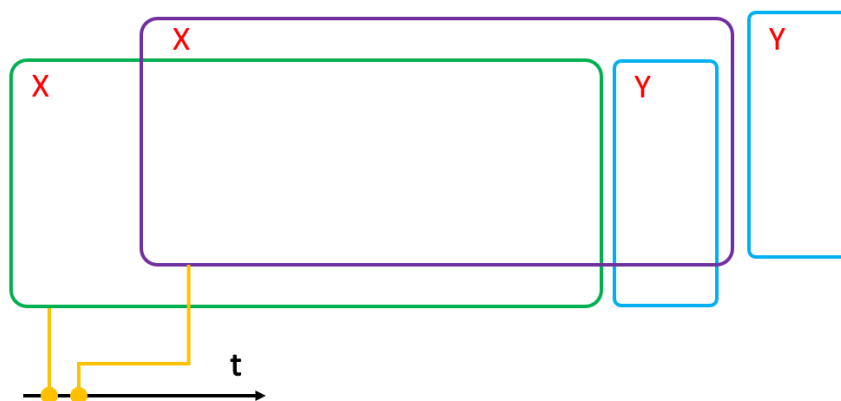
Стандартной и рекомендуемой практикой (в самой популярной библиотеке по машинному обучению `scikit-learn`) является разбиение имеющейся выборки наблюдений на обучающую и контролирующую (тестовую) [70], рисунок 1.11. Обычно в обучающую выборку включается большая часть наблюдений, около 80–90%, а в контролирующую – меньшая часть, оставшиеся 10–20%. Но получается так, что модель, построенная по одним наблюдениям, которыми чаще всего являются клиенты, должна хорошо объяснять наблюдаемые эндогенные переменные других наблюдений (клиентов). Опускаем описанную выше проблему, что модель, построенная по одним клиентам, применяется на совсем других клиентах. Обращаем внимание на то, что валидация в этом случае происходит по наблюдениям, отнесённым к тому же периоду времени, а не по будущим событиям. Ведь исходная цель – научиться объяснять и предсказывать события в будущем.



Источник: составлено автором по общеизвестной информации, опубликовано в совместной публикации [219].

Рисунок 1.11 – Стандартный подход при валидации

Валидация обязательно должна проходить по будущим событиям. Для этого надо разделять выборку на обучающую и контролирующую не по наблюдениям (клиентам) и не в рекомендуемых пропорциях (80 на 20%), а по времени и по всем 100% наблюдений, рисунок 1.12. Другими словами, надо подготовить два периода для расчёта объясняемой переменной Y , первый – для обучения, а второй – для валидации. При этом обучающую матрицу X надо также подготавливать по выборке наблюдений, в которые не включаются наблюдения из последнего контролирующего периода времени.



Источник: составлено автором по общеизвестной информации, опубликовано в совместной публикации [219].

Рисунок 1.12 – Правильная валидация при построении моделей, предсказывающих будущие события

Тут надо обратить внимание на то, что в случае неправильной валидации можно получить очень хорошие метрики качества модели. Но эти метрики качества будут просто некорректны, так как модель проверялась на способность предсказывать события в настоящем, а не в будущем. Даже если

полученная метрика качества будет близка к идеальной, в этом не будет никакого смысла. Обязательно надо правильно подготавливать обучающие и контролируемые наборы данных.

Статические параметры

Даже если разбиение обучающего и контролирующего набора данных произведено правильно, остаётся ещё одна проблема, связанная с самими применяемыми методами классификации (и регрессии). Во многих методах получают статические оценки параметров. То есть для модели будут рассчитаны параметры $a_0, a_1, a_2, a_3, \dots$ (коэффициенты перед наблюдаемыми экзогенными переменными, признаками). Эти значения коэффициентов будут объяснять ту картину мира, которая запечатлена в обучающей матрице, сформированной в предпоследний период времени (последний период оставлен для валидации). Но для последнего момента времени в обучающей матрице будет запечатлена уже другая картина мира. Нет никаких гарантий, что в разные моменты времени параметры $a_0, a_1, a_2, a_3, \dots$ должны оставаться одними и теми же. С большей уверенностью можно сказать, что сами параметры должны меняться со временем, параметры модели $a_0(t), a_1(t), a_2(t), a_3(t), \dots$ должны быть функцией времени.

Может показаться, что включение в обучающую матрицу X признака времени t может решить проблему. На самом деле это не так, параметры $a_0, a_1, a_2, a_3, \dots$ останутся статическими. К ним лишь добавится ещё один параметр, например a_4 (или несколько параметров), который будет показывать, как наблюдаемый признак времени влияет на объясняемую переменную. Вся временная зависимость будет осуществляться через этот дополнительный параметр. Но признаком времени не получится объяснить, как должны были ослабевать или усиливаться со временем влияния других признаков на объясняемую переменную (признак времени будет объяснять наблюдаемую переменную Y_i , а не параметры модели).

Если же в качестве метода выбираются нейросети или градиентный бустинг, то эти проблемы не пропадают. Например, в обученных нейросетях

веса связей будут оставаться статичными оценками w_{ij} , не функциями времени $w_{kj}(t)$. В градиентном бустинге параметр ρ и параметры внутри модели для остатков $h_m(X_i)$ также оказываются статичными.

Дополнительно надо отметить, что модели на основе нейросетей и градиентного бустинга имеют характер чёрного ящика, что делает затруднительным интерпретацию закономерностей.

В итоге для применения методов классификации при прогнозировании событий необходимо убедиться в следующем:

- отобранные или сконструированные признаки содержат информацию о закономерностях появления или не появления событий;
- ширина интервала времени, на котором фиксируется появление или не появление события, выбрана корректно и наилучшим способом соответствует решаемой практической задаче;
- выборка наблюдений, если это возможно, должна быть разделена некоторым способом на несколько выборок, чтобы в процессе обучения информация о закономерностях как можно меньше перемешивалась;
- обучающий и контролирующий набор наблюдений для валидации должен быть подготовлен корректно, чтобы метрики качества модели показывали именно способность предсказывать события в будущем, а не в настоящем.

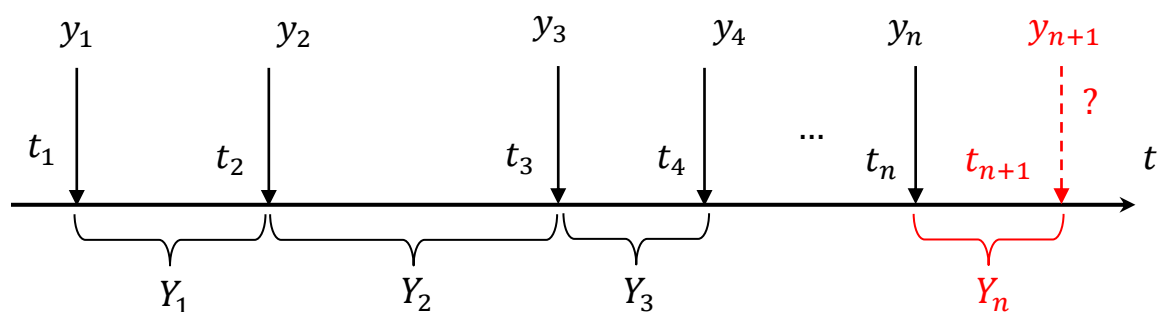
Однако, даже если всё это будет выполнено, сами недостатки, такие как потеря информации, перемешивание информации, статичность параметров и другие, никуда не денутся.

1.4 Методы регрессии в прогнозировании событий

В регрессионных методах по значениям наблюдаемых признаков $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ строятся математические модели $f(X_i)$ в виде алгебраического выражения, описывающего наблюдаемые скалярные величины Y_i , которые принимают уже произвольные числовые значения (а не

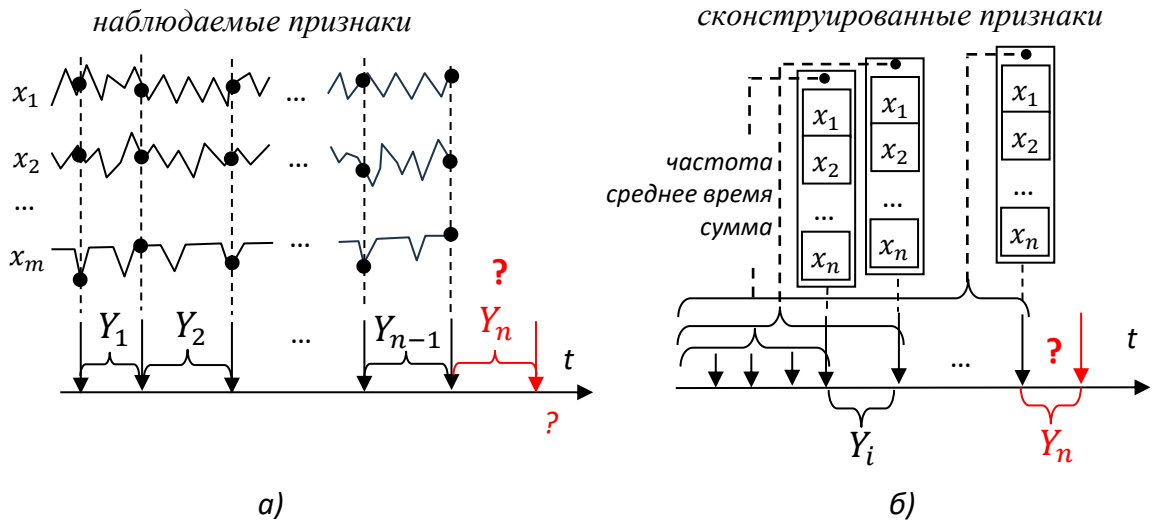
только 0 или 1), $Y_i = f(X_i) + u_i$, где u_i – ошибка (погрешность, остаток, невязка), имеющая случайный характер (случайная величина), как правило, с нормальным законом распределения (с математическим ожиданием 0 и конечной дисперсией). Достаточно часто регрессионными методами исследуются зависимости между количественными показателями, представленными в виде временных рядов.

Для прогнозирования событий, являющихся дискретными по своей природе, регрессионные методы редко используются. Обычно используют теорию потоков событий, о чём будет рассказано ниже. Но и регрессионные методы всё же можно применять при определённой адаптации. Для сведения задачи прогнозирования событий к задаче регрессии в качестве объясняемой переменной Y_i нужно выбрать время между событиями, как пояснено на рисунке 1.13. Математическая модель тогда будет объяснять время между событиями Y_i в зависимости от наблюдаемых признаков X_i . Этими признаками опять могут быть либо фактически наблюдаемые величины, либо искусственно сконструированные, рисунок 1.14. На рисунке 1.14(а) показано, что в момент возникновения очередного события наблюдаются признаки, по которым предсказывается время до следующего события (не стоит путать с распознаванием образов, когда по заданным признакам детектируются события в то же мгновение времени). На рисунке 1.14(б) показано, что в момент возникновения очередного события рассчитываются такие характеристики, как частота, среднее время или сумма событий, или любые другие, которые становятся объясняющими признаками.



Источник: составлено автором.

Рисунок 1.13 – Сведение к задаче регрессии



Источник: составлено автором.

Рисунок 1.14 – Признаки для применения регрессионных методов

Таким образом, глядя на выборку наблюдений признаков X_i , можно попробовать построить модель для времени между событиями $Y_i = f(X_i) + u_i$.

Для предсказания характеристик/воздействий событий y_i нужно строить ещё одну модель зависимости. В итоге будет две зависимости. Первая – зависимость интервалов времени $Y_i = t_{i+1} - t_i$ от признаков X_i . Второй зависимостью будет зависимость самих значений событий $Y_i = y_{i+1}$ от признаков X_i .

Для нахождения неизвестных параметров моделей $f(X_i)$ могут применяться разные методы.

Метод наименьших квадратов для линейных по параметрам моделей

В методе наименьших квадратов для моделей, которые могут быть представлены как линейная комбинация известных значений признаков $x_{i,j}$ и неизвестных параметров a_j

$$Y_i = f(X_i) + u_i = a_0 + a_1 x_{i,1} + a_2 x_{i,2} + \dots + a_m x_{i,m} + u_i, \quad (1.43)$$

ищется минимум суммы квадратов отклонений (ошибок, невязок) u_i

$$L = \sum_{i=1}^n u_i^2 = \sum_{i=1}^n (Y_i - a_0 + a_1 x_{i,1} + a_2 x_{i,2} + \dots + a_m x_{i,m})^2 \rightarrow \min. \quad (1.44)$$

В матричной форме

$$Y = XA + U \quad (1.45)$$

$$L = U^T U = (Y - XA)^T (Y - XA) \rightarrow \min,$$

где

$$Y = \begin{bmatrix} Y_1 \\ \dots \\ Y_n \end{bmatrix}, \quad A = \begin{bmatrix} a_0 \\ \dots \\ a_m \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,m} \\ \dots & \dots & \dots & \dots \\ 1 & x_{n,1} & \dots & x_{n,m} \end{bmatrix}, \quad U = \begin{bmatrix} u_1 \\ \dots \\ u_n \end{bmatrix}. \quad (1.46)$$

Примечание – Согласно историческому обзору [71; 72], метод наименьших квадратов (МНК) был изобретён в самом начале XIX века одновременно тремя разными учёными. В работе 1805 года [73] (некоторые копии выпущены в 1806 году) Анри-Маре Лежандр «опубликовал алгоритм МНК и небольшой пример его реализации, не дав обоснования возникновению идеи метода». Позже Карл Фридрих Гаусс в работе «Теория движения небесных тел...» [74] 1809 года (перевод [75]) вначале даёт вероятностное обоснование МНК, а далее указывает, что «принцип... может рассматриваться следующим образом, даже независимо от теории вероятностей». Несколько ниже, в том же разделе, Гаусс отмечает, что «наш принцип, которым мы пользуемся с 1795 г., ещё недавно был изложен известным Лежандром в его труде [73], где приведено много других свойств этого принципа». То есть в вышедшей позже работе Гаусс утверждает, что пользовался методом МНК ещё до Лежандра. Одновременно с этим в 1808 году Адраин Роберт в работе [76] самостоятельно получает принцип, основанный на минимизации суммы квадратов. Но его работа не была замечена.

Для нахождения оптимума берутся частные производные по неизвестным параметрам $A = (a_0, \dots, a_m)^T$. Выражения для взятия производных в матричной форме следующие (последнее справедливо, когда матрица C симметрична, $C^T = C$).

$$\frac{\partial Cz}{\partial z} = C, \quad \frac{\partial z^T C}{\partial z} = C^T, \quad \frac{\partial z^T Cz}{\partial z} = z^T (C + C^T) = 2z^T C, \quad (1.47)$$

Раскрывая скобки и приравнявая производные от L по A к нулю, находится выражение для получения оценок неизвестных параметров

$$\hat{A} = (X^T X)^{-1} X^T Y. \quad (1.48)$$

Примечание – Расчёт обратной матрицы вносит дополнительные погрешности вычислений, связанные с округлением вещественных чисел. Вычисление оценок \hat{A} следует осуществлять одним из методов решения систем алгебраических линейных уравнений, например методом Гаусса, например на языке R, $\hat{A} = \text{solve}(X^T X, X^T Y)$, на языке Python с помощью $\hat{A} = \text{np.linalg.solve}(X^T X, X^T Y)$.

Ковариационную матрицу оценок можно легко получить, используя выражение в матричной форме ($\text{Cov}(C^T z) = C^T \Sigma_z C$, где $\Sigma_z = \text{Cov}(z)$ – ковариационная матрица для z), если остатки u некоррелированные и имеют постоянную дисперсию ($\text{Cov}(U) = \sigma_u^2 I$), то

$$\text{Cov}(\hat{A}) = (X^T X)^{-1} \sigma_u^2, \quad (1.49)$$

где $\sigma_u^2 = \sum_{i=1}^n u_i^2 / (n - m - 1)$.

Зная $\text{Cov}(\hat{A})$, можно рассчитать погрешность определения прогнозных значений $\hat{Y} = X\hat{A} + U$

$$\begin{aligned}\text{Cov}(\hat{Y}) &= \text{Cov}(X\hat{A} + U) = X\text{Cov}(\hat{A})X^T + \text{Cov}(U) = \\ &= (X(X^T X)^{-1}X^T + I)\sigma_u^2,\end{aligned}\quad (1.50)$$

диагональные элементы которой показывают дисперсию для прогнозных значений \hat{Y}_i в точках наблюдений X_i (недиагональные элементы неинтересны).

В итоге, имея выборку n наблюдений признаков X_i и выборку объясняемых значений Y_i , строится модель $\hat{f}(X)$. Тогда для какого-то нового наблюдения X_{n+1} можно получить прогноз $\hat{Y}_{n+1} = \hat{f}(X_{n+1})$.

Иногда модель с виду может выглядеть нелинейной, но достаточно часто получается привести её к линейному виду с помощью процедуры линеаризации. Например, если модель имеет вид $f(X) = a_0 e^{a_1 x_{i,1} + \dots + a_m x_{i,m}}$, то достаточно взять логарифм от наблюдаемых Y , тогда модель вновь станет линейной по параметрам $\log Y_i = a_0^* + a_1 x_{i,1} + \dots + a_m x_{i,m}$, где $a_0^* = \log a_0$. Однако надо учесть, что в такой модели ошибки u_i включаются мультипликативно $Y_i = f(X_i) \cdot (1 + u_i)$, здесь ошибка u_i выражает относительное отклонение в долях. Тем не менее далеко не для всех нелинейных моделей возможно применение линеаризации, в общем случае модель может иметь совсем произвольный нелинейный вид.

Регрессионные методы для нелинейных по параметрам моделей

В случае, если модель нелинейна по параметрам, то описанный выше аналитический метод получения оценок параметров не подходит. Оценивать параметры модели можно с помощью метода градиентного спуска, минимизируя квадратичную функцию потерь L . Для этого необходимо рассчитать частные производные от функции потерь по параметрам и итерационно корректировать значения параметров в направлении антиградиента. Однако для нелинейных моделей возникает вопрос выпуклости. Для произвольных нелинейных моделей нет гарантий, что

локальный оптимум будет совпадать с глобальным. Вопросы проверки выпуклости нелинейных моделей выходят за рамки рассмотрения работы.

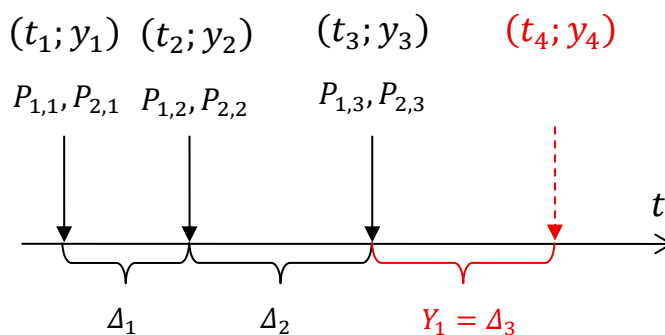
Регрессионные методы не всегда основываются на методе наименьших квадратов. Описанные выше методы, такие как деревья решений, градиентный бустинг, нейронные сети или другие методы машинного обучения, тоже могут строить модели, возвращающие числовые значения. Последнее время стали очень популярны модели, основанные на градиентном бустинге и нейросетях. Например, вместо того, чтобы строить регрессионную модель для времени между событиями и оценивать её параметры методом наименьших квадратов, можно сконструировать и обучить нейросеть, которая возвращает время до следующего события. Или с помощью градиентного бустинга рекурсивно строить модели для остатков, тем самым получая одну большую модель для времени между событиями.

Прогнозирование событий. У нас имеется n событий $(t_i; y_i)$, следовательно, $n - 1$ интервалов между ними, длительность которых как раз будет являться объясняемой переменной $Y_i = \Delta_{i-1} = t_{i+1} - t_i$. Идея использования регрессионных методов для прогнозирования событий заключается в том, чтобы обучить модель, которая смогла бы предсказать длительность интервалов времени между событиями Y_i по наблюдениям X_i . Наблюдениями будут выступать различные признаки, которые известны в момент времени возникновения событий t_i . В качестве признаков можно выбирать значения каких-нибудь индикаторов $P_{1,t}, P_{2,t}, \dots$ (значения каких-то наблюдаемых временных рядов, если такие есть) на момент возникновения событий, разницу времени между предыдущими двумя событиями $Y_{i-1} = \Delta_{i-1} = t_i - t_{i-1}$ (лаговые эндогенные переменные), характеристики предшествующих событий y_i или $\frac{y_i}{t_{i+1} - t_i}$. Однако можно задействовать информацию от нескольких предыдущих событий, например от $k = 2$. Если кроме текущего события использовать информацию от двух предыдущих

событий, рисунок 1.15, вектор признаков для первого наблюдения может быть следующим

$$X_1 = (P_{1,1}, P_{2,1}, P_{1,2}, P_{2,2}, P_{1,3}, P_{2,3}, \Delta_1, \Delta_2, y_1, y_2, y_3, \frac{y_1}{t_2 - t_1}, \frac{y_2}{t_3 - t_2}).$$

Опять заметим, что отбор и составление признаков – это отдельное искусство.



Источник: составлено автором.

Рисунок 1.15 – Пример составления первого наблюдения

Все векторы X_i группируются в одну матрицу X , а длительность интервалов Y_i – в столбец Y . После этого обучается модель на основе одного из регрессионных методов, например с помощью МНК. Для прогноза момента появления будущего события \hat{t}_{n+1} определяем по модели время до следующего события $\hat{Y}_{n-k} = \hat{\Delta}_n$ по наблюдению X_{n-k} (здесь k показывает, сколько предыдущих событий включается в каждое наблюдение, из-за чего индексация сдвигается, в обучающей выборке будет $n - k - 1$ наблюдений), после чего определяем $\hat{t}_{n+1} = t_n + \hat{\Delta}_n$ (заметим, что в зависимости от признаков модель может дать даже отрицательный прогноз интервала времени между событиями $\hat{\Delta}_n$).

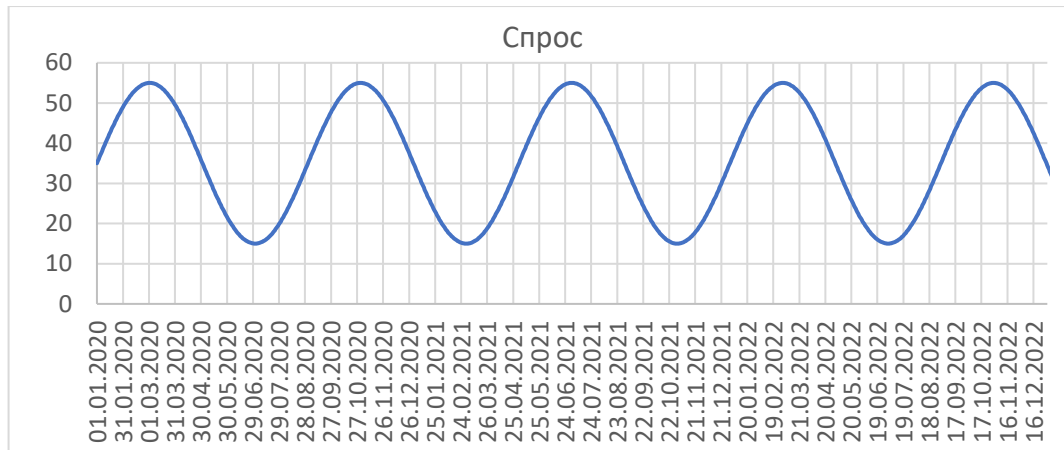
Методами регрессии помимо прогнозирования момента появления будущего события \hat{t}_{n+1} можно спрогнозировать сами характеристики события \hat{y}_{n+1} . Но для этого необходимо строить отдельную модель. Наблюдениями будут все те же векторы X_i , а объясняемой переменной значения $Y_i = y_i$. Это единственное изменение, которое нужно сделать. В итоге будет две отдельные модели: одна – для прогнозирования момента времени, другая – для характеристик событий (в нейросетевых регрессионных моделях, если

использовать два нейрона на выходном слое, то можно обучать модель, которая выдаёт сразу два числовых значения, однако качество может быть хуже, чем если строить две отдельные модели).

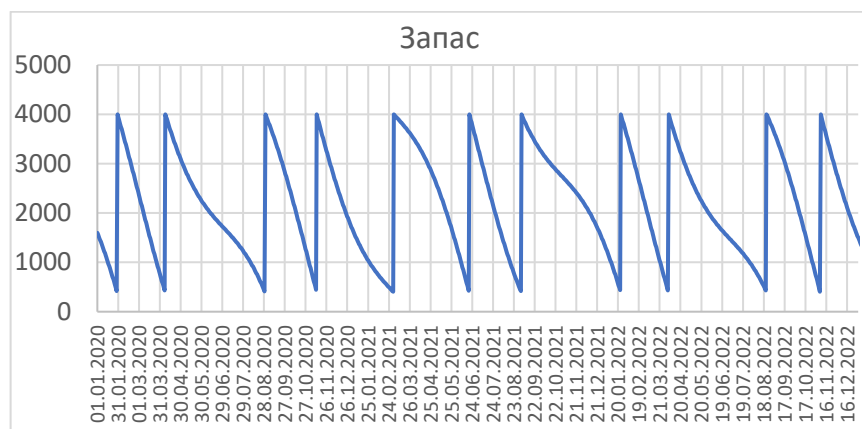
1.5 Границы применимости и недостатки регрессионных методов при прогнозировании событий

Для применения регрессионных методов при прогнозировании событий в первую очередь необходимо, чтобы наблюдаемые признаки $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ были способны объяснить наблюдаемые значения времени до следующего события $Y_i = t_{i+1} - t_i$. Этого может не происходить, нужные признаки могут не наблюдаться или не нести полезную информацию. Например, даже если в качестве одного из признаков используются характеристики событий y_i , которые идут вместе с событиями $(t_i; y_i)$, эти характеристики могут не объяснять интервал времени до следующего события Y_i . Рассмотрим пример.

Пусть наблюдается некоторая модель системы управления запасами. Если задать нестационарный спрос, например в виде гармонической функции (причём даже разброса), рисунок 1.16, задать постоянными максимальный и критический уровень запасов, то наблюдается знакомая пилообразная динамика запасов, рисунок 1.17. Требуется построить модель для интервалов времени между событиями. В этом случае в качестве объясняющей переменной хорошо взять количество потраченных средств или объём покупки. Можно попробовать объяснить интервалы между покупками объёмом совершённой покупки. По-хорошему должна соблюдаться зависимость: чем больше клиент покупает, тем больше интервалы времени между покупками. По крайней мере так учат в логистике в моделях управления запасами. Однако это справедливо лишь при стационарном спросе, но не при нестационарном спросе.



Источник: составлено автором, опубликовано в совместной публикации [219].
Рисунок 1.16 – Нестационарный спрос



Источник: составлено автором, опубликовано в совместной публикации [219].
Рисунок 1.17 – Динамика запасов

Так как в этом примере максимальный и критический уровень запасов не меняется, то будут наблюдаться приблизительно одни и те же объёмы покупок (различия только из-за использования и восполнения страховых запасов, что несущественно). В итоге отчётливо различающиеся разные интервалы между покупками (некоторые отличаются более чем в 2 раза от других) не получится объяснить практически одинаковыми объёмами этих покупок.

Разница в наблюдаемых интервалах времени продиктована именно нестационарным спросом, который не виден со стороны аналитика, наблюдающего события пополнения запасов. Но очевидно, что для предсказания событий потребуется строить модель для ненаблюдаемого нестационарного спроса.

В итоге признаки, которые могут отчётливо наблюдаться и даже следовать вместе с самими событиями, могут не объяснять время между событиями. В этом случае регрессионные методы не подходят. Надо либо искать новые признаки, либо использовать другие методы.

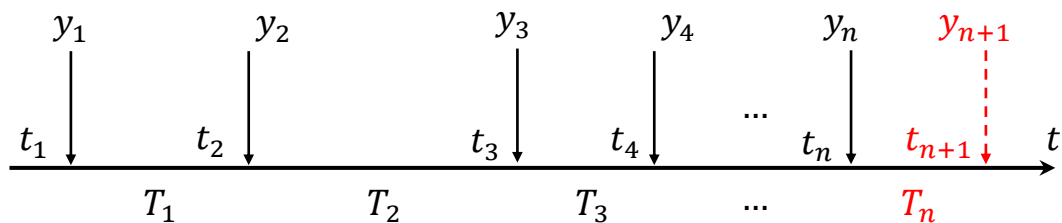
Для регрессионных методов можно выделить те же группы недостатков, что были перечислены для методов классификации:

- при конструировании искусственных абстрактных признаков теряются сами события (вместо того, чтобы учитывать каждое событие, строятся абстрактные агрегированные признаки);
- усреднение по наблюдениям перемешивает информацию (когда наблюдениями являются не каждое событие, а, например, клиенты, обучающая матрица строится по клиентам, модель объясняет последний интервал времени между событиями для каждого клиента, рисунок 1.10, тогда информация от одного клиента смешивается с информацией другого клиента);
- неправильная валидация и ошибочное применение метрик качества модели (когда в качестве контролирующей выборки используется не последнее событие для каждого клиента, а подвыборка самих клиентов, т.е. строк, рисунок 1.11, таким образом, модель предсказывает события в настоящем для других клиентов, метрика качества модели может показывать хоть 100%, но модель будет плохо предсказывать будущие события);
- статичные параметры (оценки параметров модели $a_0, a_1, a_2, a_3, \dots$ или веса w_{kj} являются статичными, а не динамичными, т.е. функциями времени, например, $a_0(t), a_1(t), a_2(t), a_3(t), \dots$ или $w_{ij}(t)$, включение фактора времени t не делает оценки динамичными, просто появляется ещё один параметр, например параметр a_4 , или появляются новые связи с весами w_{kj} , вся временная зависимость будет объясняться этим фактором, другие параметры остаются статичными, хотя их влияние может ослабевать или усиливаться со временем);

– в случае использования регрессионных нейросетей или градиентного бустинга получающиеся модели могут иметь характер чёрного ящика, когда определяемые закономерности получаются неинтерпретируемыми.

1.6 Методы на основе потоков событий в прогнозировании событий

Помимо прогнозирования событий с помощью сведения, к задачам классификации или регрессии существует ещё один подход, основанный на потоках событий из теории случайных процессов [11]. Поток событий представляет собой последовательность моментов времени возникновения событий t_i , графически это выглядит как нанесение точек на ось времени, рисунок 1.18. Отличительной особенностью является то, что события появляются одно за другим через случайные периоды времени T_i , причём вероятность появления одновременно двух событий на достаточно малом интервале времени пренебрежимо мала (ординарность). Если же такие события могут возникать пачками, например, лифт поднял на этаж сразу десяток посетителей, то поток будет неординарным.



Источник: составлено автором по материалам [11].

Рисунок 1.18 – Поток событий

Интервалы между событиями T_i предполагаются независимыми друг от друга, но соответствующими одному и тому же закону распределения. Если закон распределения стационарен, т.е. его характеристики не зависят от времени, то такой поток называют потоком Пальма. Если характеристики потока меняются со временем, то поток нестационарный. В случае, если интервалы времени между событиями T_i соответствуют экспоненциальному закону распределения, то такой поток называют потоком Пуассона

(стационарный поток Пуассона является частным случаем потока Пальма). В пуассоновском потоке количество событий, выпавших на один интервал времени, никак не зависит от того, сколько событий выпало на любой другой не пересекающийся интервал времени. Данное свойство называют отсутствием последействия. Для потоков, обладающих отсутствием последействия, справедливо то, что появление следующего события никак не зависит от того, сколько времени прошло от предыдущего события. В отличие от пуассоновского потока поток Пальма называют потоком с ограниченным последействием, так как вероятность появления следующего события зависит от того, сколько времени прошло спустя предыдущее событие.

Основной характеристикой потоков событий является интенсивность $\lambda(t)$, которая показывает среднее количество событий, возникающих за единицу времени в момент времени t . Для стационарного потока, чьи характеристики не меняются, интенсивность потока постоянна $\lambda(t) = \lambda = \text{const}$ (у пуассоновского потока интенсивность является единственной характеристикой, другие характеристики отсутствуют). Для пуассоновского потока эта же интенсивность λ стоит как степень затухания вероятности в формуле экспоненциального закона распределения для интервалов времени между событиями $f(t) = \lambda e^{-\lambda t}$, $F(t) = 1 - e^{-\lambda t}$.

Примечание – Интенсивность $\lambda(t) = \lim_{\Delta t \rightarrow 0} E[X(t, t + \Delta t)]/\Delta t$, где $X(t, t + \Delta t)$ – количество событий, выпавших на интервал от t до $t + \Delta t$, здесь $E(\cdot)$ – математическое ожидание случайной величины.

Так как интервалы времени в потоках событий являются случайными величинами, то количество событий, выпадающих на заданном периоде времени, также является случайной величиной. Среднее количество событий $X(t, t + \Delta t)$, выпавших на интервал от t до $t + \Delta t$, для стационарных потоков определяется как $E[X(t, t + \Delta t)] = \lambda \Delta t$, а для нестационарных как $E[X(t, t + \Delta t)] = \int_t^{t+\Delta t} \lambda(t) dt$. Для пуассоновских потоков количество событий X , выпадающих на заданном интервале времени, соответствует распределению Пуассона $P(X = x) = a^x e^{-a} / x!$, где a – ожидаемое

количество событий на этом интервале ($a = \lambda\Delta t$ или $a = \int_t^{t+\Delta t} \lambda(t)dt$). Распределение Пуассона является предельным случаем Биномиального распределения, когда вероятность успеха стремится к нулю, а количество испытаний стремится к бесконечности.

Примечание – Биномиальный закон определяет вероятность появления m успехов в серии из n испытаний $P_m = C_n^m p^m (1-p)^{n-m}$, где p – вероятность успеха, а $C_n^m = n! / m!(n-m)!$ – количество сочетаний.

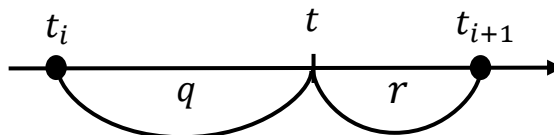
Распределение Пуассона называют законом редких событий, а пуассоновский поток называют потоком редких событий, так как события возникают по одному через ненулевые интервалы времени. Заметим: даже когда время между событиями измеряется микросекундами, пуассоновский поток по-прежнему называют потоком редких событий.

В отличие от пуассоновских потоков для потоков Пальма вероятность появления событий и, следовательно, их количество на заданном периоде времени зависит от времени, прошедшего после предыдущего события. Время до следующего события будет случайной величиной, закон распределения которой рассчитывается следующим образом

$$f_R(r|q) = \frac{f(q+r)}{1-F(q)}, \quad (1.51)$$

где $f_R(r|q)$ – плотность условного закона распределения оставшегося времени r до следующего события при известном значении времени q , прошедшего спустя предыдущее событие;

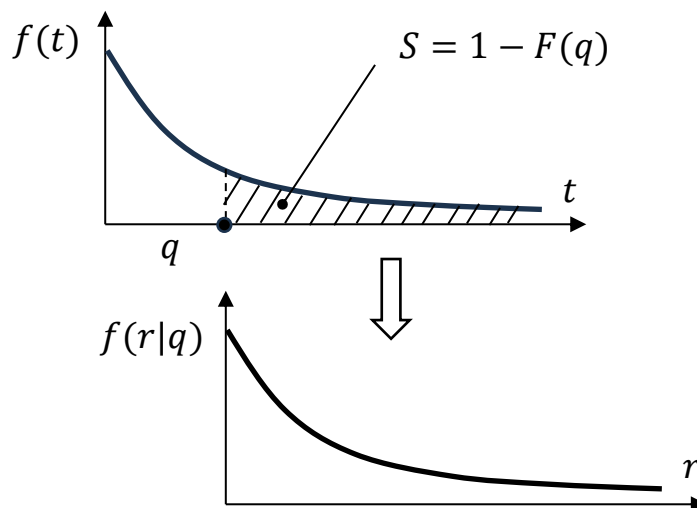
$f(t)$ и $F(t)$ – плотность и (интегральный) закон распределения для самих интервалов времени между событиями, рисунок 1.19.



Источник: составлено автором по материалам [11].

Рисунок 1.19 – Оставшееся время до следующего события r зависит от прошедшего времени после предыдущего события q

У этой формулы (1.51) есть очень простая графическая интерпретация. Она всего лишь показывает то, как выглядит оставшаяся часть распределения (правый «хвостик») после некоторой координаты q , с дополнительным перемасштабированием, чтобы оставшуюся часть можно было вновь назвать законом распределения (делится на полную площадь оставшейся части распределения). Например, для экспоненциального распределения картина будет такой, как на рисунке 1.20. Когда правый «хвостик» экспоненциального распределения делится на число меньше единицы, то функция как бы растягивается по вертикальной оси, и вновь получается экспоненциальное распределение. Свойство отсутствия последействия как раз говорит, что форма закона распределения не меняется, сколько бы времени ни прошло спустя предыдущее событие.



Источник: составлено автором.

Рисунок 1.20 – Закон распределения оставшегося времени до следующего события r не зависит от прошедшего времени q спустя предыдущее событие (для экспоненциального распределения)

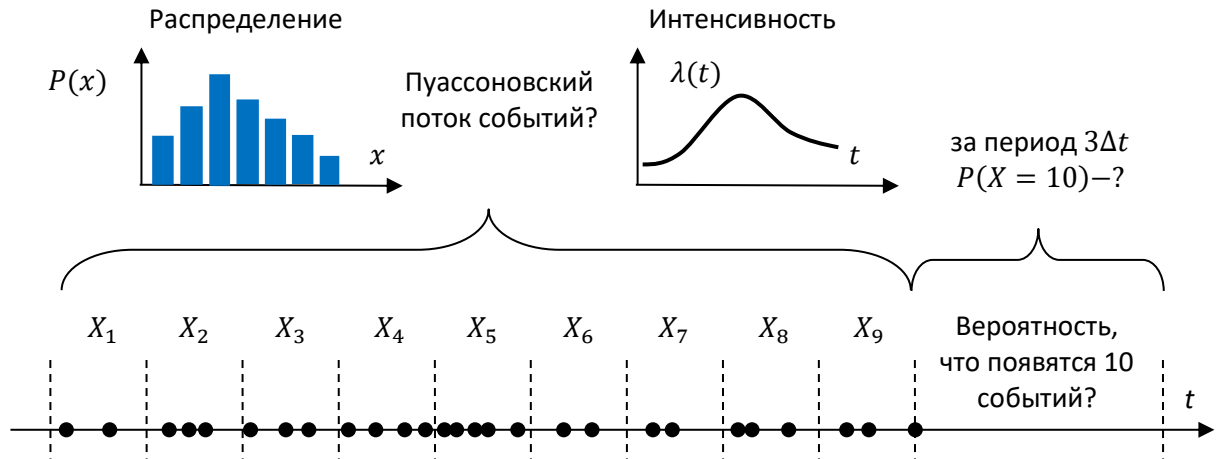
Для других законов распределения форма оставшейся части распределения может значительно отличаться от формы первоначального распределения. Особенно это касается распределений, у которых мода (наиболее вероятное значение) не в нуле, а где-то на положительной оси. Тогда по мере того, как постепенно проходит время q , а событие всё не наступает, наиболее вероятное время наступления события будет постепенно приближаться. Для распределений с ограниченной областью определения,

например для равномерного распределения от 0 до 10 (прямоугольник), по мере того как проходит время, вероятность наступления события в оставшийся промежуток времени будет всё нарастать и нарастать, а сам оставшийся промежуток времени всё сужаться и сужаться. Если же событие так и не настанет, то это будет свидетельствовать, что первоначальное распределение было выбрано неправильно. Формула (1.51) распределения оставшегося времени до события играет ключевую роль в теории надежности [77; 78].

На самом деле теория потоков событий редко применяется для прогнозирования отдельных событий. Намного чаще либо рассчитывается вероятность появления событий в заданный интервал времени, либо прогнозируется количество событий в этот интервал времени, рисунок 1.21. Обратим внимание на то, что представление о самих дискретных событиях как о потоке событий не используется. Используются лишь характеристики пуассоновского закона распределения (хотя закон Пуассона описывает поток дискретных событий).

Примечание – На практике пуассоновское распределение часто применяется для определения размера запасов. В начале определяется интенсивность пуассоновского процесса поступления $\lambda(t)$. В простейшем случае как суммарный спрос за несколько интервалов, поделённый на количество таких интервалов [79; 80; 81], то есть как средний спрос за интервал времени. Такая интенсивность получается стационарной. Для определения нестационарной интенсивности иногда используют скользящее среднее или ядерные функции [82; 83; 84]. Далее, зная значение интенсивности λ (в нестационарном случае надо экстраполировать $\lambda(t)$ на интересующий исследователя период времени), по формуле вероятности для распределения Пуассона $P(x) = a^x e^{-a} / x!$, где a – ожидаемое количество событий на интересующем периоде времени, рассчитывают вероятности возникновения определённого количества событий $P(0), P(1), P(2), \dots$. Тогда размер запаса устанавливается на том уровне, чтобы обеспечить с заданной вероятностью нужное количество товара. Для этого подсчитывают сумму вероятностей $P(0) + P(1) + P(2) + \dots$ до тех пор, пока не будет превышена доверительная вероятность, после чего запас устанавливается на уровне последнего значения, чья вероятность была добавлена в эту сумму последней (запас $S = k: \sum_k P(k) > P_{\text{дов}}$). Если объёмы исчисляются сотнями и тысячами, то расчёт вероятности по распределению Пуассона становится затруднительным, так как там используется факториал. Практичнее тогда переформулировать задачу так, чтобы эти сотни или тысячи стали единицей, и далее решать задачу с этими новыми единицами.

Пуассоновский поток чаще других встречается в научных публикациях. Связано это с предельными теоремами теории потоков, которые говорят о том, что суммарный поток событий, сформированный из произвольных потоков, сходится к пуассоновскому потоку. Или что, если суммарный поток подвергать многократному случайному разрежению, то результирующий поток сходится к пуассоновскому.



Источник: составлено автором.

Рисунок 1.21 – Определение вероятности возникновения событий в заданном промежутке времени

Редко используют другие потоки событий. Иногда используют потоки с интервалами времени между событиями, соответствующими закону распределения Эрланга, которое показывает, как распределена сумма из k экспоненциальных случайных величин. Потоки, в которых время между событиями соответствует распределению Эрланга, также называют прореженными пуассоновскими потоками, когда из потока берётся только каждое k -ое событие. У закона распределения Эрланга уже два параметра – это интенсивность λ суммируемых экспоненциальных величин и количество самих суммируемых величин k (при $k = 1$ распределение совпадает с экспоненциальным, параметр k задаёт порядок распределения). Средний интервал следования событий в потоке Эрланга составляет k/λ , а количество событий за единицу времени λ/k . В работе [85], посвящённой использованию потоков Эрланга для прогнозирования редкого (прерывистого) спроса, средний спрос за единицу времени определяется как $E(y_t) = \lambda\mu/k$, где μ – среднее значение для ненулевого спроса (величины μ и λ/k определяются из данных), также в этой работе определяется дисперсия $Var(y_t)$, но уже более сложным образом. Рассчитав $E(y_t)$ и $Var(y_t)$, можно определить величину запаса, требуемого для удовлетворения спроса на необходимом уровне. В работе [86] также используется распределение Эрланга, но при этом утверждается, что так как оценка количества событий за единицу времени λ/k

определяется через оценку интервалов времени между событиями, оценка спроса за единицу времени $\lambda\mu/k$ получается смещённой и надо вводить корректирующий множитель $\frac{k-1}{k}$ (при небольших значениях порядка распределения Эрланга k эта корректировка может быть значительной).

Авторы этих работ утверждают, что потоки Эрланга более естественны, так как спрос у клиентов может возникать под действием других потоков, что в результате похоже на прореживание потоков, когда k событий одного потока порождают одно событие другого потока. В описанном выше подходе поток событий Эрланга опять используется только для перехода к закону распределения Эрланга и дальнейшему использованию лишь формул закона распределения. Сама же идея рассмотрения потока событий с точки зрения того, что эти события являются дискретными и по-своему уникальными, теряется в обобщающих формулах закона распределения.

Прогнозирование событий. Правильным подходом для прогнозирования момента времени будущего события будет откладывать от момента предыдущего события наиболее вероятное значение (моду) распределения, а не математическое ожидание. У нормального распределения мода совпадает с математическим ожиданием, тогда надо откладывать математическое ожидание. Заметим: для экспоненциального распределения мода в нуле, т.е. откладывать надо 0 единиц времени. Если для экспоненциального распределения отложить математическое ожидание ($m = 1/\lambda$), то это будет некорректно, так как событие, вероятнее всего, возникнет раньше этого времени, а вы будете к этому не готовы.

Сложный вопрос, что надо откладывать от момента предыдущего события для равномерного события. У него нет моды (или их бесконечное количество). С теоретической точки зрения можно откладывать любое значение области определения равномерного закона, так как каждое значение равновероятно. С практической точки зрения всё же лучше отложить

математическое ожидание, т.е. середину отрезка, при этом надо иметь в виду, что событие с той же вероятностью может наступить как раньше, так и позже.

Для многомодовых распределений, у которых несколько пиков, лучше откладывать самую первую моду. В этом случае исследователь будет готов к событию. Но если событие не наступит, то ожидать его следует ближе к следующей моде и т.д.

В случае применения предположения о том, что поток событий является потоком Пуассона, то, как было сказано выше, ожидать событие следует сразу же, через 0 единиц времени. Но если всё же хочется прогнозировать событие с помощью математического ожидания, то надо сначала определить интенсивность потока $\lambda(t)$ с помощью ядерных функций. Далее отложить среднее значение от последнего события $\hat{t}_{n+1} = t_n + 1/\lambda(t_n)$.

Примечание – Ядерные функции используются для получения сглаженного значения в произвольный момент времени t на непрерывной оси времени $\hat{y}(t) = \sum_{i=1}^n y_i \cdot K\left(\frac{t_i-t}{h}\right) / \sum_{i=1}^n K\left(\frac{t_i-t}{h}\right)$, где y_i – наблюдение в момент времени t_i , h – параметр сглаживания (радиус сглаживания), $K(x)$ – одна из ядерных функций, задающая веса наблюдениям. В качестве ядерной функции может использоваться стандартное нормальное распределение $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$. Предложены Надарей и Уотсоном в 1964 году [87; 88].

Если использовать предположение о нормальном стационарном распределении интервалов между событиями (поток Пальма), то достаточно откладывать среднее значение интервалов $\bar{\Delta}$.

Если использовать распределение Эрланга, то надо сначала задать его характеристики. Параметр k можно предположить, например, взяв его равным $k = 2$. Параметр λ распределения Эрланга определяем методом моментов, приравнявая математическое ожидание $m = k/\lambda$ к среднему значению интервалов между событиями $\bar{\Delta}$, откуда получаем $\lambda = k/\bar{\Delta}$. Для получения прогноза момента будущего события откладываем моду $(k-1)/\lambda$ от последнего события $\hat{t}_{n+1} = t_n + (k-1)/\lambda = t_n + \bar{\Delta}(k-1)/k$.

Для других предположений о распределении вероятности интервалов между событиями надо каким-нибудь способом определять его

характеристики по имеющейся выборке событий, после чего откладывать моду от последнего события.

В случае, если известно количество времени, прошедшее спустя предыдущее событие, то можно откладывать от текущего времени моду распределения, сдвинутого на соответствующую величину влево, формула (1.51). Однако такой сдвиг приведёт лишь к тому, что сама мода сдвигается на ту же величину. Результат ожидаемо будет таким же, как если бы не от текущего момента, а от последнего события откладывать моду несдвинутого распределения.

1.7 Границы применимости и недостатки метода, основанного на потоках событий

Потоки событий можно использовать тогда, когда действительно события появляются случайным образом через случайные интервалы времени. В этом случае событие полностью случайное, может выпасть раньше, может выпасть позже рассчитанного срока. Тогда насколько само это прогнозирование становится обоснованным? Если вернуться к приведенному выше примеру из систем управления запасами, рисунки 1.16, 1.17, то очевидно, что, наблюдая лишь сами события пополнения запаса, максимум можно определить закон распределения для интервалов времени между событиями, но никак не получится определить причину, почему событие выпадает раньше или позже среднего значения. А эта причина ведь явно есть в этом примере. Рассмотрение событий как случайных является ограничением своих возможностей.

Если деваться некуда и события действительно абсолютно случайные (о чем спорят философы – есть ли истинная случайность или нет), то тогда остаётся надеяться, что интервалы времени между событиями смогут рассказать о законе распределения между событиями. Изначально предполагая, что имеется пуассоновский поток событий, можно найти лишь вероятности появления заданного количества событий за некоторое время, но

нельзя определить моменты времени возникновения событий. С помощью потоков Пальма уже можно определить моменты времени наступления следующих событий, отложив моду распределения интервалов между событиями от предыдущего события. Однако, так как закон распределения интервалов времени в потоках Пальма стационарен, все будущие события будут идти с одинаковым шагом, что далеко не всегда соответствует действительности. Использование нестационарных потоков с ограниченным последствием, то есть таких, где время между событиями соответствует нестационарному не экспоненциальному закону, для прогнозирования отдельных будущих событий не встречается в литературе. Само определение параметров для нестационарных потоков может быть непростой задачей. Например, когда предполагается, что события происходят через интервалы времени с нормальным законом распределения, лишь из имеющихся данных о произошедших событиях надо определить нестационарные значения как математического ожидания, так и дисперсии для интервалов между событиями (наблюдаемое изменение интенсивности потока может быть связано с изменением дисперсии, а не с изменением математического ожидания, и наоборот).

Также стоит сказать, что в отличие от методов классификации или регрессионных методов, в которых предполагается построение модели по наблюдаемым признакам, в теории потоков событий никакие признаки не используются. Оцениваются лишь параметры закона распределения интервалов времени между событиями, как правило, стационарного. Всё это сильно «упрощает» задачу, в том смысле, что игнорируется какая-либо причина появления событий.

1.8 Другие задачи и методы, связанные с прогнозированием событий

Есть и другие методы, в которых одним из элементов является прогноз событий. Правда, этот прогноз событий не используется напрямую, а

используется в основном, чтобы накопить выборку событий, по которой получаются агрегированные характеристики. У таких методов совсем другая цель – прогноз агрегированных характеристик.

Метод Кростона

В 1972 году Кростон в работе [89] предложил очень простую методику для прогнозирования прерывистого спроса (редких продаж), базирующуюся на жёстких допущениях. Под прерывистым спросом понимается такой спрос, что если представить его в виде временного ряда, то в нём будет подавляющее количество нулевых значений. Само построение временного ряда, где много нулей, но изредка появляются не нули, очень похоже на то, как выглядят данные объясняемой переменной в методах классификации (когда фиксируют факт возникновения события на интервале времени как 0 или 1). Только в отличие от методов классификации ненулевые значения полученного ряда данных являются не единицей, а произвольным положительным числом, обозначающим спрос за соответствующий период времени.

Предполагается, что ненулевой спрос Y_i в каждый интервал наблюдения возникает случайно и независимо с одинаковой вероятностью p (распределение Бернулли). В то же время размер ненулевого спроса распределён по нормальному закону с математическим ожиданием m и дисперсией σ^2 , причём независимо от интервалов времени между ненулевым спросом. Характеристики закона распределения Бернулли и нормального распределения могут быть нестационарные, т.е. p , m и σ^2 могут меняться со временем. Поэтому для определения этих характеристик Кростон предлагает использовать экспоненциальное сглаживание.

Метод заключается в следующем. Исходные данные разбиваются на два ряда: ряд ненулевых значений Y_i и ряд количества интервалов между ненулевым спросом τ_i (для идущих подряд $\tau_i = 1$, через одно $\tau_i = 2$, и т.д.). Эти два ряда сглаживаются экспоненциально описанным ниже способом (коэффициент сглаживания α Кростон рекомендует брать в районе 0.1–0.2), и

получаются ожидаемые значения ненулевого спроса Z_i и количества интервалов между ненулевым спросом T_i

$$Z_i = (1 - \alpha)Z_{i-1} + \alpha Y_i, \quad (1.52)$$

$$T_i = (1 - \alpha)T_{i-1} + \alpha \tau_i. \quad (1.53)$$

Тогда вероятность появления ненулевого спроса определяется как $p = 1/T_i$.

Далее Кростон определяет превышение «ожидаемого» значения спроса над фактическим $e_i = Z_i - Y_i$ (формирует новый ряд), опять сглаживает полученные значения экспоненциально $\sigma_i = (1 - \alpha)\sigma_{i-1} + \alpha|e_i|$ и определяет уровень страховых запасов в трёхкратном объёме $S_i = k\sigma_i$, где $k = 3$. Величина заказа определяется с учётом «ожидаемого» спроса и страхового запаса $R_i = Z_i + k\sigma_i$. Также Кростон вводит специальные условия: $(1 - p)^{t-t_n} < 0.01$ – при получении нулевого спроса сигнализирует, что очень долго нет спроса; $(t - t_n)p < 0.2$ – при получении ненулевого спроса сигнализирует о том, что спрос наступил слишком рано (где $t - t_n$ показывает, сколько единиц/интервалов времени прошло с последнего ненулевого спроса).

Из-за своей простоты метод Кростона стал очень популярным, с момента выхода публикации (1972) до текущего момента количество международных статей, цитирующих эту работу, составляло 429, а запрос в обычном поисковике выдает более 4 млн результатов на различных сайтах. Конечно же, существует множество модификаций данного метода, в работе [90] делается обзор и краткий анализ существующих версий этого метода. Метод Кростона используют до сих пор. Например, в конце 2022 года на конференции по моделированию [91; 92] разработчики из компании SimWell рассказывали о применении метода Кростона в задаче обратной логистики, когда нужно было отправлять грузовик к клиентам, чтобы забрать стеклянную и пластиковую тару на переработку.

Прогнозирование событий. По сути, в методе Кростона прогнозируется событие через T_i (или $1/p$) интервалов времени с ненулевым

значением Z_i , которые определяются с помощью экспоненциального сглаживания. Поэтому сначала надо представить события в виде временного ряда, т.е. разбить ось времени на интервалы и подсчитать количество событий в каждом. Желательно брать малые интервалы, чтобы на интервал попадало не более одного события (например дни). Затем надо рассчитать расстояния между ненулевыми интервалами τ_i (например, количество дней между событиями $\tau_i = t_{i+1} - t_i$). Полученный ряд τ_i сгладить с помощью экспоненциального сглаживания, формула (1.53). Затем отложить последнее значение сглаженного ряда T_i от последнего ненулевого интервала ($\hat{t}_{n+1} = t_n + T_N$, где N – индекс последнего значения сглаженного ряда).

Метод Виллемейна (бутстрэппинга)

В 1979 году Бернанд Эфрон опубликовал в статье [93] новый метод анализа статистической информации и построения доверительных интервалов, а в последующих работах развил его. Метод получил название «Bootstrap», что можно приблизительно по смыслу перевести как «Самовытягивание». Наиболее популярными были работы [94; 95], которые были процитированы огромное количество раз (на момент написания этого текста 2278 и 3817 соответственно) в научных статьях, а в Интернете поиск выдает 9 млн результатов. Наиболее полно метод представлен в монографии [96], в которой метод представлен с разных сторон.

Метод преимущественно предназначен для построения более точного доверительного интервала для некоторой интересующей исследователя характеристики θ . Вместо того, чтобы откладывать от оценки среднего значения $\hat{\theta}$, оценённой по некоторой выборке $X = \{x_1, \dots, x_n\}$, заданное количество оценок среднего квадратичного отклонения $\hat{\sigma}$ (например, $t_{кр}$ – критическое значение распределения Стьюдента для установленной доверительной вероятности), в результате чего получается симметричный доверительный интервал, метод предполагает получить из имеющейся выборки множество случайных выборок $X^* = \{x_1^*, \dots, x_n^*\}$ с помощью

случайного выбора элементов из исходной выборки, причём элементы могут выбираться повторно. По каждой такой выборке X^* определяется соответствующая оценка θ^* интересующей исследователя характеристики. Количество случайным образом сформированных выборок X^* берётся как можно больше, например 100 000. Полученные значения оценок θ^* сортируются по возрастанию (по сути, строится эмпирическая функция распределения $G(\theta)$). Наконец, доверительный интервал для доверительной вероятности $1 - 2\alpha$, например для вероятности 95%, определяется как интервал между значениями, стоящими на позиции, определяющими, что доля α всех значений оказались меньше и больше соответственно (т.е. $\theta \in [G^{-1}(\alpha); G^{-1}(1 - \alpha)]$).

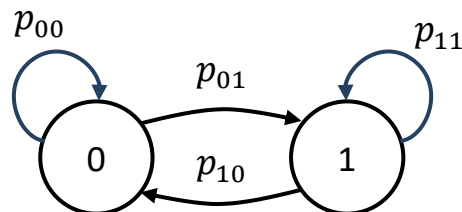
Получающийся доверительный интервал хорош тем, что он может учитывать асимметрию, в отличие от доверительного интервала на основе критерия Стьюдента. В случае, если исходная выборка была репрезентативна (т.е. характеристики, полученные в результате выборочных наблюдений, соответствуют характеристикам всей генеральной совокупности [97]), набор бутстрэп оценок, полученных по случайным выборкам, также будет репрезентативным. Вариация оценок при увеличении количества случайных выборок X^* асимптотически сходится к вариации оценки, заключённой в исходной выборке. Однако, если в исходную выборку попал какой-то выброс, то при построении бутстрэп оценок этот выброс возьмётся с той же вероятностью, что и любое другое наблюдение, что приведёт к размножению этого выброса и в результате к смещению границ доверительного интервала. Метод бутстрэппинга очень удобен, и его легко реализовать, его можно применить к любому набору данных, не беспокоясь о выполнении предположения о нормальности.

В 2004 году в работе [98] Томас Виллемейн адаптировал этот метод для прогноза прерывистого спроса и внёс несколько модификаций. Виллемейн постарался исправить недостаток, присущий как методу Кростона, так и обычному методу бутстрэппинга при прогнозировании спроса, а именно

неспособность учитывать возможную корреляцию между следующими нулевыми и ненулевыми значениями. Метод, предложенный Виллемейном, предполагал сначала оценить вероятности переходов для марковского процесса с двумя состояниями – нулевой и ненулевой спрос, рисунок 1.22.

Другими словами, вначале требуется оценить следующие вероятности:

- появления ненулевого спроса, если предыдущий спрос был нулевым;
- появления нулевого спроса, если спрос был нулевым;
- появления ненулевого спроса, если предыдущий спрос был ненулевым;
- появление нулевого спроса, если спрос был ненулевым.



Источник: составлено автором по материалам [98].

Рисунок 1.22 – Марковский процесс с двумя состояниями – нулевой и ненулевой спрос, определению подлежат вероятности перехода между состояниями

Далее с рассчитанными вероятностями генерируется по указанной схеме марковского процесса набор случайных величин, причём количество генерируемых случайных величин ограничено временем исполнения заказа, например на три месяца вперёд. Следующим шагом сгенерированные ненулевые элементы заменяются на значения ненулевого спроса, выбранные случайным образом (равновероятно) из имеющейся исходной выборки.

Затем Виллемейн, рассматривая основной недостаток метода бутстрэппинга, а именно возвращать только те значения, которые есть в выборке, предложил на ненулевые значения дополнительно накладывать небольшой разброс по нормальному закону распределения.

$$X^{*'} = 1 + \lfloor X^* + z\sqrt{X^*} \rfloor, \quad (1.54)$$

где X^* – выбранное из выборки ненулевое значение;

z – сгенерированная стандартная случайная величина;

$\sqrt{X^*}$ – задаёт разброс новой случайной величины.

Причём, если новая величина $X^{*'}$ получается отрицательной, то разброс убирается $X^{*'} = X^*$. Данная процедура в отличие от обычного метода бутстрэппинга гарантирует, что случайным образом отобранные величины получаются уникальными.

Следующим шагом подсчитывается суммарный сгенерированный спрос на интересующем исследователя интервале времени $\theta^* = \sum_i X^{*'}_i$, например за три месяца. Затем вся процедура повторяется многократно, например 100 000 раз, чтобы получить объёмную выборку из значений суммарного сгенерированного спроса θ^* . Наконец, по выборке значений θ^* о суммарном спросе строится доверительный интервал для заданной доверительной вероятности.

С помощью этой процедуры можно легко определять объём заказа с необходимым уровнем удовлетворения спроса (уровень сервиса). То есть, чтобы удовлетворить 99% спроса, достаточно установить объём заказа как 99% квантиль функции распределения случайной величины θ^* (т.е. $G^{-1}(0.99)$). Проверая точность предложенного метода на реальных данных о прерывистом спросе в промышленности, Виллемейн определил, что его метод значительно превосходит по точности обычное экспоненциальное сглаживание и метод Кростона. Причём на реальных данных оказалось, что метод Кростона был иногда даже хуже, чем обычное экспоненциальное сглаживание, хотя в предыдущей статье [99] метод Кростона оказался лучше. В недавних работах [100; 101] делается широкий обзор множества подходов для прогнозирования или работы в условиях прерывистого спроса.

Прогнозирование событий. По сути, из этого метода нужно взять только шаг, где оцениваются переходные вероятности марковского процесса P_{00} и P_{01} (переходные вероятности P_{10} , P_{11} не будут нужны). Для этого надо опять разбить ось времени на интервалы Δ , например шириной в день. Далее подсчитываем количество переходов n_{00} (из состояния 0 в 0) и n_{01} (из 0 в 1), т.е. сколько раз после нулевого интервала вновь шёл нулевой интервал и

сколько раз после нулевого интервала шёл ненулевой интервал. Рассчитываем переходные вероятности $P_{00} = n_{00}/(n_{00} + n_{01})$ и $P_{01} = n_{01}/(n_{00} + n_{01})$. Далее надо рассчитать $\bar{t}_{0,1}$ среднее количество шагов до первого перехода из состояния 0 в состояние 1. Так как у нас всего два состояния, расчёт достаточно простой. Вероятность первого перехода на первом шаге равна P_{01} . На втором шаге – $P_{00}P_{01}$, третьем шаге – $P_{00}^2P_{01}$, четвёртом – $P_{00}^3P_{01}$ и т.д. Умножая эти вероятности на количество шагов и складывая, получаем среднее время первого перехода $\bar{t}_{0,1} = 1 \cdot P_{01} + 2 \cdot P_{00}P_{01} + 3 \cdot P_{00}^2P_{01} + \dots$. Тут можно выделить ряд вида $a_n = nq^n$, бесконечная сумма которого будет $\frac{q}{(1-q)^2}$.

Тогда с учётом, что $P_{00} + P_{01} = 1$, формула упрощается до тривиальной

$$\bar{t}_{0,1} = \frac{P_{01}(1 \cdot P_{00} + 2 \cdot P_{00}^2 + 3 \cdot P_{00}^3 + \dots)}{P_{00}} = \frac{P_{01}P_{00}}{P_{00}(1 - P_{00})^2} = \frac{1}{P_{01}}. \quad (1.55)$$

Примечание – Вероятность первого перехода из состояния i в состояние j за k шагов $\hat{p}_{ij}^{(k)}$ для произвольной схемы марковской цепи определяется по рекурсивной формуле $\hat{p}_{ij}^{(k)} = \sum_{m \neq j} p_{im} \hat{p}_{mj}^{(k-1)}$, где $\hat{p}_{ij}^{(1)} = p_{ij}$ совпадает с переходной вероятностью. Среднее количество шагов $\bar{t}_{i,j}$, необходимых для первого перехода из состояния i в состояние j , определяется по формуле $\bar{t}_{i,j} = \sum_{t=1}^{\infty} t \cdot \hat{p}_{ij}^{(t)}$.

Прогноз получаем, откладывая от последнего события ожидаемое число интервалов $\hat{t}_{n+1} = t_n + \frac{1}{P_{01}} \cdot \Delta$, где Δ – ширина интервалов времени, например день.

Границы применимости и недостатки методов Кростона и Виллимейна

Описанные выше два метода предназначены в основном для прогнозирования агрегированных характеристик, таких как суммарный спрос за интересующий период времени, т.е. сразу многих событий. Для прогнозирования отдельных событий такие методы не предназначались. Тем не менее могут быть интересны принципы, заложенные в этих методах.

Существенным ограничением является то, что события предполагаются случайными. В методе Кростона событие возникает по схеме испытания Бернулли с оценённой вероятностью p . Событие либо появилось, либо не появилось. Анализа причин возникновения не происходит. Отдельной

критике подвергается предположение о независимости наблюдаемых значений спроса. Если событие не выпало в прошлый период, то вероятность выпадения в следующем периоде не изменяется. Во время экспоненциального сглаживания не происходит никакого поиска закономерностей, которые, безусловно, должны быть. Неправильно предполагать, что ненулевое значение появится через ожидаемое количество нулевых значений.

В методе Виллимейна события ненулевого спроса моделируются марковской цепью с помощью разыгрывания случайных чисел. Имея оценённые вероятности перехода между состояниями, по сути, бросается кубик, чтобы определить, произойдёт событие или нет. Разве можно броском кубика прогнозировать события? Даже если из переходных вероятностей марковского процесса рассчитать среднее количество шагов для перехода в нужное состояние, это всё равно не будет корректным прогнозом, так как на каждом шаге происходит бросок кубика. Событие может появиться сразу, на первом броске. В формуле расчёта среднего количества шагов используется бесконечное число бросков. Также переходные вероятности не меняются со временем (параметры статичные, а марковский процесс однородный).

Тут то же самое нужно сказать про все другие модели, в которых появление события основано на разыгрывании случайных чисел. Например, при использовании имитационного моделирования в дискретно-событийных моделях появление событий, как правило, реализуется с помощью разыгрывания случайной величины, имеющей экспоненциальный закон распределения, используя генератор случайных чисел. Но моделирование случайных событий броском кубика не является прогнозированием событий.

Примечание – Генерация экспоненциальных случайных величин происходит по формуле $X = -\log(1 - U) / \lambda$, где U – равномерная случайная величина, выданная генератором случайных чисел, λ – интенсивность потока событий.

Заметим: есть ещё направления, схожие по названию с задачей прогнозирования редких событий. В процессах Леви и теории больших прыжков [102; 103; 104] моделируется такое случайное блуждание, когда

направление движения выбирается случайно, но время движения по выбранному направлению моделируется распределением с тяжёлыми хвостами (вероятность больших отклонений падает медленно, медленнее, чем в экспоненциальном законе). В результате именно редкий, но большой прыжок доминирует в расчётах координат. В такой теории не предсказываются отдельные события, а определяются законы распределения для суммарного движения. В теории экстремальных событий (значений или катастроф) [105] события моделируются пуассоновскими потоками, но каждое событие несёт ущерб, который отдельно моделируется также распределением с тяжёлыми хвостами. В итоге редкие катастрофы с большим ущербом составляют значительную долю в общей сумме ущерба всех событий. В этой теории отдельные события также не прогнозируются, в ней определяют квантили распределений для суммарного ущерба. В итоге процессы Леви, теория больших прыжков и теория экстремальных событий не рассматриваются в данной работе, посвящённой предсказанию в первую очередь отдельных конкретных событий.

Если интересует вопрос, какой из существующих методов следует выбирать для прогнозирования событий, то ответом будет следующее. Если достаточно прогнозировать факт появления события на интервале, причём неважно, когда именно на этом интервале событие появится, то можно использовать методы классификации. Если нужно прогнозировать сам момент времени появления событий, то выбор сложнее. Если имеются наблюдаемые признаки, по которым можно судить о появлении событий в будущем, то следует использовать регрессионные методы. Если имеется только выборка событий, т.е. поток событий, то следует использовать соответствующие методы из теории потоков событий. Конечно же, в современную эпоху принято применять не один метод, а множество методов, из которых выбирать лучший (по способности прогнозировать события из тестовой выборки).

В следующих главах представлен новый подход прогнозирования событий, основанный на восстановлении механизмов их образования,

точность которого зачастую превосходит современные методы машинного обучения.

Выводы по главе 1

1) Даны определения терминам «событие» и «редкое событие». Событие – факт или явление, которое происходит в определённый момент времени как результат проявления экономических, политических, социальных, физических и других процессов, протекающих в той или иной сфере. Термин «редкое событие» является качественным проявлением количества событий или вероятности их появления в контексте решаемой исследователем задачи. Редкие события – события, которые появляются в среднем в количестве два и менее за период требуемой гранулярности. Гранулярность данных – требуемая степень детализации данных, которая может существенно влиять на точность, надёжность и удобство использования прогнозов. В работе термин «редкость» акцентирует внимание на дискретности событий и на форме представления в виде потока событий. Таким образом, термин «редкие события» говорит о представлении событий в виде потока событий. Но если говорить в контексте неожиданных/значимых/удивительных событий, то редкие события – это часть от событий, которые в результате работы механизма приобретут неожиданные (для стороннего наблюдателя) значения.

2) Исследована возможность и важность прогнозирования отдельных событий в изменяющейся парадигме от непрерывных функций к дискретным событиям. Обоснована важность прогнозирования событий в качестве средства предиктивной аналитики, применяемой в бизнесе, и в качестве части системы раннего предупреждения, оповещающей о приближении кризиса.

3) Проведён анализ существующих методов классификации (метод опорных векторов, логистической регрессии, ближайших соседей, деревьев решений, градиентного бустинга, нейронных сетей), показано, как их применять в прогнозировании событий. Определены особенности, границы

применимости и недостатки методов классификации для прогнозирования событий.

4) Проведён анализ методов регрессии для прогнозирования событий, показано, как их применять в прогнозировании событий, определены их особенности, границы применимости и недостатки.

5) Проведён анализ методов на основе потоков событий, показано, как их применять в прогнозировании событий, определены их особенности, границы применимости и недостатки.

6) Проведены анализ и адаптация для прогнозирования событий методов Кростона и Виллимейна. Определены их особенности, границы применимости и недостатки.

7) Недостатки существующих методов следующие: неспособность работать с событиями напрямую и необходимость конструировать из событий абстрактные признаки; потеря информации из-за неспособности работать с событиями напрямую; перемешивание информации из-за заложенных в методах операций усреднения; дополнительные трудности при валидации моделей; статичность параметров (влияние факторов не меняется со временем); невозможность применения методов, когда наблюдаемые признаки не несут информацию о появлении событий или когда такие признаки вовсе отсутствуют; модели зачастую получаются не интерпретируемыми (чёрный ящик), что делает использование прогноза ненадёжным; рассмотрение событий как случайных неспособно объяснить, почему событие отклоняется раньше или позже своего ожидаемого срока; у некоторых законов распределения мода либо в нуле, либо не существует, что делает затруднительным предсказание будущего события; потоки Пальма всегда стационарные, и они неспособны описать динамические изменения; потоки событий игнорируют причины появления событий; в методе Кростона или Виллимейна событие возникает по схеме испытания Бернулли, по сути «броском кубика», что не является корректным прогнозом. Всё это заставляет искать более совершенные способы прогнозирования событий.

Глава 2

Модели механизмов образования событий и математические методы восстановления их параметров

Ранее были рассмотрены существующие подходы для прогнозирования событий. Некоторые подходы пытаются определить статистическую связь между наблюдаемыми признаками и фактом возникновения событий или интервалом времени между событиями, некоторые определяют закон распределения для времени между событиями, предполагая, что события возникают случайно. В этой главе предлагается новый подход. Отталкиваясь от общего определения события: «что сбылось или произошло» и добавляя: «...как результат проявления процессов», анализируются причины появления событий и предлагается, по сути, новый подход в анализе данных, основанный на построении моделей механизмов образования событий и восстановлении их параметров.

Материалы этой главы были опубликованы в следующих работах автора [195–211; 215–219].

2.1 Общая схема предлагаемого подхода прогнозирования событий

В видении автора работы события появляются вследствие функционирования некоторых механизмов, где:

Механизм – детерминированная последовательность операторов над внутренними переменными и параметрами.

Внутренние переменные X_i изменяются операторами внутри механизма. Параметры P_i изменяются вовне механизма (от внешних условий). Параметры P_i могут быть статичные или динамичные. Часть параметров P_i могут быть известными, другая часть – неизвестными (искомыми параметрами). Внутренние переменные X_i инициализируются в самом начале работы (моделирования) механизма либо нулями, либо значениями параметров.

Примечание – Термин «механизм» использует, во-первых, потому, что термин «процесс» уже был занят в теории случайных процессов. Не хотелось бы, чтобы читатель в первую очередь думал о случайных процессах, таких как пуассоновский процесс, процесс броуновского движения и др. Во-вторых, развивается именно механистическое представление о способе формирования события.

Механизм может состоять из арифметических операторов, операторов ветвления, элементарных функций и оператора формирования событий. Оператор формирования событий, как правило, располагается внутри оператора ветвления (выполняется при соблюдении некоторых условий). Оператор формирования событий является ключевым элементом механизма образования событий. Событие может нести некоторую информацию, характеристики. Механизм функционирует во времени, т.е. реализован некоторый цикл, в каждой итерации которого продвигается время на один шаг вперёд, например на день. Если шаг времени бесконечно малый, то механизм работает в непрерывном времени, но с ограничением на возможности вычислений. Сформировав одно событие, механизм не останавливается, а продолжает работу.

Однако как досконально устроены механизмы появления реальных экономических событий, неизвестно. Приходится использовать модели. Модель как упрощённая версия объективной реальности будет концентрироваться на основных аспектах появления событий. Исследователь должен соблюсти баланс между детальностью и простотой модели. Чем детальнее и сложнее она окажется, тем затруднительнее будет идентификация всех её параметров.

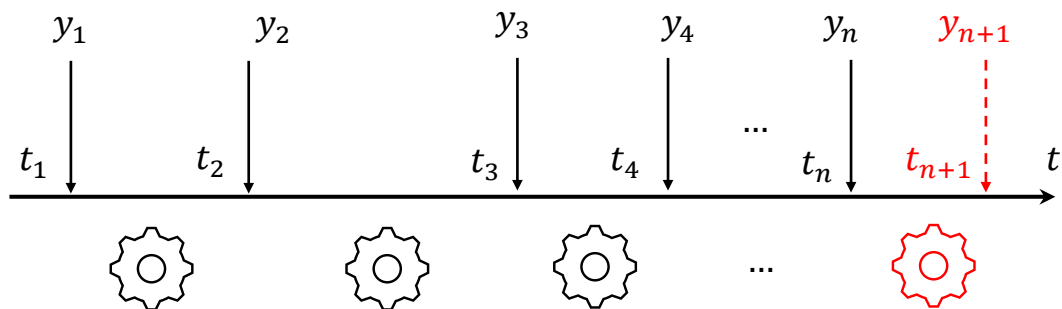
Из-за неполного знания появляется некоторая неопределённость. Параметры механизма могут иметь вероятностную природу. Внутри механизма осуществляются арифметические операции с такими параметрами. В результате формируемые события могут иметь некоторый разброс как во времени появления, так и в их характеристиках. Однако данная неопределённость будет значительно меньше, чем если изначально воспринимать события как полностью случайные, как это делается в теории случайных процессов. Большая часть наблюдаемых интервалов времени

между событиями будет объясняться именно механизмом образования событий, а не случайными факторами. Из-за этой неопределённости событие может появиться чуть раньше или чуть позже прогнозируемого момента времени.

Примечание – В этом случае под прогнозируемым моментом времени понимается момент появления события при функционировании механизма со значениями параметров, равными их математическому ожиданию. Такой механизм образования событий также можно назвать случайным процессом. Но в первую очередь интерес составляет та его детерминированная часть, которая задаёт операции над его элементами, имеющими некоторый разброс.

Формальное определение механизма образования событий на языке аксиом Колмогорова вынесено в приложение А.

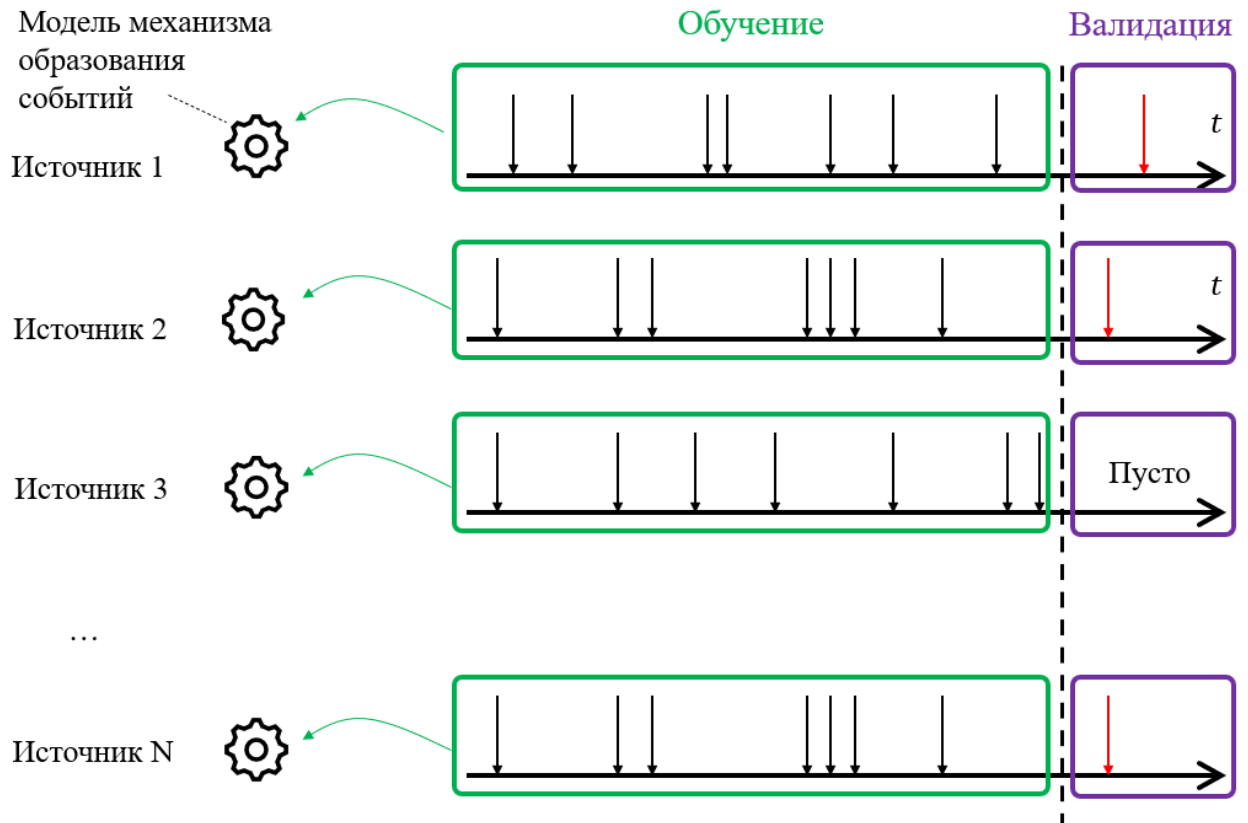
Основная задача наглядно представлена на рисунке 2.1 (шестерёнками изображена работа одного и того же механизма образования событий, сравните с рисунками 1.3, 1.13 и 1.18). Если по последовательности событий получится восстановить механизм их образования, то не составит труда предсказывать будущее событие $(t_{n+1}; y_{n+1})$, и даже не одно, а несколько.



Источник: составлено автором.

Рисунок 2.1 – Основная задача – восстановить механизм образования событий

Во многих популярных сегодня задачах анализа данных решаются практические задачи, касающиеся множества индивидуумов (клиентов). Если в подобных задачах требуется предсказать будущее событие для каждого клиента, то схема прогнозирования событий будет выглядеть, как на рисунке 2.2, сравните с рисунками 1.10 и 1.11.



Источник: составлено автором, опубликовано в совместной публикации [219].

Рисунок 2.2 – Схема прогнозирования событий для многих источников

На данной схеме шестерёнками обозначены уже разные механизмы, порождающие множество событий по каждому источнику. У каждого источника может быть собственный механизм образования событий. Под источником понимается необязательно один индивидуальный клиент, а, возможно, несколько клиентов сразу, например семья, члены которой обращаются за товарами или услугами, или организация, состоящая из людей, делающих общее дело. В этом случае события от нескольких клиентов надо объединить в один поток событий. В то же время, если исходные данные были представлены одним потоком для абсолютно разных клиентов, то поток событий необходимо разделить на множество разных потоков, выделяя события по источнику, где они образованы.

На данной схеме приведено обозначение, какие события должны использоваться для построения моделей механизмов образования событий и определения их параметров (обучения), а какие события или их отсутствие – для валидации полученных моделей. Данная схема имеет существенные

различия с тем, как поступают чаще всего в анализе данных, рисунок 1.11. На самом деле при обучении и валидации схема значительно усложняется, о чём будет рассказано ближе к концу этой главы.

Следует выделить пять этапов для реализации работы описанного подхода, основанного на восстановлении механизмов образования событий:

- 1) подготовить данные, то есть разделить события по разным выборкам в зависимости от того, каким механизмом они образованы;
- 2) предположить модель механизма $Program(X, P)$, в результате которой образуются события в источнике (внутренние переменные $X = \{X_i\}$ инициализируются внутри модели);
- 3) по имеющейся выборке событий $(t_i; y_i)$ восстановить неизвестные параметры $P = \{P_i\}$ (статичные или динамичные);

Примечание – Для простых моделей $Program(X, P)$ восстановление неизвестных параметров $\{P_i\}$ возможно осуществить аналитическими методами. Для сложных моделей надо использовать численную оптимизацию.

- 4) произвести экстраполяцию параметров P механизма на будущее;
- 5) получить прогноз будущих событий $(t_i; y_i)$, моделируя механизм их образования $Program(X, P)$ с установленными значениями параметров.

Возможно добавление шестого этапа, связанного с проверкой адекватности и корректировкой модели.

Первый этап зачастую выполнен автоматически, если имеется только один набор событий и нет никакой информации, как его можно было бы разделить, например, когда данные выглядят, как на рисунке 1.1. Но если каждое событие снабжено, например, идентификатором клиента, то поток событий следует разделить по этим идентификаторам. Если же идентификаторы разные, но они по каким-то причинам очевидно связаны с одним и тем же механизмом образования событий, тогда надо объединить такие данные в один поток.

Остальные этапы рассмотрим более подробно.

2.2 Выбор и описание модели механизма образования событий

Описание модели

Чтобы построить модель механизма образования событий, вначале следует разобраться, что такое «образование» (появление) нового события. Как было сказано в начале первой главы, в Колмогоровском определении события заложено выполнение определённой комбинации условий (подмножество из множества элементарных исходов). То есть событие появляется в результате проверки логических выражений. Только в этой работе оно появляется не в результате сравнения результатов случайного опыта, принимающего одно из возможных значений из множества элементарных исходов, а в результате сравнения друг с другом значений, вычисляющихся внутри механизма из внутренних переменных или параметров.

Так как механизм функционирует во времени по шагам (имеется цикл из последовательности операторов), то за один шаг (за одно выполнение тела цикла) должна осуществляться хотя бы одна проверка условий наступления события. Даже когда шаги очень маленькие, например соответствуют микросекундам, всё равно на каждом таком малом шаге проверяются условия наступления события.

Как записывать такие механизмы образования событий? На самом деле, любая «форма» записи информации – лишь способ донесения этой информации. Можно донести информацию как обычными словами, так и в виде алгоритма и математических формул. Это не должно иметь принципиального значения, лишь бы исследователь понимал, как формируется событие. Можно комбинировать разные формы записи для большей доступности и понимания.

В качестве **сквозного примера** возьмём модель, как в системах управления запасами [106], где строятся пилообразные графики динамики запасов некоторого предприятия, но сконцентрируемся исключительно на

образовании событий пополнения запаса. Модель механизма образования событий можно описать словами. Например, можно сказать, что ежедневно из запасов вычитается величина спроса; проверка оставшегося запаса осуществляется в конце дня; если запасы опустились ниже критической отметки, то формируем заказ на пополнение запасов до фиксированного максимума, который для простоты пусть прибывает мгновенно и пополняет запасы. Или же можно записать алгоритм:

- 1) Пока ($t \leq t_{end}$)
- 2) $X = X - P_1(t)$
- 3) Если ($X \leq P_2$), то
- 4) Создать событие ($t, P_3 - X$)
- 5) $X = P_3$
- 6) Продвинуть время t и обновить параметр $P_1(t)$
- 7) Конец цикла,

где X – запасы, $P_1(t)$ – это нестационарный и, возможно, случайный спрос, P_2 – критический уровень запасов, P_3 – максимальный уровень запасов. Здесь X является внутренней переменной, а $P_1(t)$, P_2 и P_3 – параметрами. Причём обратим внимание, что в этом примере параметр $P_1(t)$ – функция времени, а P_2 и P_3 являются статичными. Разница между внутренними переменными и параметрами в том, что переменные изменяют свои значения в результате работы механизма, а параметры, если они динамические, изменяют свои значения от некоторых внешних факторов. Внутренние переменные должны быть проинициализированы в начале работы механизма. В этом примере переменная X инициализируется значением параметра P_3 (максимальным запасом). Также обратим внимание, что формируемое событие, помимо момента времени, возвращает значение u_i , равное в этом примере величине заказа $P_3 - X$.

Примечание – В системах управления запасами не мгновенное поступление заказа (длительность функционального цикла) выражается лишь повышением уровня критической отметки запаса. А разброс во времени поступления компенсируется использованием страховых запасов.

В математической же форме можно сказать, что момент возникновения следующего события определяется в соответствии с функцией верхнего предела

$$t_{i+1} = t, \quad t: \int_{t_i}^t P_1(\tau) d\tau = y_i, \quad (2.1)$$

где $P_1(t)$ – тот же нестационарный спрос, а y_i – величина заказа, совершённого ранее ($y_i = P_3 - X$). Стоит сделать оговорку, что на самом деле даже в случае отсутствия разброса у спроса $P_1(t)$ наблюдаемый объём заказа y_i не совпадает с расходуемым, они лишь приближённо соответствуют друг другу $y_i \approx \int_{t_i}^{t_{i+1}} P_1(t) dt$, об этом подробно написано в работе [190]. Также сам знак интеграла является обобщением операции суммирования.

Примечание – Динамические параметры, такие как $P_1(t)$, являются непрерывным приближением некоторых реальных изменений экономических показателей. Чтобы избавиться от необходимости использования приближения непрерывными функциями, можно строить иерархические модели, когда накопление изменений вследствие событий, образующихся в одной модели, используется в проверке условия формирования событий более высокой по иерархии модели.

Заметим, обычно потребление ресурсов происходит дискретно, а не непрерывно, поэтому параметр $P_1(t)$ логично называть не скоростью потребления ресурса, а интенсивностью.

Пару слов надо сказать о самих моделях управления запасами. Во-первых, в задачах управления запасами моделируется расход собственных запасов или запасов подконтрольных пунктов распространения. В предлагаемом подходе появляется возможность прогнозировать события для неподконтрольных клиентов. Во-вторых, в задачах управления запасами прогноз спроса уже задан, рассчитываются издержки и оптимизируются точка и объём заказа. В предлагаемом подходе, наоборот, предстоит определить нестационарный параметр $P_1(t)$ (параметр P_2 и P_3 потребуются определять, если, помимо момента времени, требуется определить величину заказа). То есть решается обратная задача. В-третьих, тут не происходит управление ничьими запасами, не считаются издержки и не оптимизируется точка заказа, а прогнозируются будущие события.

Аналогично можно сказать про дискретно-событийные модели в имитационном моделировании. В подавляющем большинстве дискретно-событийных моделей события формируются с помощью генератора случайных чисел. И цель таких моделей заключается в том, чтобы оценить эффективность таких систем по собранной статистике. Здесь же генераторы случайных чисел не используются, никакой эффективности не считается. Задача заключается в восстановлении параметров модели по выборке произошедших в реальности событий и дальнейшем прогнозировании будущих событий.

Выбор модели

Когда есть выборка событий, для которой требуется научиться предсказывать будущие события, у исследователя уже могут быть какие-то соображения, как эти события могли быть образованы. Если хоть какие-то соображения имеются, то надо попробовать составить модель механизма образования событий. Можно не строить очень детальную модель, а ограничиться лишь ключевыми элементами, которые, на наш взгляд, способны объяснить появление событий. Если модель не подходит или подходит плохо, можно постепенно вносить в неё изменения, чтобы её объясняющая способность улучшалась. Даже если не получится полностью угадать структуру механизма образования событий, а только частично, всё равно это, скорее всего, будет лучше, чем изначально списать появление событий на абсолютную случайность.

Если же совсем не имеется никаких соображений о том, как образуются события, то есть несколько рекомендаций:

- можно ещё раз взглянуть на имеющуюся выборку, возможно, её следует ещё раз разделить/объединить по разным источникам, тогда, быть может, проявится некоторая закономерность;
- объединять несколько событий в одно событие, руководствуясь некоторыми принципами;

- перейти на более высокий уровень абстракции, отказаться от подробной детализации, работать с более абстрактными агрегированными показателями;
- попробовать автоматически подбирать схему модели, пока она не будет соответствовать наблюдаемым событиям (об этом будет рассказано в главе 3);
- и в случае, если ничего не помогает, отказаться от предлагаемого подхода, применять методы, описанные в первой главе, или признать слишком большую неопределённость и рассматривать события как абсолютно случайные.

Заметим, что любое моделирование относится к искусству, где разработчик модели должен проявить свои опыт и интуицию. Формализация реальных происходящих механизмов образования событий в виде логически выстроенных операторов сопряжена со многими проблемами, преодоление которых ложится на плечи разработчика модели. Подробные обсуждения проблем формализации выходят за рамки данной работы.

Далее рассмотрим примеры моделей.

Модель с накоплением или опустошением

Ранее рассмотренный пример, где события образуются в результате опустошения запаса, как в системах управления запасами, на самом деле представляет нечто большее, чем просто случай из задачи логистики.

Примечание – В случае, если событие образуется в результате сравнением некоего динамического параметра (например, когда уровень временного ряда пересекает какую-то границу), модель механизма образования событий будет иметь очень простой вид

- 1) Пока ($t \leq t_{end}$)
- 2) Если ($P_1(t) \geq P_2$), то
- 3) Создать событие ($t, P_1(t)$)
- 4) Продвинуть время t и обновить параметр $P_1(t)$
- 5) Конец цикла,

Данный механизм будет являться лишь частным случаем модели из сквозного примера. Достаточно представить изменение параметра $P_1(t)$ в виде накопления значений X её первой производной $P_1'(t)$. Прогноз событий в такой модели сводится к экстраполяции временных рядов любыми доступными методами. Однако экстраполяция временных рядов может быть трудной задачей, если изменение их значений происходило под действием скрытого сложного механизма, без построения модели которого экстраполяция может не получиться.

В предыдущих работах автора модель с накоплением или опустошением называлась «ёмкостной», а сам метод «ёмкостным методом». Продиктовано это было тем, что именно наполнение или опустошение этой материальной или абстрактной ёмкости приводило к событиям.

Само значение сравниваемой внутренней переменной может обозначать нематериальные абстрактные показатели. Например, в работах [200; 201] представлено использование схожей модели образования событий в сфере услуг. Прогнозируются будущие обращения клиента в салон красоты, чтобы сделать стрижку или маникюр (это разные примеры). В модели событие возникает вследствие роста волос и ногтей, или можно дать другую интерпретацию – накопление желания приобрести услугу. Тем самым внутренняя переменная увеличивает своё значение, пока не достигнет определённой отметки, после чего происходит обращение за услугой. В другой работе [206] применяется модель переполнения чаши терпения для социальных событий, таких как цветные революции. В зависимости от того, какие значения используются в качестве характеристики события y_i (факт – значения y_i всегда 1; количество протестующих; или результат протестов $y_i \in \{1,2,3\}$), накапливаемые значения можно интерпретировать как относительную скорость нарастания недовольства, скорость увеличения/подготовки количества недовольных или интенсивность усилий для достижения результата. В работе [206] также используется модель переполнения чаши терпения, но для исторических событий русско-турецких войн. В зависимости от того, что выбирается в качестве характеристики события (факт, длительность или результат войны), накапливаемыми значениями являются скорость нарастания непонимания, усталость от войны, интенсивность усилий для победы одной из сторон. Однако модели для социальных и исторических событий являлись слишком абстрактными, поэтому ответственность за прогноз будущего события не бралась, работы ограничивались восстановлением указанных нестационарных параметров, которые могут иметь определённый научный интерес.

Такая модель хорошо подходит именно для задач, связанных с материальными товарами [196], так как естественно предположить, что приобретение товаров вызывается исчерпанием запаса. В работе [215, с. 203–248; 265–271] приводятся как удачные, так и неудачные примеры

прогнозирования событий, связанных с поставками лекарственных биологически активных добавок в аптечные пункты разных регионов России. Восстанавливаемым параметром механизма образования событий является скорость расхода запаса (потребления/спроса).

В работе [215, с. 248–261] приводятся примеры использования данной модели в сфере финансов, когда событиями являются займы (выпуск облигаций), а параметром – скорость расхода заёмных средств.

Даже такая простая модель, когда событие, образуется вследствие того, что что-то исчерпалось или, наоборот, наполнилось, имеет очень широкое применение. Многие модели, таким образом, являются частным случаем или развитием этой простой модели. Добавляя дополнительные операторы и логические условия, можно сделать модель более детальной. Рассмотрим различные варианты.

Модель таймера

Если события происходят по расписанию или через заранее известные или постоянные интервалы времени (регулярный поток событий), то такой способ появления событий также можно описать с помощью модели накопления или опустошения, описанной ранее. В этом случае внутренняя переменная будет обозначать запас времени, а уменьшаться она будет на одну и ту же постоянную величину, пропорциональную шагу модели. И в тот момент, когда этот запас времени заканчивается, и будет появляться событие, которое пополняет или увеличивает соответствующий запас времени. Принципиальной разницы от описанной выше модели нет. Можно сказать, что данный способ образования событий является частным случаем механизма с накоплением/опустошением.

Примечание – При программировании корректно откладывать от предыдущего события заданный интервал времени, а затем с определённой периодичностью проверять, подошло ли текущее время к запланированному времени следующего события. В имитационном дискретно-событийном моделировании скачком продвигают время к следующему событию.

Модель с изменяющимся максимальным уровнем

В описанном выше примере из задачи управления запасами предполагалось, что запасы пополняются до постоянного максимума. В работе [215, с. 25–37] рассмотрен пример, когда клиент изменяет свой максимальный уровень запаса в зависимости от того, как быстро у него закончился предыдущий запас. Для этого вводятся два дополнительных параметра, раннее и позднее время, обозначающие критические уровни для времени исчерпания запаса. Если предыдущий запас закончился быстрее раннего времени, то максимальный запас увеличивается вдвое. Если запас закончился позже, чем позднее время, то максимальный запас сокращается вдвое¹⁾. И если время исчерпания предыдущего запаса попало в интервал между ранним и поздним временем, то максимальный запас не меняется.

Заметим: в этой работе не строятся модели управления запасами, чтобы оптимизировать собственные издержки, а решается обратная задача, когда по наблюдаемым событиям (t_i, y_i) определяются параметры механизма. Для такой модели механизма образования событий в работе [215, с. 29–34] по заранее смоделированным событиям удалось с неплохой точностью восстановить все параметры (как нестационарный спрос, максимальный запас, так и новые параметры раннего и позднего времени исчерпания запаса). После этого удалось получить прогноз будущих событий.

Модель с периодической проверкой запаса

В другом практическом примере на реальных данных [196; 215, с. 37–40], касающихся поставки бутылок кваса в универсам, из данных событий следовало, что интервалы между поставками составляли практически всегда 7, 14, 21, 28, 35 дней, т.е. были кратны неделе. Тогда модель механизма образования событий была скорректирована соответствующим образом, что запасы проверяются только раз в неделю. Параметры модели были

¹⁾ В данной модели число 2, которое является множителем или делителем максимального запаса, было предопределено в самой модели и не являлось неизвестным параметром. Однако можно и это число сделать параметром, подлежащем определению.

определены, а дата прогноза последнего события (полученная по выборке без этого последнего события) совпала с фактической, так как проверка запасов в модели была кратна 7 дням.

Модель с дополнительными случайными событиями

В этой модели появляются дополнительные события, которые происходят ранее положенного срока. Причина появления этих событий остаётся неизвестной, т.е. имеется неопределённость. Эти дополнительные события появляются из-за внешних условий, которые нельзя учесть в рамках модели. При этом количество (частота) этих дополнительных событий намного меньше, чем основных событий. Дополнительные события как бы разбавляют поток событий, порождаемый механизмом образования событий. Такими событиями в том же примере, схожем с управлениями запасами, могут быть события, связанные с внезапными акциями и скидками, вынуждающие пополнять запасы преждевременно, или же связанные с тем, что отдел планирования закупок уходит в отпуск.

Такие дополнительные события предсказывать не получится, так как они оказываются случайными. Можно будет определить их вероятностные характеристики. Но эти дополнительные случайные события будут оказывать влияние на восстановление параметров модели механизма образования основных событий. Во-первых, их надо будет как-то отделить от основного потока событий. Для этого предлагается использовать приём, который используется в методе деревьев решения при определении уровня разбиения. А именно предлагается в отсортированном по возрастанию списке значений u_i (например, объёмы покупок) перебирать разные уровни деления, как середина между двумя соседними значениями. И события, обладающие меньшими значениями, чем выбранный уровень деления, относить к дополнительным событиям, которые обрабатывать отдельно. Подробнее на практическом примере этот вид модели будет рассмотрен в главе 5.

Сложные модели

Логика появления событий в той или иной сфере может подсказывать достаточно сложные модели механизмов образования событий. Однако не слишком сложные, иначе может не получиться построить модель или не получится определить её параметры. Например, для событий, которых никто не ожидал, но они всё же произошли, которых по Николасу Талебу называют «чёрные лебеди» [1], возможно, придётся строить чрезвычайно детальную и сложную модель. Настолько сложную, что её можно будет назвать моделью «жизнь». Однако вряд ли кому-то в ближайшее время это удастся. С другой стороны, как было сказано выше, можно строить модель, которая основывается на агрегированных характеристиках с высоким уровнем абстракции. Но события в таких моделях будут обладать большой неопределённостью. Тем не менее, если модель механизма сможет объяснить хотя бы часть наблюдаемого разброса в появлении событий, то это уже будет хорошо.

Как было сказано выше, выбор модели механизма образования событий продиктован информацией, опытом и интуицией, которыми обладает исследователь. Можно пробовать разные модели – не подошла одна модель, возможно, подойдёт другая. Можно выдвигать различные гипотезы об устройстве механизма образования событий и строить модели, чтобы их подтвердить или опровергнуть. В третьей главе будет обсуждаться подход, как можно автоматически подбирать такие модели.

Ещё одной отличительной особенностью данной работы является предсказание не одного или нескольких будущих событий для одного источника событий, а предсказание событий для множества разных источников. Это является дальнейшим развитием разрабатываемых методов, чтобы их можно было применять в анализе данных. В этом случае можно предложить механизм для выбора наиболее подходящей модели для каждого источника. Подробно это будет рассмотрено в одном из примеров, представленных в главе 5.

2.3 Восстановление параметров механизма образования событий регрессионными методами

В этом параграфе рассматриваются математические методы восстановления параметров моделей механизма образования событий, представленных выше.

Для рассмотренного выше примера механизма образования событий, связанного с проверкой внутренней переменной, изменяющейся в результате накопления или опустошения, требуется по имеющейся выборке событий (t_i, y_i) определить параметры $P_1(t)$, P_2 и P_3 , где $P_1(t)$ – нестационарная скорость изменения внутренней переменной, P_2 – критический уровень, P_3 – максимальный уровень. Начнём с определения нестационарного параметра $P_1(t)$.

Восстановление нестационарного спроса $P_1(t)$

Выше, в математической форме записи механизма образования событий, формула (2.1), момент возникновения очередного события t_{i+1} определяется в соответствии с функцией верхнего предела $t_{i+1} = t, t: \int_{t_i}^t P_1(\tau) d\tau = y_i$. То есть моментом появления события t_{i+1} является момент, когда исчерпан запас y_i некоторого ресурса. В работах [190; 192; 193; 197; 203; 215] подробно показано, что даже в моделях, где $P_1(t)$ – детерминированная функция (без разброса), значения объёма пополнения запаса y_i всё равно будут иметь разброс. В итоге можно представить, что наблюдаемые объёмы пополнения запаса y_i являются интегралами от неизвестной функции $P_1(t)$ с добавлением некоторой погрешности ε_i

$$y_i = \int_{t_i}^{t_{i+1}} P_1(t) dt + \varepsilon_i. \quad (2.2)$$

В итоге имеем зашумлённые данные.

Примечание – В отличие от привычных эконометрических моделей, где погрешность/ошибка ε_i является ошибкой модели в определении фактических (истинных) наблюдений y_i , в нашем случае ε_i является ошибкой именно наблюдений относительно истинных значений расходуемого запаса. Но это различие не влияет на восстановление параметра $P_1(t)$ регрессионными методами

Можно выделить два способа восстановления неизвестного параметра $P_1(t)$ регрессионными методами.

Первый способ – восстановление по интегралам

Для восстановления нестационарного параметра $P_1(t)$ можно записать задачу минимизации суммы квадратов отклонений

$$S = \sum_{i=1}^{n-1} \left(y_i - \int_{t_i}^{t_{i+1}} P_1(t) dt \right)^2 \rightarrow \min, \quad (2.3)$$

где в качестве модели для зависимости $P_1(t)$ можно брать какую-нибудь функцию, позволяющую её восстановить методом наименьших квадратов. Например, полином степени m

$$P_1(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_m t^m, \quad (2.4)$$

где m – степень полинома.

Чтобы воспользоваться методом наименьших квадратов, необходимо взять определённый интеграл от этой полиномиальной модели

$$\int_{t_i}^{t_{i+1}} P_1(t) dt = a_0(t_{i+1} - t_i) + a_1 \frac{t_{i+1}^2 - t_i^2}{2} + a_2 \frac{t_{i+1}^3 - t_i^3}{3} + \dots + a_m \frac{t_{i+1}^{m+1} - t_i^{m+1}}{m+1}. \quad (2.5)$$

Оптимизационную задачу можно записать в матричном виде

$$S = (Y - XA)^T (Y - XA) \rightarrow \min, \quad (2.6)$$

где $Y = (y_1, \dots, y_{n-1})^T$;

$A = (a_0, a_1, \dots, a_m)^T$;

X – матрица размерностью $(n-1) \times (m+1)$, элементы которой соответствуют множителям при соответствующих коэффициентах,

$$X_{i,1} = t_{i+1} - t_i, X_{i,2} = \frac{t_{i+1}^2 - t_i^2}{2}, \dots, X_{i,m+1} = \frac{t_{i+1}^{m+1} - t_i^{m+1}}{m+1}.$$

Решение получается в знакомом виде

$$A = (X^T X)^{-1} X^T Y. \quad (2.7)$$

Однако при решении сразу всплывает несколько проблем.

1) Нормализация. При высокой степени полинома m значения $t_{i+1}^{m+1} - t_i^{m+1}$ рассчитать зачастую невозможно напрямую по данным наблюдений t_i , так как происходит переполнение разрядной сетки при арифметических операциях. Необходима нормализация значений t_i , например, привести к шкале $[-1, +1]$ (или другая нормализация, уменьшающая абсолютные значения переменной)

$$t_i^* = \frac{(t_i - (\max t + \min t)/2)}{(\max t - \min t)/2}. \quad (2.8)$$

Проблема с арифметическими вычислениями, таким образом, решается, но появляется другая. Такая нормализация сужает сами интервалы между событиями $t_{i+1} - t_i$, они теперь измеряются в малых значениях. Получается, что все ещё большие значения интеграла y_i определяются интегрированием функции на очень узком интервале. В результате значения полинома $P_1(t)$ становятся очень большими. Чтобы исправить ситуацию, можно уменьшать значения y_i , но не нормализовать их. Уменьшать значения y_i надо пропорционально тому, во сколько раз изменился размах наблюдений t_i^* (равный 2) по сравнению с ненормализованными t_i

$$y_i^* = y_i \times 2/(\max t - \min t), \quad (2.9)$$

или же можно уменьшать значения полученного полинома

$$P_1^*(t) = P_1(t) \times 2/(\max t - \min t). \quad (2.10)$$

2) Мультиколлинеарность. С ростом степени полинома m столбцы $(t_{i+1}^k - t_i^k)/3$ матрицы X при высоких степенях k становятся всё больше и больше линейнозависимы. Полученные значения параметров a_0, a_1, \dots, a_m становятся всё менее и менее стабильными, погрешность их оценки ($\text{Cov}(\hat{A})$) становится всё больше и больше. Это приводит к большим погрешностям оценки определённых интегралов и самого полинома. В примере ниже уже при степени полинома m больше 10 матрица $(X^T X)$ приближается к сингулярной.

Полученные параметры оценки полинома принимают очень большие значения (в примере ниже около миллиона).

3) Когда количество параметров $m + 1$ сравнивается с количеством наблюдений $n - 1$, то имеется только одно точное решение. Погрешность оценки интегралов близка к нулю. Но оценка дисперсии становится бесконечной, так как делить приходится на ноль ($\sum_{i=1}^n \varepsilon_i^2 / (n - 1 - m - 1)$).

Если же параметров становится ещё больше, то нужна регуляризация, например, добавить $\alpha \sum_{j=0}^m a_j^2 = \alpha A^T A$ в целевую функцию, где α – достаточно малый коэффициент регуляризации. Выражение (2.7) для оценки параметров изменится следующим образом

$$A = (X^T X + \alpha I)^{-1} X^T Y. \quad (2.11)$$

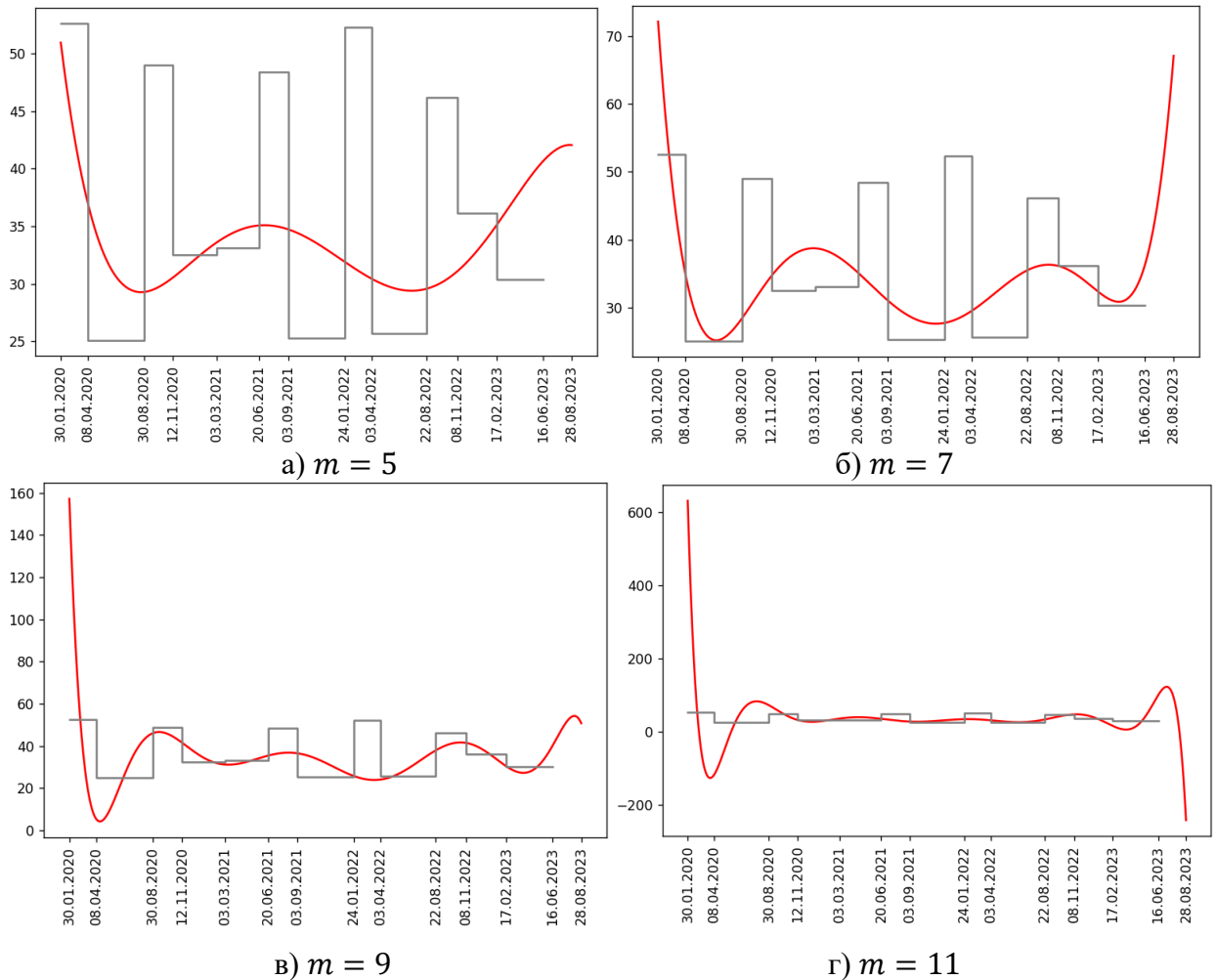
Такая регуляризация (штраф) препятствует параметрам принимать очень большие значения, но приводит к смещённости оценок. Сам полином $P_1(t)$ большой степени m даже при такой регуляризации всё равно сильно отличается от того, что должно быть.

4) Полином плохо объясняет наблюдаемые данные. Даже при высокой степени полинома m результат остаётся удручающим. Для наглядности рассмотрим пример из раздела 1.4, по данным событий, таблица 2.1, которые получаются в результате моделирования системы управления запасами, пилообразный рисунок 1.17, пытаемся определить нестационарный спрос (функцию на рисунке 1.16, но ещё не зная, что она гармоническая). Результаты восстановления функции $P_1(t)$ полиномом разных степеней изображены на рисунке 2.3.

Таблица 2.1 – Данные результата моделирования системы управления запасами для дальнейшего восстановления исходной функции нестационарного спроса

Дата t_i	Объём y_i	Дата t_i	Объём y_i	Дата t_i	Объём y_i
30.01.2020	3628,994	20.06.2021	3631,264	08.11.2022	3649,903
08.04.2020	3615,591	03.09.2021	3614,047	17.02.2023	3612,137
30.08.2020	3625,53	24.01.2022	3608,333	16.06.2023	3629,912
12.11.2020	3606,921	03.04.2022	3618,266	28.08.2023	3620,067
03.03.2021	3609,151	22.08.2022	3601,899	-	-

Источник: составлено автором.



Ступеньки (горизонтальные отрезки) показывают среднее значение функции на этом интервале $y_i/(t_{i+1} - t_i)$, площадь под каждой из ступенек (до оси ноль) – наблюдаемый интеграл y_i .

Источник: составлено автором.

Рисунок 2.3 – Восстановление параметра $P_1(t)$ регрессионными методами с помощью полинома разных степеней: а) $m = 5$; б) $m = 7$; в) $m = 9$; г) $m = 11$

Как можно заметить, полином даже высокой степени плохо объясняет данные. На рисунках 2.3(б) и (в) в середине отчётливо заметен провал, не соответствующий центральному наблюдению.

Второй способ – восстановление интегральной функции

В отличие от предыдущего способа восстанавливается интегральная (первообразная функция) $\Pi(t)$

$$\Pi(t) = \int_{\tau=0}^t P_1(\tau) d\tau. \quad (2.12)$$

Входными данными являются данные $Y = (Y_1, Y_2, \dots, Y_n)^T$, где (заметим, что Y уже состоит из n элементов)

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ \dots \\ Y_n \end{bmatrix} = \begin{bmatrix} 0 \\ y_1 \\ y_1 + y_2 \\ y_1 + y_2 + y_3 \\ \dots \\ y_1 + y_2 + y_3 + \dots + y_{n-1} \end{bmatrix}. \quad (2.13)$$

Восстановление этой интегральной функции $\Pi(t)$ будет происходить методом наименьших квадратов

$$S = \sum_{i=1}^n (Y_i - \Pi(t_i))^2 \rightarrow \min. \quad (2.14)$$

В качестве модели для зависимости $\Pi_1(t)$ также можно брать полином степени m

$$\Pi_1(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_m t^m. \quad (2.15)$$

Оценки параметров получаются по тому же выражению

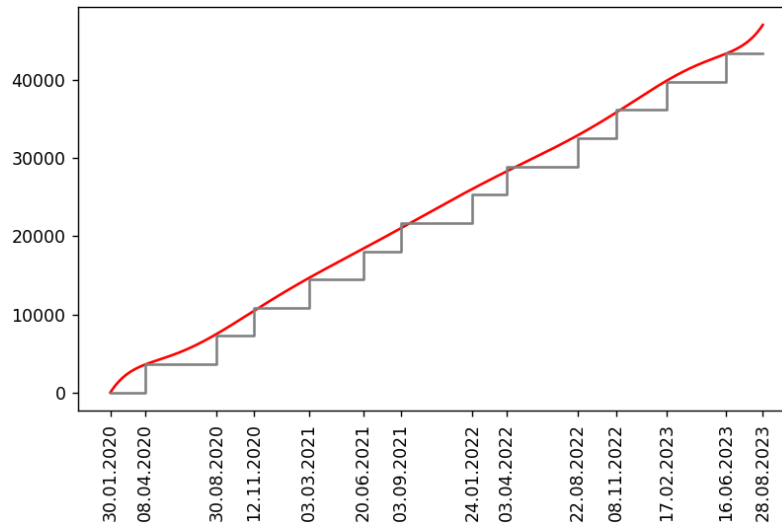
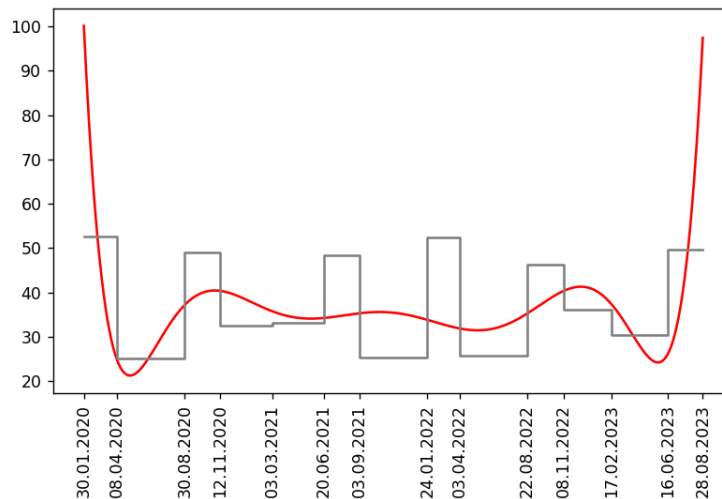
$$A = (X^T X)^{-1} X^T Y, \quad (2.16)$$

где X – матрица размерностью $n \times (m + 1)$, с элементами $X_{i,1} = 1$, $X_{i,2} = t_i$, $X_{i,3} = t_i^2$, ..., $X_{i,m+1} = t_i^m$.

Чтобы получить значения исходного параметра $P_1(t)$, необходимо взять производную от $\Pi_1(t)$, заметим: $P_1(t)$ будет полиномом степени меньшей на единицу (без параметра a_0)

$$P_1(t) = 0 + a_1 + 2a_2 t + \dots + m a_m t^{m-1}.$$

В этом способе будут все те же проблемы, что и ранее: необходимость в нормализации, мультиколлинеарность, потребность в регуляризации, полином всё равно плохо объясняет данные. На рисунке 2.4 изображена интегральная функция $\Pi(t)$ и параметр $P_1(t)$ для того же самого примера, описанного выше (степень полинома $m = 9$).

а) интегральная функция $\Pi(t)$ б) параметр $P_1(t)$

На рисунке а) ступенчатая функция обозначает накопленные значения, высота каждой ступеньки соответствует значению y_i и визуально не отличается. Интегральная функция стремится проходить через вершины ступенек. На рисунке б) ступенчатая функция показывает среднее значение функции на этом интервале $y_i/(t_{i+1} - t_i)$, площадь под каждой из ступенек (до оси ноль) – наблюдаемый интеграл y_i .

Источник: составлено автором.

Рисунок 2.4 – Восстановление: а) интегральной функции $\Pi(t)$ и б) параметра $P_1(t)$ с помощью полиномиальной модели степени $m = 9$

Можно с уверенностью констатировать, что применение регрессионных методов и полиномиальной модели плохо подходит для восстановления нестационарного параметра $P_1(t)$. Ниже будет описан другой способ восстановления нестационарного параметра $P_1(t)$.

Восстановление стационарных параметров P_2 и P_3

В разбираемом примере механизма образования событий дополнительные параметры имеют такую интерпретацию: P_2 – критический уровень (после которого происходит пополнение запаса), P_3 – максимальный уровень (до которого пополняются запасы). Необходимо по данным наблюдений $(t_i; y_i)$ определить эти параметры.

На самом деле эти два параметра связаны. В том смысле, что если одновременно увеличить один и уменьшить другой на одинаковую величину, то результат моделирования не изменится, будет получена одна и та же последовательность событий. В логистике критический запас зависит от времени исполнения заказа. Чем больше требуется времени от заказа до его поступления, тем больше этот критический уровень. А максимальный уровень соответствует сумме критического уровня и объёма пополнения y_i (из $X \leq P_2$ и $y_i = P_3 - X$, где X – текущий запас, следует, что $\bar{y}_i \approx P_3 - P_2$, откуда $P_3 \approx \bar{y}_i + P_2$).

Таким образом, в качестве параметра P_2 можно брать произвольное положительное значение, например, можно брать ноль, $P_2 = 0$, если у нас нет информации или нас не интересует время исполнения заказа. Заметим, что если исследователем, который анализирует события, является организация, осуществляющая поставку этих заказов, то такая информация может иметься. Тем не менее на дальнейший прогноз событий параметр P_2 влиять не будет, так как расходоваться будет разница между P_3 и P_2 .

Можно было заметить, что из условия $X \leq P_2$ следует, что величина y_i получается немного завышенной. Событие происходит, когда запас X опустился ниже P_2 , но на сколько ниже? Эта величина находится в интервале от 0 до $P_1(t_i)$ (то, на сколько уменьшился запас в тот день, когда возникло событие). Причём есть веские причины полагать, что она распределена равномерно на этом интервале. В среднем (по случайным реализациям, а не по

наблюдениям) каждый наблюдаемый объём пополнения запаса y_i завышен на $P_1(t_i)/2$. В итоге можно оценить параметр P_3 как

$$P_3 = P_2 + \frac{1}{n} \sum_{i=1}^n \left(y_i - \frac{P_1(t_i)}{2} \right) = 0 + \frac{1}{n} \sum_{i=1}^n \left(y_i - \frac{P_1(t_i)}{2} \right). \quad (2.17)$$

Заметим, что для оценки параметра P_3 нужно вначале получить оценку нестационарного параметра $P_1(t)$.

2.4 Восстановление параметров механизма образования событий методами машинного обучения

Методы машинного обучения, в том числе нейросети, способны выявлять зависимость объясняемой переменной Y от входных признаков X , $Y = f(X)$ (при этом сама модель $f(X)$ имеет характер чёрного ящика). Однако в методах машинного обучения нет возможности, не залезая внутрь конкретного метода, восстанавливать зависимость $f(X)$ по значениям определённых интегралов $y_i = \int_{t_i}^{t_{i+1}} f(X(t))dt$ (когда признаки зависят от времени).

Единственным выходом остаётся восстанавливать интегральную (первообразную) функцию $\Pi(t) = \int_{\tau=0}^t P_1(\tau)d\tau$ по накопленным значениям Y и признакам X , где Y столбец значений вида

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ \dots \\ Y_n \end{bmatrix} = \begin{bmatrix} 0 \\ y_1 \\ y_1 + y_2 \\ y_1 + y_2 + y_3 \\ \dots \\ y_1 + y_2 + y_3 + \dots + y_{n-1} \end{bmatrix}, \quad (2.18)$$

а в качестве признаков X можно брать либо только время в момент наблюдений, либо время в разных степенях (как для полиномиальной модели) вплоть до некоторой степени m , тем самым помогая методу машинного обучения найти закономерность

$$X = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ \dots \\ t_n \end{bmatrix} \text{ или } X = \begin{bmatrix} 1 & t_1 & t_1^2 & \dots & t_1^m \\ 1 & t_2 & t_2^2 & \dots & t_2^m \\ 1 & t_3 & t_3^2 & \dots & t_3^m \\ 1 & t_4 & t_4^2 & \dots & t_4^m \\ \dots & \dots & \dots & \dots & \dots \\ 1 & t_n & t_n^2 & \dots & t_n^m \end{bmatrix}. \quad (2.19)$$

Методов машинного обучения большое количество. Для демонстрации были разобраны практически все регрессионные методы машинного обучения очень популярной в текущее время библиотеки `scikit learn` [107] и дополнительные популярные методы, не входящие в эту библиотеку методов. Такими методами являются SVR (Support Vector Regression) [108], KernelRidge (Kernel ridge regression) [109], KNeighborsRegressor (Nearest Neighbors Regression) [110], GaussianProcessRegressor [111], PLSRegression (Partial Least Squares Regression) [112], DecisionTreeRegressor [113], GradientBoostingRegressor [114], MLPRegressor (Multi-layer Perceptron Regression) [115], LGBMRegressor (Light Gradient Boosting Regressor) [116], XGBRegressor (библиотека XGBoost, буква X не является аббревиатурой, а означает улучшение градиентного бустинга) [117].

Применяя эти методы машинного обучения, в итоге получаем зависимость интегральной функции $\Pi(t)$. Нас же интересует её производная

$$P_1(t) = \Pi'(t). \quad (2.20)$$

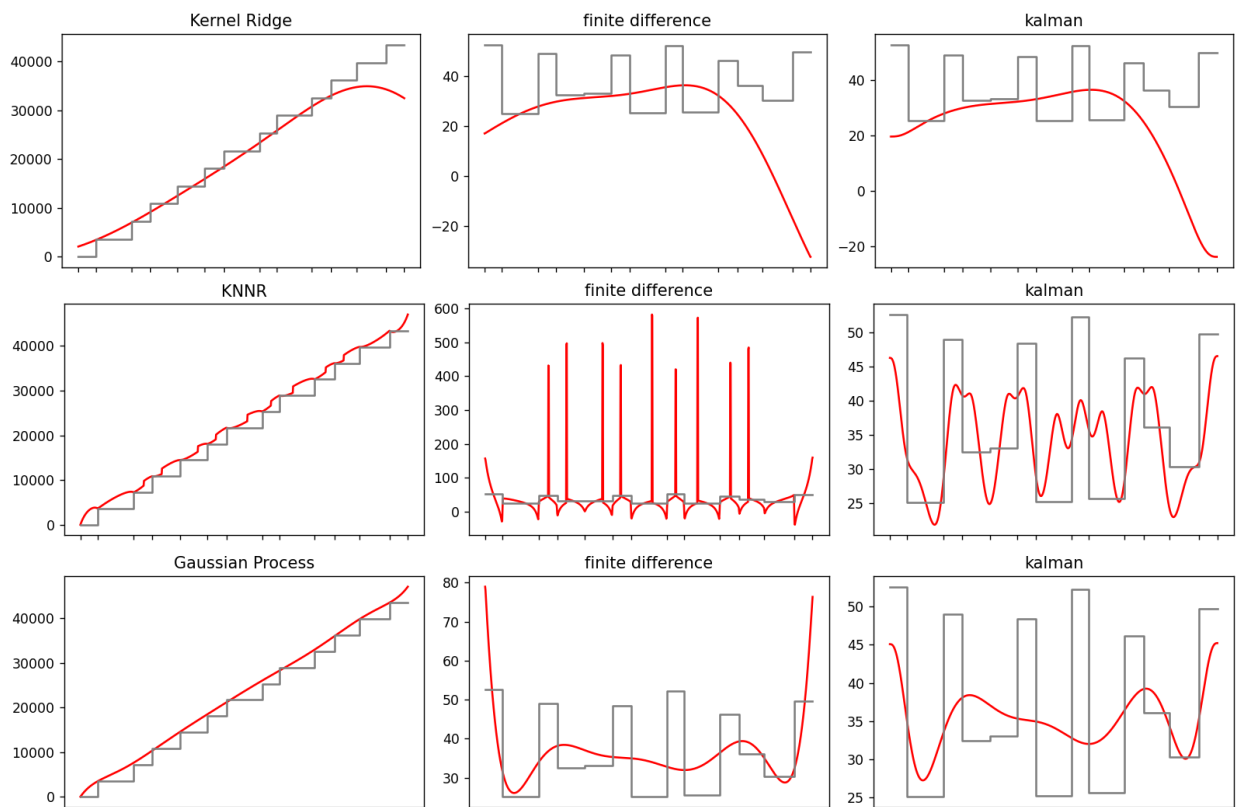
Однако обученные методами машинного обучения модели $\Pi(t)$ имеют характер чёрного ящика. То есть модель $\Pi(t)$ не может быть дифференцирована аналитически. Единственным выходом остаётся использовать численное дифференцирование. Методов численного дифференцирования тоже достаточно много. В языке Python есть большая библиотека методов численного дифференцирования `derivative` [118]. В примере ниже самые лучшие результаты получаются при дифференцировании с помощью сглаживания Калмана (программная реализация [119]), для сравнения также показан метод конечных разностей.

Случай 1, $X = (t_1, t_2, \dots, t_n)^T$ – столбец. Приведём результаты применения методов машинного и численного дифференцирования для восстановления параметра $P_1(t)$.

Метод PLSRegression не смог запуститься для заданных данных.

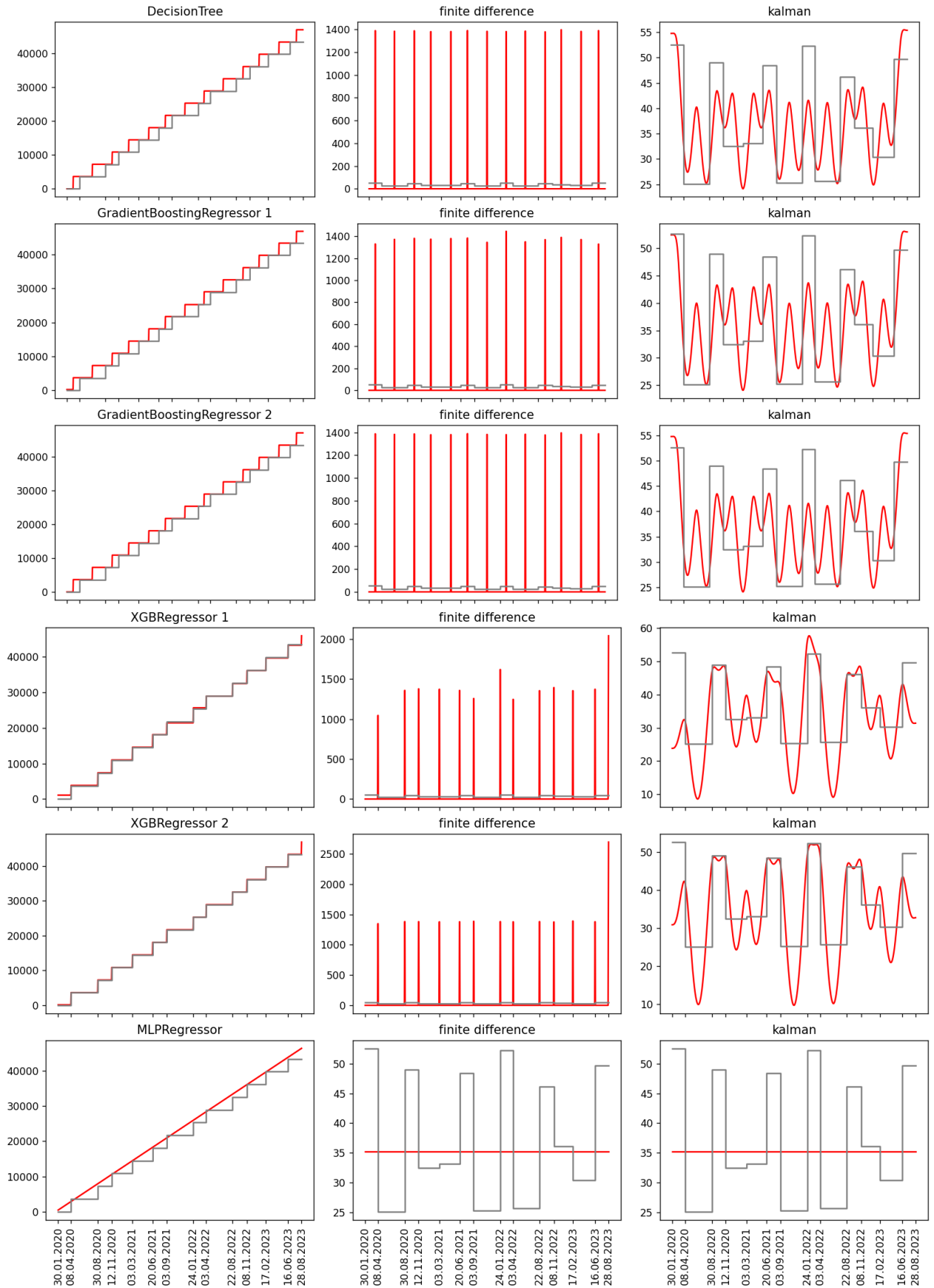
Метод SVR (с разными ядерными функциями: линейная, полиномиальная, радиальные базисные функции) и метод LGBMRegressor не смогли даже близко восстановить интегральную функцию $\Pi(t)$.

Остальные методы смогли восстановить интегральную функцию $\Pi(t)$, но производная от неё всё ещё плохо соответствует исходной функции, заложенной при моделировании, рисунки 2.5 и 2.6.



Источник: составлено автором.

Рисунок 2.5 – Применение методов KernelRidge, KNeighborsRegressor, GaussianProcessRegressor для восстановления параметра $P_1(t)$ по X в виде столбца

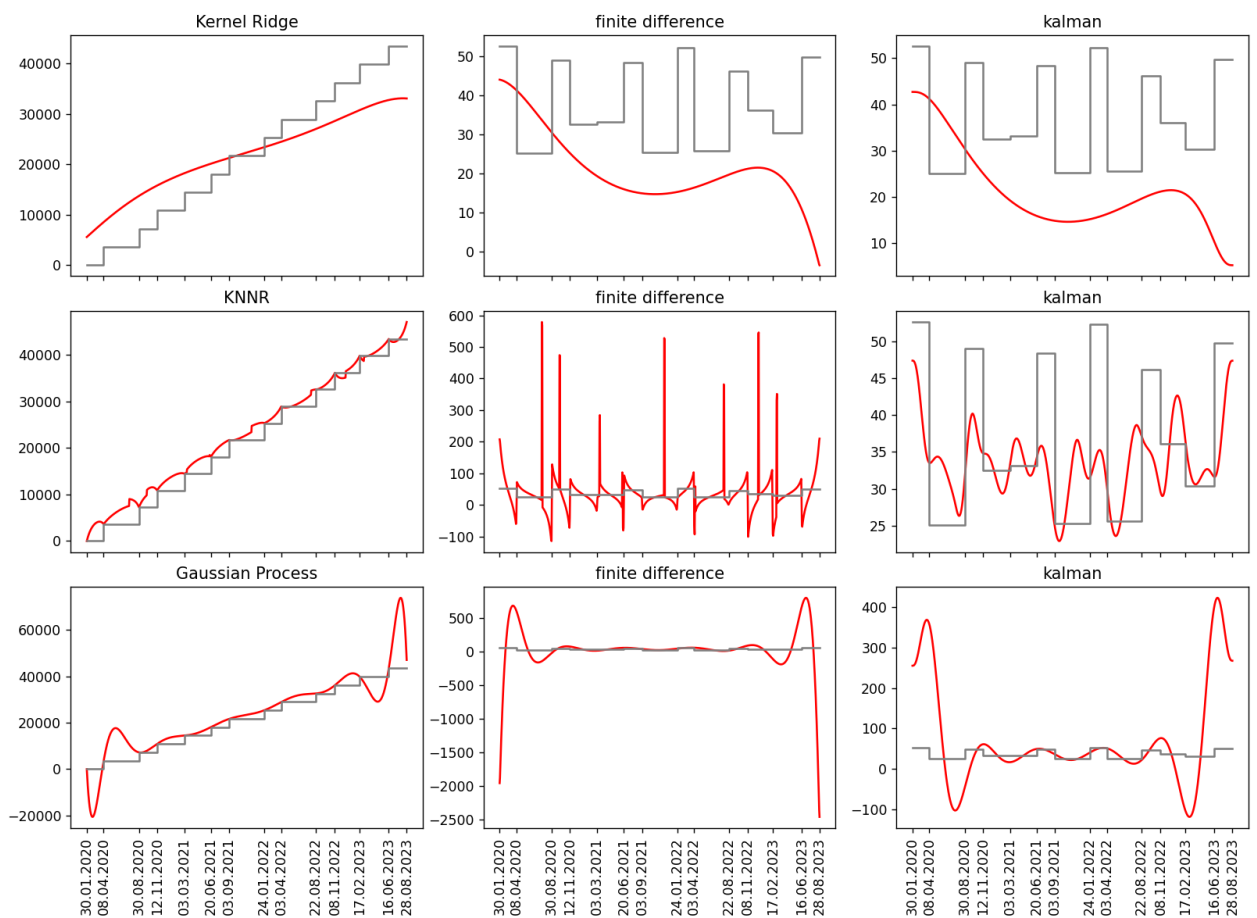


Источник: составлено автором.

Рисунок 2.6 – Применение методов DecisionTreeRegressor, GradientBoostingRegressor с глубиной дерева 1 и 2, XGBRegressor с глубиной дерева 1 и 2, MLPRegressor с количеством скрытых слоев 15 для восстановления параметра $P_1(t)$ по X в виде столбца

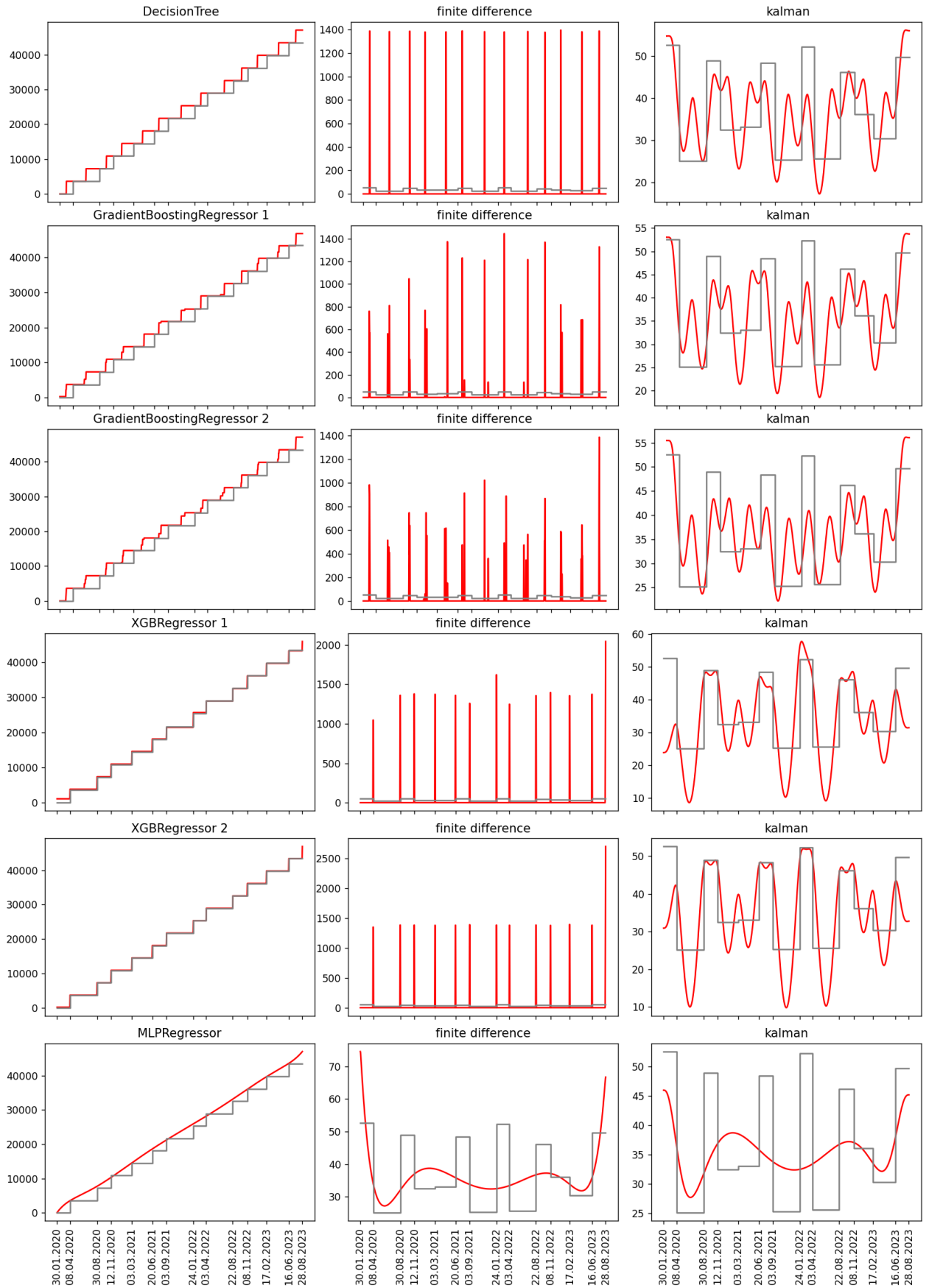
Случай 2, X – матрица $n \times (m + 1)$, $m = 3$. Попробуем помочь методам машинного обучения, подавая в качестве признаков не столбец, а матрицу в виде (2.19). Результаты представлены на рисунках 2.7 и 2.8.

Метод `PLSRegression` не смог запуститься при заданных данных. Метод `SVR` и метод `LGBMRegressor` не смогли восстановить интегральную функцию. Остальные методы могут восстановить интегральную функцию, но вот её производная всё ещё неправильно восстановилась. В методе, основанном на гауссовых процессах, результат мог бы быть самым близким к тому, что ожидалось получить, если бы не огромные всплески в самом начале и самом конце. Также в методе, основанном на `XGBoost`, результат мог бы считаться хорошим, если бы не большие отклонения вниз. Результат остальных методов удручающий. Напомним, что ожидаемым результатом является гармоническая функция, как на рисунке 1.16.



Источник: составлено автором.

Рисунок 2.7 – Применение методов `KernelRidge`, `KNeighborsRegressor`, `GaussianProcessRegressor` для восстановления параметра $P_1(t)$ по X в виде матрицы



Источник: составлено автором.

Рисунок 2.8 – Применение методов DecisionTreeRegressor, GradientBoostingRegressor с глубиной дерева 1 и 2, XGBRegressor с глубиной дерева 1 и 2, MLPRegressor с количеством скрытых слоев 15 для восстановления параметра $P_1(t)$ по X в виде матрицы

В итоге можно констатировать, что распространённые методы машинного обучения плохо справляются с задачей восстановления функции по последовательности интегралов.

2.5 Восстановление параметров механизма образования событий методами сплайновой коллокации

Напомним, что перед нами стоит задача восстановления неизвестного нестационарного параметра $P_1(t)$ по последовательности интегралов y_i , которые наблюдаются с погрешностью ($y_i = \int_{t_i}^{t_{i+1}} P_1(t)dt + \varepsilon_i$). Некоторые причины возникновения этой погрешности и каковы её вероятностные характеристики обсуждаются в работах [190; 192; 193; 197; 199; 203; 215].

Сплайн, в отличие от обычного полинома, заданного на всей области определения, есть последовательность некоторых функций, в основном тех же самых полиномов, но заданных на более узких интервалах, определённых между узлами сплайна. В самих узлах сплайна происходит смена с одной функции на другую, но происходит это с условием непрерывности как самой итоговой функции, так и её производных до некоторого порядка.

Сплайны бывают интерполяционными, т.е. точно проходящими через заданные точки наблюдений, а бывают сглаживающие (аппроксимирующие), которые используются при зашумлённых наблюдениях.

С точки зрения решаемой задачи к интерполяционным сплайнам можно отнести интегро-дифференциальные сплайны (ИД-сплайны) [120; 121; 122; 123]. Интегро-дифференциальные сплайны являются частным случаем методов сплайновой коллокации при решении интегро-дифференциальных уравнений [124; 125; 126]. Дифференциальными условиями являются условия непрерывности производной в узлах сплайна (в точках сочленения). Интегральными условиями являются равенства интегралов от сплайна наблюдаемым интегралам, $\int_{t_i}^{t_{i+1}} g(t)dt = y_i$, где $g(t)$ – искомый сплайн, $y_i = \int_{t_i}^{t_{i+1}} P_1(t)dt$ – наблюдаемые интегралы. Полученные дифференциальные

и интегральные условия составляют систему линейных уравнений, решение которой позволяет определить параметры сплайна $g(t)$. Однако интегральные условия в виде строгих равенств заставляют построенный сплайн строго соответствовать всем наблюдениям. Такие условия даже Карл де Бур [127, с. 80] называл «интерполяционными» условиями. Есть и другие работы, где по интегралам восстанавливается функция, например [128; 129], но в них тоже получается интерполяционный сплайн.

В сглаживающих (аппроксимирующих) сплайнах предполагается приближённое соответствие значения сплайна наблюдениям.

Примечание – Теория сглаживающих сплайнов имеет достаточно глубокую историю. В считающейся основополагающей монографии Альберга, Нильсона, Уолша 1967 года [130, с. 2], в которой собраны все основные достижения по сплайнам к тому времени, утверждается, что «аппроксимация сплайнами в её современном виде впервые появилась в работе Шёнберга» 1946 года [131]. При этом отмечается «очень тесная связь между теорией сплайнов и расчётов балок». В статье [132], посвящённой истории появления сплайнов, говорится о том, что «современное представление кусочно-полиномиальных функций встречается в работе [133] Бубнова “Строительная механика корабля” 1912 года» и что «подобный подход с упругой деформацией гибкой рейки использует Уиттекер в работе [134] 1923 года». При этом теория расчёта изгиба балок была востребованной ещё перед строительством Эйфелевой башни в конце XIX века, и основы этой теории заложили Эйлер и Бернулли приблизительно в 1750-х годах [132]. В книге Шумайкера [135, с. 10] 1981 года также упоминается серия ранних работ (авторов Runge, Eagle, Quade и Colladz, Favard, выпущенных в 1901-м, 1928-м, 1938-м, 1940-м годах соответственно), в которых до Шёнберга используются сплайны, но не используется термин «сплайн».

Ещё одной основополагающей работой считается работа Kimeldorf и Wahba [136], в которой показывается связь (эквивалентность) интерполяционных и сглаживающих сплайнов с теорией воспроизводящих ядер гильбертова пространства и гауссовых процессов.

В первоисточнике на английском языке в монографии Альберга, Нильсона, Уолша 1967 года всюду используется термин approximation. В практически дословном переводе на русский язык этой монографии 1972 года используется термин «приближение». В связи с этим в математике часто аппроксимацию называют «приближением функций». На данный момент термин «аппроксимация» устойчиво укоренился в русском языке.

Обширный обзор научных работ по аппроксимирующим сплайнам (изучен весь список работ и их аннотация, цитирующих работу Kimeldorf и Wahba [136], и просмотрены более чем 100 полнотекстовых публикаций из этого списка) показал отсутствие готового решения той задачи, которая решается здесь (тем более в том виде, в котором она представлена тут). Научных публикаций других авторов и готовых решений для восстановления функции по интегралам, которые наблюдаются с погрешностью, обнаружить не удалось. Описанные ниже решения были получены самостоятельно, основываясь на великолепно написанной работе Green & Silverman [137].

Сглаживающие сплайны

В отличие от интерполяционного сплайна, который определяется через условия равенства, записанные в виде системы уравнений, в классическом сглаживающем сплайне (сглаживающем лишь сами значения) минимизируется сумма квадратов отклонений значений сплайна от наблюдений с учётом дополнительного штрафа на меру кривизны (о котором будет сказано чуть позже)

$$S(g) = \sum_{i=1}^n (y_i - g(t_i))^2 + \alpha \int_{t_1}^{t_n} (g''(t))^2 dt \rightarrow \min, \quad (2.21)$$

где n – количество наблюдений;

$\int_a^b (g''(x))^2 dx$ – штраф на нелинейность;

α – коэффициент сглаживания, также известный как коэффициент регуляризации Тихонова [114].

Сплайн $g(t)$ задаётся следующим способом. Первым делом на интересующей исследователя области аргумента t (на оси) задаются узлы сплайна $s_1 < s_2 < \dots < s_m$, где m – количество узлов. Эти узлы сплайна делят интересующую область аргумента t на $m - 1$ интервал. Обычно в качестве самого первого узла выбирают самое первое наблюдение $s_1 = t_1$, а в качестве последнего узла выбирают последнее наблюдение $s_m = t_n$. Само количество узлов m может быть как меньше, так и больше, чем количество наблюдений n . Если узлов столько же, сколько и наблюдений $m = n$, то можно все узлы располагать в точках наблюдений t_i (именно так делается в классических работах по сплайнам), но это необязательно. Другим распространённым способом является расположение узлов через равные расстояния.

Примечание – В литературе, посвящённой сглаживающим сплайнам, слагаемое $\alpha \int_a^b (g''(x))^2 dx$ называют штрафом на нелинейность (шероховатость, roughness), коэффициент α – просто сглаживающим параметром, однако в некоторых работах это второе слагаемое называют регуляризирующим слагаемым, а параметр α – регуляризирующим параметром [138]. В нашем случае параметр α также можно называть параметром регуляризации, так как в случае отсутствия этого параметра, когда $\alpha = 0$, задача восстановления по интегралам будет некорректно поставленной, так как может существовать бесконечное множество решений, а получающаяся в

результате аналитического решения матрица становится вырожденной и вычисление решения становится невозможным. При очень малых значениях α решение получается нестабильным.

Сам сплайн $g(t)$ является полиномом третьей степени на каждом интервале между узлами s_k и s_{k+1} , $k = 1, \dots, m$. При переходе с одного интервала на другой полином меняется. Но меняется так, чтобы обеспечивалась непрерывность сплайна и его первой и второй производной. Сам полином можно выражать по-разному, например как обычный полином третьей степени (через четыре коэффициента a_0, a_1, a_2, a_3). Однако есть более удобное представление только через два неизвестных значения: само значение сплайна в узле $g_k = g(s_k)$ и его вторую производную в узле $\gamma_k = g''(s_k)$, известное также как *value-second derivative representation* [137, с. 22–24] (также иногда встречается как представления через «наклоны» и «моменты» [130, с. 10], такая терминология кажется неудачной, сами авторы указанной работы отмечают, что моменты «не являются моментами балки в обычном смысле»)

$$g(t) = \frac{(t - s_k)g_{k+1} + (s_{k+1} - t)g_k}{s_{k+1} - s_k} - \frac{1}{6}(t - s_k)(s_{k+1} - t) \left\{ \left(1 + \frac{t - s_k}{s_{k+1} - s_k} \right) \gamma_{k+1} + \left(1 + \frac{s_{k+1} - t}{s_{k+1} - s_k} \right) \gamma_k \right\},$$

$$k: s_k \leq t \leq s_{k+1}. \quad (2.22)$$

В итоге сплайн полностью идентифицируется через $g = (g_1, \dots, g_m)^T$ – столбец значений сплайна, $\gamma = (\gamma_2, \dots, \gamma_{m-1})^T$ – столбец вторых производных (вторая производная в натуральном сплайне в начальной и последней точке равна нулю, $\gamma_1 = \gamma_m = 0$, а сплайн за границами первого и последнего узла превращается в прямую линию). Это представление удобно тем, что параметры сплайна имеют ясную интерпретацию в отличие от параметров полиномов (a_0, a_1, a_2, a_3), записанных в классическом виде.

Для данного представления условия непрерывности самой функции $g(s_k - 0) = g(s_k + 0)$ и её второй производной $g''(s_k - 0) = g''(s_k + 0)$ в точках сочленения (всего имеем $m - 2$ точек сочленения, $k = 2, \dots, m - 1$) выполняются автоматически. Однако условия непрерывности первой

производной автоматически не выполняются, нужно вводить дополнительные условия. Если взять первую производную от (2.22) и для некоторого k подставить значения $s_k + 0$ и $s_k - 0$, то получим:

$$g'(s_k + 0) = \frac{g_{k+1} - g_k}{s_{k+1} - s_k} - \frac{1}{6}(s_{k+1} - s_k)\{\gamma_{k+1} + 2\gamma_k\}, \quad (2.23)$$

$$g'(s_k - 0) = \frac{g_k - g_{k-1}}{s_k - s_{k-1}} + \frac{1}{6}(s_k - s_{k-1})\{2\gamma_k + \gamma_{k-1}\}. \quad (2.24)$$

Записывая условие равенства первых производных с обеих сторон от точек сочленения $g'(s_k + 0) = g'(s_k - 0)$, $k = 2, \dots, m - 1$, имеем систему $m - 2$ уравнений

$$\frac{g_{k+1} - g_k}{s_{k+1} - s_k} - \frac{g_k - g_{k-1}}{s_k - s_{k-1}} = \frac{1}{6}(s_{k+1} - s_k)\{\gamma_{k+1} + 2\gamma_k\} + \frac{1}{6}(s_k - s_{k-1})\{2\gamma_k + \gamma_{k-1}\},$$

$$k = 2, \dots, m - 1. \quad (2.25)$$

Данную систему уравнений компактно записывают с помощью матричного выражения

$$Q^T g = R\gamma, \quad (2.26)$$

где Q – трёхдиагональная матрица размерностью $m \times (m - 2)$

$$Q =$$

	2	3	...	$m - 1$
1	h_1^{-1}	0	...	0
2	$-h_1^{-1} - h_2^{-1}$	h_2^{-1}	...	0
3	h_2^{-1}	$-h_2^{-1} - h_3^{-1}$...	0
4	0	h_3^{-1}	...	0
...
$m - 2$		0	...	h_{m-2}^{-1}
$m - 1$	0	0	...	$-h_{m-2}^{-1} - h_{m-1}^{-1}$
m	0	0	...	h_{m-1}^{-1}

R – трёхдиагональная матрица размерностью $(m - 2) \times (m - 2)$

$$R =$$

	2	3	4	...	$m - 1$
2	$(h_1 + h_2)/3$	$h_2/6$	0	...	0
3	$h_2/6$	$(h_2 + h_3)/3$	$h_3/6$...	0
4	0	$h_3/6$	$(h_3 + h_4)/3$...	0
5	0	0	$h_4/6$...	0
...
$m - 2$	0	0	0	...	$h_{m-2}/6$
$m - 1$	0	0	0	...	$(h_{m-2} + h_{m-1})/3$

где $h_k = s_{k+1} - s_k$, $k = 1, \dots, m - 1$.

Благодаря условиям непрерывности $Q^T g = R\gamma$ можно выразить значения вторых производных γ через значения сплайна g или наоборот

$$\gamma = R^{-1}Q^T g. \quad (2.27)$$

Штраф на кривизну сплайна $\int_{t_1}^{t_n} (g''(x))^2 dx$ упрощается [113, с. 24–25] до операций с теми же матрицами

$$\int_{t_1}^{t_n} (g''(t))^2 dt = \gamma^T Q^T g = \gamma^T R\gamma = g^T [QR^{-1}Q^T]g = g^T K g, \quad (2.28)$$

где $K = QR^{-1}Q^T$ – симметричная матрица размерностью $m \times m$.

Примечание – Для расчетов используется следующее

$$\begin{aligned} \int_{t_1}^{t_n} (g''(t))^2 dt &= \int_{s_1}^{s_m} (g''(t))^2 dt = \int_{s_1}^{s_m} g''(t)g''(t)dt = \left| \begin{array}{l} u = g''(t); dv = g''(t)dt \\ du = g'''(t); v = g'(t) \end{array} \right| = \\ &= g''(t)g'(t)|_{s_1}^{s_m} - \int_{s_1}^{s_m} g'(t)g'''(t)dt = 0 - \int_{s_1}^{s_m} g'(t)g'''(t)dt = - \sum_{k=1}^{m-1} \frac{\gamma_{k+1} - \gamma_k}{t_{k+1} - t_k} \int_{s_k}^{s_{k+1}} g'(t)dt = \\ &= \sum_{k=1}^{m-1} \frac{\gamma_{k+1} - \gamma_k}{h_k} (g_k - g_{k+1}) = \sum_{k=2}^{m-1} \gamma_k \left(\frac{g_{k+1} - g_k}{h_k} - \frac{g_k - g_{k-1}}{h_{k-1}} \right) = \gamma^T Q^T g = g^T QR^{-1}Q^T g = g^T K g, \end{aligned}$$

здесь использовалось то, что $g''(s_1) = g''(s_m) = 0$ и $g'''(t) = (\gamma_{k+1} - \gamma_k)/(t_{k+1} - t_k) = \text{const}$.

Сплайновая коллокация

В нашем случае необходимо восстанавливать функцию не только по значениям, а по интегралам. Обобщим задачу на восстановление функции по разным функционалам, как непосредственно самим значениям, так и первым производным, вторым производным и определённым интегралам. Такое обобщение может пригодиться в будущем, так как модели механизмов образования событий могут быть разными, и некоторые события могут возвращать не интегралы, а производные. В случае, если функционалами будут являться какие-то другие значения, то описанный ниже способ может быть хорошим пособием, чтобы научиться восстанавливать функцию по этим разным функционалам.

Оптимизационная задача будет следующей

$$\begin{aligned}
S(g) = & \sum_{i=1}^{n_f} w_i^f (y_i - g(t_i))^2 + \mu \sum_{j=1}^{n_{df}} w_j^{df} (y'_j - g'(t_j))^2 + \nu \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} (y''_l - g''(t_l))^2 + \\
& + \psi \sum_{u=1}^{n_{int}} w_u^{int} \left(Y_u - \int_{t_u^a}^{t_u^b} g(t) dt \right)^2 + \alpha \int_{t_{start}}^{t_{end}} (g''(t))^2 dt,
\end{aligned} \tag{2.29}$$

где $w_i^f, w_j^{df}, w_l^{d^2f}, w_u^{int}$ – индивидуальные веса групп наблюдений;

μ (мю) – вес всей группы наблюдений первых производных;

ν (ню) – вес всей группы наблюдений вторых производных;

ψ (пси) – вес всей группы наблюдений интегралов;

последнее слагаемое – штраф на нелинейность (шероховатость);

α – коэффициент сглаживания (регуляризации);

t_{start} и t_{end} – соответственно границы, в которых происходит восстановление функции;

y_i, y'_j, y''_l, Y_u – наблюдаемые (с погрешностью) значения, значения первой и второй производной, значения определённых интегралов;

$t_i, t_j, t_l, t_u^a, t_u^b$ – моменты наблюдений и границы интегралов (при одинаковых индексах могут обозначать разное время);

$n_f, n_{df}, n_{d^2f}, n_{int}$ – количество соответствующих наблюдений.

Данную задачу можно рассматривать как многокритериальную, когда разным группам наблюдения ставится отдельный вес. Заметим, что отсутствует весовой коэффициент для группы наблюдений обычных значений, т.е. он предполагается равным единице, а все остальные коэффициенты тогда показывают вес по сравнению с этой первой группой наблюдений.

Сначала необходимо выразить используемые функционалы через параметры сплайна. Для этого потребуется рассчитать производные и интеграл от самого сплайна, заданного выражением (2.22). Удобно переписать выражение (2.22) более компактно

$$\begin{aligned}
g(t) = & \frac{h_k^{-t}}{h_k} g_{k+1} + \frac{h_k^{+t}}{h_k} g_k - \frac{h_k^{-t} h_k^{+t} (h_k + h_k^{-t})}{6h_k} \gamma_{k+1} - \frac{h_k^{-t} h_k^{+t} (h_k + h_k^{+t})}{6h_k} \gamma_k, \\
& k: s_k \leq t < s_{k+1};
\end{aligned} \tag{2.30}$$

где $h_k = s_{k+1} - s_k$ – шаг между узлами (возможно, разный);

$h_k^{-t} = t - s_k$ – расстояние от узла s_k до момента t ;

$h_k^{+t} = s_{k+1} - t$ – расстояние от момента t до следующего узла s_{k+1} .

Запись используемых в оптимизационной задаче функционалов будет следующей:

– Функционал $g(t_i)$ определяется по самой формуле (2.30).

– Функционал $g'(t_j)$ определяется следующим образом

$$g'(t_j) = \frac{g_{k+1}}{h_k} - \frac{g_k}{h_k} - \left(\frac{h_k}{6} - \frac{(h_k^{-t_j})^2}{2h_k} \right) \gamma_{k+1} + \left(\frac{h_k}{6} - \frac{(h_k^{+t_j})^2}{2h_k} \right) \gamma_k. \quad (2.31)$$

– Функционал $g''(t_l)$ определяется следующим образом

$$g''(t_l) = \frac{h_k^{-t_l}}{h_k} \gamma_{k+1} + \frac{h_k^{+t_l}}{h_k} \gamma_k. \quad (2.32)$$

Замечание. В выражениях (2.30), (2.31), (2.32) вместо t при расчёте h_k^{-t} и h_k^{+t} надо подставлять значения t_i , t_j и t_l соответственно. Во всех этих выражениях вначале надо определить, на какой интервал k выпали эти моменты наблюдения t_i , t_j и t_l ($k: s_k \leq t \leq s_{k+1}$).

– Расчёт функционала $\int_{t_u^a}^{t_u^b} g(t) dt$ происходит по сложной формуле (подробности в работах [202; 215, с. 286–290]). Вначале надо определить, на какой интервал k выпал нижний предел интегрирования t_u^a и на какой интервал $k + L$ выпал верхний предел интегрирования, где L – количество интервалов между ними (может быть равно нулю, если оба предела выпали на один интервал)

$$\begin{aligned}
\int_{t_u^a}^{t_u^b} g(t) dt = & \sum_{l=0}^L \left[\frac{h_{k+l}}{2} g_{k+l+1} + \frac{h_{k+l}}{2} g_{k+l} - \frac{h_{k+l}^3}{24} \gamma_{k+l+1} - \frac{h_{k+l}^3}{24} \gamma_{k+l} \right] \\
& - \frac{(h_k^{-t_u^a})^2}{2h_k} g_{k+1} - \frac{h_k^2 - (h_k^{+t_u^a})^2}{2h_k} g_k \\
& - \frac{(h_k^{-t_u^a})^2 \left((h_k^{-t_u^a})^2 - 2h_k^2 \right)}{24h_k} \gamma_{k+1} + \frac{(h_k^{-t_u^a})^2 (h_k^{+t_u^a} + h_k)^2}{24h_k} \gamma_k \\
& - \frac{h_{k+L}^2 - (h_{k+L}^{-t_u^b})^2}{2h_{k+L}} g_{k+L+1} - \frac{(h_{k+L}^{+t_u^b})^2}{2h_{k+L}} g_{k+L} \\
& + \frac{(h_{k+L}^{+t_u^b})^2 (h_{k+L}^{-t_u^b} + h_{k+L})^2}{24h_{k+L}} \gamma_{k+L+1} - \frac{(h_{k+L}^{+t_u^b})^2 \left((h_{k+L}^{+t_u^b})^2 - 2h_{k+L}^2 \right)}{24h_{k+L}} \gamma_{k+L},
\end{aligned} \tag{2.33}$$

$L: s_{k+L} < t_u^b \leq s_{k+L+1},$
 $k: s_k \leq t_u^a < s_{k+1},$

где $h_k^{-t_u^a} = t_u^a - s_k$, $h_k^{+t_u^a} = s_{k+1} - t_u^a$, $h_k = s_{k+1} - s_k$, $h_{k+L}^{-t_u^b} = t_u^b - s_{k+L}$, $h_{k+L}^{+t_u^b} = s_{k+L+1} - t_u^b$, $h_{k+L} = s_{k+L+1} - s_{k+L}$.

Все эти выражения (2.30), (2.31), (2.32), (2.33) линейны по отношению к неизвестным параметрам сплайна g_k и γ_k . Это позволяет выразить оптимизационную задачу (2.29) в матричном виде

$$\begin{aligned}
S(g) = & (Y_f - V_f g + P_f \gamma)^T W_f (Y_f - V_f g + P_f \gamma) + \\
& + \mu (Y_{df} - V_{df} g + P_{df} \gamma)^T W_{df} (Y_{df} - V_{df} g + P_{df} \gamma) + \\
& + \nu (Y_{d^2f} - 0g + P_{d^2f} \gamma)^T W_{d^2f} (Y_{d^2f} - 0g + P_{d^2f} \gamma) + \\
& + \psi (Y_{int} - V_{int} g + P_{int} \gamma)^T W_{int} (Y_{int} - V_{int} g + P_{int} \gamma) + \\
& + \alpha g^T K g \rightarrow \min.
\end{aligned} \tag{2.34}$$

Матрицы V_f , V_{df} , V_{int} , P_f , P_{df} , P_{d^2f} , P_{int} заполняются следующим образом:

V_f размерностью $n_f \times m$, её каждая i -тая строка выглядит как

$$\begin{array}{cccccccc}
1 & \dots & k-1 & k & k+1 & k+2 & \dots & m \\
0 & \dots & 0 & h_k^{+t_i}/h_k & h_k^{-t_i}/h_k & 0 & \dots & 0
\end{array}$$

P_f размерностью $n_f \times (m-2)$, её каждая i -тая строка выглядит как

$$\begin{array}{cccccccc}
2 & \dots & k-1 & k & k+1 & k+2 & \dots & m-1 \\
0 & \dots & 0 & h_k^{-t_i} h_k^{+t_i} (h_k + h_k^{+t_i}) / 6h_k & h_k^{-t_i} h_k^{+t_i} (h_k + h_k^{-t_i}) / 6h_k & 0 & \dots & 0
\end{array}$$

V_{df} размерностью $n_{df} \times m$, её каждая j -тая строка выглядит как

$$\begin{array}{cccccccc} \hline 1 & \dots & k-1 & k & k+1 & k+2 & \dots & m \\ \hline 0 & \dots & 0 & -1/h_k & 1/h_k & 0 & \dots & 0 \end{array}$$

P_{df} размерностью $n_{df} \times (m-2)$, её каждая j -тая строка выглядит как

$$\begin{array}{cccccccc} \hline 2 & \dots & k-1 & k & k+1 & k+2 & \dots & m-1 \\ \hline 0 & \dots & 0 & -h_k/6 + (h_k^{+t_j})^2/2h_k & h_k/6 + (h_k^{-t_j})^2/2h_k & 0 & \dots & 0 \end{array}$$

P_{d^2f} размерностью $n_{d^2f} \times (m-2)$, её каждая l -тая строка выглядит как

$$\begin{array}{cccccccc} \hline 2 & \dots & k-1 & k & k+1 & k+2 & \dots & m-1 \\ \hline 0 & \dots & 0 & -h_k^{+t_l}/h_k & -h_k^{-t_l}/h_k & 0 & \dots & 0 \end{array}$$

V_{int} размерностью $n_{int} \times m$, её каждая u -тая строка заполняется следующим образом

$$\begin{aligned} V_{u,k} &= \frac{(h_k^{+t_u^a})^2}{2h_k}; V_{u,k+l} = \frac{h_{k+l-1} + h_{k+l}}{2}, l = 1, \dots, L; V_{u,k+1} = V_{u,k+1} - \frac{(h_k^{-t_u^a})^2}{2h_k}; \\ V_{u,k+L} &= V_{u,k+L} - \frac{(h_{k+L}^{+t_u^b})^2}{2h_{k+L}}; V_{u,k+L+1} = \frac{(h_{k+L}^{-t_u^b})^2}{2h_{k+L}}. \end{aligned} \quad (2.35)$$

Например, в случае если $L > 2$, каждый u -тая строка выглядит как

$$\begin{array}{cccccccc} \hline k-1 & k & k+1 & k+2 & \dots k+l \dots & k+L & k+L+1 & k+L+2 \\ \hline 0 & \frac{(h_k^{+t_u^a})^2}{2h_k} & \left(\frac{h_k + h_{k+1}}{2} \right) & \frac{h_{k+1} + h_{k+2}}{2} & \frac{h_{k+l-1} + h_{k+l}}{2} & \left(\frac{h_{k+L-1} + h_{k+L}}{2} \right) & \frac{(h_{k+L}^{-t_u^b})^2}{2h_{k+L}} & 0 \end{array}$$

P_{int} размерностью $n_{int} \times (m-2)$, её каждая u -тая строка заполняется следующим образом

$$\begin{aligned}
P_{u,k} &= \frac{h_k^3}{24} - \frac{(h_k^{-t_u^a})^2 (h_k^{+t_u^a} + h_k)^2}{24h_k}; P_{u,k+l} = \frac{h_{k+l-1}^3 + h_{k+l}^3}{24}, l = 1, \dots, L; \\
P_{u,k+1} &= P_{u,k+1} + \frac{(h_k^{-t_u^a})^2 ((h_k^{-t_u^a})^2 - 2h_k^2)}{24h_k}; \\
P_{u,k+L} &= P_{u,k+L} + \frac{(h_{k+L}^{+t_u^b})^2 ((h_{k+L}^{+t_u^b})^2 - 2h_{k+L}^2)}{24h_{k+L}}; \\
P_{u,k+L+1} &= \frac{h_{k+L}^3}{24} - \frac{(h_{k+L}^{+t_u^b})^2 (h_{k+L}^{-t_u^b} + h_{k+L})^2}{24h_{k+L}}.
\end{aligned} \tag{2.36}$$

Например, в случае если $L > 2$, каждый u -тая строка выглядит как

$k - 1$	k	$k + 1$	$k + 2$
0	$\left(\frac{\frac{h_k^3}{24} - \frac{(h_k^{-t_u^a})^2 (h_k^{+t_u^a} + h_k)^2}{24h_k}}{\frac{(h_k^{-t_u^a})^2 ((h_k^{-t_u^a})^2 - 2h_k^2)}{24h_k}} \right)$	$\left(\frac{\frac{h_k^3 + h_{k+1}^3}{24} + \frac{(h_k^{-t_u^a})^2 ((h_k^{-t_u^a})^2 - 2h_k^2)}{24h_k}}{\frac{(h_k^{-t_u^a})^2 (h_{k+L}^{-t_u^b} + h_{k+L})^2}{24h_{k+L}}} \right)$	$\frac{h_{k+1}^3 + h_{k+2}^3}{24}$
$\dots k + l \dots$	$k + L$	$k + L + 1$	$k + L + 2$
$\frac{h_{k+L-1}^3 + h_{k+L}^3}{24}$	$\left(\frac{\frac{h_{k+L-1}^3 + h_{k+L}^3}{24} + \frac{(h_{k+L}^{+t_u^b})^2 ((h_{k+L}^{+t_u^b})^2 - 2h_{k+L}^2)}{24h_{k+L}}}{\frac{(h_{k+L}^{+t_u^b})^2 (h_{k+L}^{-t_u^b} + h_{k+L})^2}{24h_{k+L}}} \right)$	$\left(\frac{\frac{h_{k+L}^3}{24} - \frac{(h_{k+L}^{+t_u^b})^2 (h_{k+L}^{-t_u^b} + h_{k+L})^2}{24h_{k+L}}}{\frac{(h_{k+L}^{+t_u^b})^2 (h_{k+L}^{-t_u^b} + h_{k+L})^2}{24h_{k+L}}} \right)$	0

Замечание. Во всех этих выражениях номер узла k определяется независимо для каждого наблюдения (в каждом ряду, $k: s_k \leq t \leq s_{k+1}$).

Матрица $K = QR^{-1}Q^T$ симметричная матрица размерностью $m \times m$.

Матрицы $W_f, W_{df}, W_{d^2f}, W_{int}$ являются диагональными матрицами индивидуальных весов наблюдений, размерностями $n_f \times n_f, n_{df} \times n_{df}, n_{d^2f} \times n_{d^2f}$ и $n_{int} \times n_{int}$ соответственно.

Для продолжения решения необходимо заменить одну из переменных с помощью условий непрерывности (2.26) $Q^T g = R\gamma$, позволяющих выразить $\gamma = R^{-1}Q^T g$. В результате остаётся только одна неизвестная g .

$$\begin{aligned}
S(g) &= (Y_f - C_f g)^T W_f (Y_f - C_f g) \\
&+ \mu (Y_{df} - C_{df} g)^T W_{df} (Y_{df} - C_{df} g) \\
&+ \nu (Y_{d^2f} - C_{d^2f} g)^T W_{d^2f} (Y_{d^2f} - C_{d^2f} g) \\
&+ \psi (Y_{int} - C_{int} g)^T W_{int} (Y_{int} - C_{int} g) \\
&+ \alpha g^T K g \rightarrow \min,
\end{aligned} \tag{2.37}$$

где $C_f = V_f - P_f R^{-1} Q^T$, $C_{df} = V_{df} - P_{df} R^{-1} Q^T$, $C_{d^2f} = 0 - P_{d^2f} R^{-1} Q^T$,
 $C_{int} = V_{int} - P_{int} R^{-1} Q^T$ – матрицы размерностями $n_f \times m$, $n_{df} \times m$, $n_{d^2f} \times m$,
 $n_{int} \times m$ соответственно.

Транспонируя, раскрывая скобки и приравнявая производную по g к нулю, находим выражение для искомых параметров g

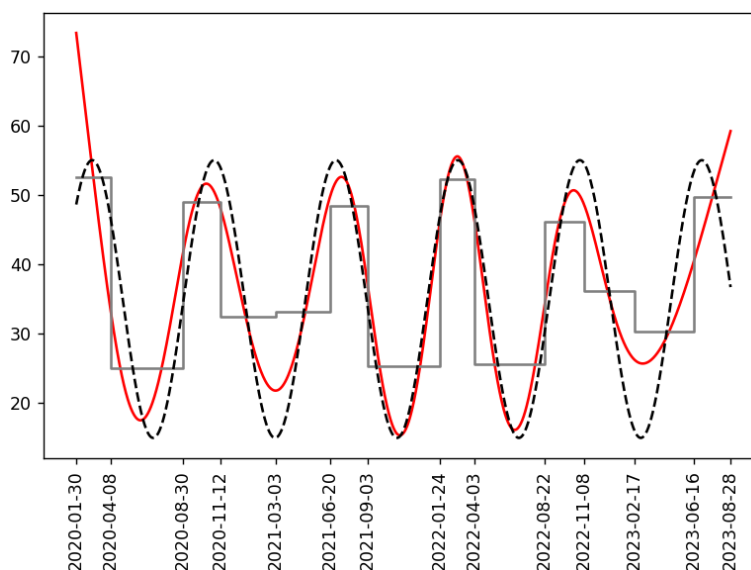
$$g = (C_f^T W_f C_f + \mu C_{df}^T W_{df} C_{df} + \nu C_{d^2f}^T W_{d^2f} C_{d^2f} + \psi C_{int}^T W_{int} C_{int} + \alpha K)^{-1} \\ \times (C_f^T W_f Y_f + \mu C_{df}^T W_{df} Y_{df} + \nu C_{d^2f}^T W_{d^2f} Y_{d^2f} + \psi C_{int}^T W_{int} Y_{int}) \quad (2.38)$$

После получения значений g необходимо рассчитать значения вторых производных $\gamma = R^{-1} Q^T g$, далее можно рассчитывать значение сплайна $g(t)$ в любой точке t по выражению (2.22) или (2.30).

Пример

Как в примере для регрессионных методов и методов машинного обучения, рассмотрим данные моделирования системы управления запасами, таблица 2.1. В нашем случае имеются лишь данные наблюдений за определёнными интегралами. Поэтому надо заполнить лишь матрицы V_{int} и P_{int} . Количество узлов сплайна m смело можно брать в несколько раз больше, чем количество наблюдений, так как штраф на меру кривизны (шероховатость) не позволит функции быть слишком «шероховатой». В нашем случае берём $m = 3 \cdot n_{int} = 3 \cdot 14 = 42$ узла и располагаем их равномерно между первым и последним наблюдением. Коэффициент сглаживания можно взять $\alpha = 10^3$, так как квадраты отклонений интегралов, измеряемых тысячами, могут иметь большое значение (на самом деле достаточно брать любое не очень маленькое значение, при котором матрица перестаёт быть сингулярной. Качество восстановления в этом примере мало отличается что при $\alpha = 10^0$, что при $\alpha = 10^5$. Но при очень большом коэффициенте сглаживания наблюдения уже начинают заметно приглушаться). Подробнее о выборе коэффициента сглаживания будет ниже. Результат восстановления функции по последовательности интегралов

изображен на рисунке 2.9, можно сравнить данный рисунок с рисунками 2.3–2.8 и с тем, что ожидалось получить на рисунке 1.11.



Ступеньки (горизонтальные отрезки) показывают среднее значение функции на этом интервале $y_i/(t_{i+1} - t_i)$, площадь под каждой из ступенек (до оси ноль) – наблюдаемый интеграл y_i . Пунктирная линия – исходная функция, заложенная при моделировании. Сплошная линия – восстановленная по интегралам функция.

Источник: составлено автором.

Рисунок 2.9 – Применение сплайновой коллокации для восстановления параметра $P_1(t)$ по последовательности интегралов

Как видим, сплайновой коллокацией получилось заметно лучше восстановить параметр механизма образования событий $P_1(t)$ по наблюдаемым с погрешностью интегралам. Отчётливо угадывается гармоническая функция, заложенная в качестве нестационарного спроса в модели управления запасами. Самое лучшее восстановление происходит в центре. На краях качество восстановления функции слабее (из-за недостатка данных, сплайн не знает, что надо было поворачивать вниз).

Замечание по выбору оптимального коэффициента сглаживания α

Можно выбирать параметр сглаживания вручную (на глаз, чтобы восстановленная функция визуально хорошо соответствовала имеющимся данным), но существуют специальные процедуры автоматического расчёта данного параметра на основе входных данных. Для этого используют либо метод L -кривой [139; 140], либо метод кросс-валидации [141; 137, с. 30].

Автором данной работы проведено подробное исследование возможности применения этих подходов для задачи восстановления функции по интегралам [210]. Оказалось, что метод L -кривой не подходит.

Идея метода кросс-валидации заключается в том, чтобы минимизировать квадраты отклонений наблюдений и значений модели, построенной по выборке без этих наблюдений. То есть из выборки выбрасывается сперва первое наблюдение, строится модель (оцениваются её параметры) по выборке без этого первого наблюдения, рассчитывается квадрат отклонения для этого первого наблюдения. Затем выбрасывается второе наблюдение, оценивается модель по выборке без второго наблюдения, и опять рассчитывается квадрат отклонения этого второго наблюдения. И так далее со всеми наблюдениями. Сумма таких квадратов отклонения и является оценкой значения кросс-валидации при заданном α .

Такая процедура предполагает повторение процедуры оценки параметров модели столько же раз, сколько есть наблюдений, что может быть трудоёмко. Однако существует способ рассчитать эти погрешности пропущенных наблюдений по оценённой лишь один раз по всей выборке модели. Подробности в работе [210] или основополагающей работе [141].

Выражение для оценки значения кросс-валидации $CV(\alpha)$ при заданном коэффициенте сглаживания α будет следующим

$$CV(\alpha) = n_f^{-1} \sum_{i=1}^{n_f} w_i^f \left(\frac{y_i - \sum_{k=1}^m C_{ik}^f g_k}{1 - \sum_{k=1}^m C_{ik}^f A_{ki}^f(\alpha)} \right)^2 + n_{df}^{-1} \sum_{j=1}^{n_{df}} w_j^{df} \left(\frac{y_j' - \sum_{k=1}^m C_{jk}^{df} g_k}{1 - \sum_{k=1}^m C_{jk}^{df} A_{kj}^{df}(\alpha)} \right)^2 + \\ + n_{d^2f}^{-1} \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} \left(\frac{y_l'' - \sum_{k=1}^m C_{lk}^{d^2f} g_k}{1 - \sum_{k=1}^m C_{lk}^{d^2f} A_{kl}^{d^2f}(\alpha)} \right)^2 + n_{int}^{-1} \sum_{u=1}^{n_{int}} w_u^{int} \left(\frac{Y_u - \sum_{k=1}^m C_{uk}^{int} g_k}{1 - \sum_{k=1}^m C_{uk}^{int} A_{ku}^{int}(\alpha)} \right)^2, \quad (2.39)$$

где матрицы C^f , C^{df} , C^{d^2f} , C^{int} такие же, как в формуле (2.37).

Матрицы $A^f(\alpha) = A(\alpha)C_f^T W_f$, $A^{df}(\alpha) = A(\alpha)\mu C_{df}^T W_{df}$, $A^{d^2f}(\alpha) = A(\alpha)\nu C_{d^2f}^T W_{d^2f}$, $A^{int}(\alpha) = A(\alpha)\psi C_{int}^T W_{int}$, где $A(\alpha) = (C_f^T W_f C_f + \mu C_{df}^T W_{df} C_{df} + \nu C_{d^2f}^T W_{d^2f} C_{d^2f} + \psi C_{int}^T W_{int} C_{int} + \alpha K)^{-1}$.

Минимизация $CV(\alpha)$ любым известным способом относительно α даёт искомое значение параметра сглаживания α .

В идеале метод кросс-валидации позволяет найти такой параметр сглаживания α , который делает процедуру оценки функции эффективной с точки зрения предсказания будущих наблюдений (прогноз обладает наименьшей дисперсией). Но проблема в том, что при исключении некоторых наблюдений может катастрофически сильно меняться вид восстановленной функции. Каждое наблюдение несёт крайне необходимую информацию для восстановления функции, исключение наблюдения из выборки может привести к очень и очень большим ошибкам. Метод кросс-валидации может быть хорошим решением для определения параметра сглаживания α тогда, когда одни наблюдения лишь статистически дополняют информацию других наблюдений и исключение одного из наблюдений не приводит к значительным изменениям восстановленной функции.

В нашем случае функция восстанавливается по небольшой последовательности интегралов. В проведённом исследовании [210] выяснилось, что метод кросс-валидации работает только в половине случаев, в другой половине приводит либо к очень сильному занижению, либо к завышению коэффициента сглаживания.

Другим подходом является принцип невязки Морозова (Morozov's discrepancy principle) [142; 143, с. 72] или, что то же самое, рекомендация Рейнша [144], в которых предлагается брать коэффициент сглаживания α на таком уровне, чтобы он сглаживал наблюдения на величину, соответствующую уровню шума σ_ε в исходных данных.

В проведённом исследовании [210] оказалось, что наилучшее восстановление функции по интегралам будет при таком коэффициенте сглаживания, который бы приглушал исходные наблюдения приблизительно на 65% от уровня шума. Например, если уровень шума исходных наблюдений σ_ε ожидается как 5%, то надо выбрать такой коэффициент сглаживания α , чтобы после сглаживания наблюдения приглушились на $0,65 \cdot 5 = 3,25\%$.

Под приглушением наблюдений понимается, на сколько процентов уменьшилось прогнозное значение наблюдения \hat{y}_i относительно исходного y_i . Однако, так как изначально предполагается, что именно исходные наблюдения y_i наблюдаются с погрешностью, то прогнозные значения \hat{y}_i оказываются ближе к неизвестным истинным значениям. Поэтому приглушение лучше отсчитывать относительно прогнозных значений \hat{y}_i

$$\hat{y}_i = [Cg]_i = \sum_{k=1}^m C_{ik}g_k, \quad (2.40)$$

$$\sigma_y(\alpha) = \left(n^{-1} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{\hat{y}_i} \right)^2 \right)^{0,5} \times 100\%, \quad (2.41)$$

где под матрицей C понимается одна из матриц $C_f, C_{df}, C_{d^2f}, C_{int}$, а под n – соответствующее количество наблюдений (в работе [210] проводилось исследование наилучшего восстановления функции именно по интегралам, рекомендация 65% от уровня шума σ_ε относится к интегралам, т.е. использовать надо матрицу C_{int}).

С другой стороны, коэффициент сглаживания у нас будет не единственным гиперпараметром. Да и сама задача заключается не только в восстановлении функции, а в прогнозировании событий, которые образуются в результате функционирования модели образования событий. Вопрос подбора гиперпараметров обсуждается чуть ниже.

2.6 Монотонная/положительная сплайновая коллокация

Во многих прикладных задачах, в том числе нашей, часто требуется, чтобы функция принимала лишь неотрицательные значения или была монотонно возрастающей. Под монотонностью функции можно понимать неотрицательность её производной. Термины «монотонность» (строгая или нет) и «неотрицательность/положительность функции» идут рука об руку.

По монотонным/положительным сглаживающим сплайнам существует незначительное количество научных работ. В то же время аналогичная задача иногда встречается под другими названиями, такими как: «вычисление

сглаживающих сплайн-функций с ограничениями» («computing constrained smoothing spline functions») [145], «интерполяция с ограничениями» («constrained interpolation») [146], «аппроксимация сплайнами заданной формы» («approximation by spline of given shape») [147]. В работе [147] даётся обширный, но беглый обзор существующей на то время литературы по подобным задачам.

Изучение имеющихся работ показало, что существует три подхода по обеспечению неотрицательности и монотонности функции.

1) Часто цитируемой работой по монотонным сплайнам является работа Ramsay J.O. [148], хотя существовали и более ранние работы по монотонным сплайнам, например [149]. Согласно этой ранней работе Ramsay сплайн представляется в виде линейной комбинации заранее известных неотрицательных полиномиальных функций $M_k(x)$ с неизвестными и подлежащими определению положительными коэффициентами a_k

$$g(x) = \sum_{j=k}^m a_k M_k(x). \quad (2.42)$$

Примечание – Функциями $M_k(x)$ могут быть полиномы степени r , определённые рекурсивно как разделённые разности [150]

$$M_k(x, r) = \frac{r[(x - s_k)M_k(x, r - 1) + (s_{k+r} - x)M_{k+1}(x, r - 1)]}{(r - 1)(s_{k+m} - s_k)},$$

где $M_k(x, 1) = 1/(s_{k+1} - s_k)$; r – степень полинома; s_k – узлы сплайна, каждая функция $M_k(x, r)$ определена на интервале $s_k \leq x < s_{k+r}$ (равна 0 вне этого интервала).

Если восстановленная функция $g(x)$ является производной некоторой $F(x)$, тогда $F(x)$ может быть выражена как линейная комбинация заранее известных интегральных функций $I_k(x)$, являющихся полиномами на единицу большей степени, чем полиномы $M_k(x)$

$$F(x) = \sum_{j=k}^m a_k I_k(x), \quad (2.43)$$

$$I_k(x) = \int_{-\infty}^x M_k(x) dx. \quad (2.44)$$

Недостаток данного подхода в том, что если вычислять коэффициенты a_k математически, как это делается в линейной регрессии ($A = (M^T M)^{-1} M^T Y$, где M – матрица, каждый элемент $M_{i,k}$ которой содержит значения $M_k(x_i)$, Y – столбец наблюдений сглаживаемых значений f_i), некоторые коэффициенты a_k весьма вероятно окажутся отрицательными. Для обеспечения неотрицательности Ramsey добавляет условия $a_k \geq 0, \sum a_k = 1$ и использует итерационный алгоритм, схожий с градиентным спуском. Так как такой итерационный алгоритм использует критерии останова, основанные на достижении заданной точности, получаемые решения оказываются приближёнными, а не точными. Другой недостаток в том, что такое представление сплайна неинформативно. В заданный момент времени активно сразу несколько полиномов (базисных функций), которые смешиваются с помощью коэффициентов a_k . Эти коэффициенты сложно проинтерпретировать с точки зрения прикладной задачи (этот подход не будем подробно рассматривать).

2) Другой подход основывается на простой идее, а именно поместить сплайн в степень экспоненты (более поздняя работа Ramsay и Silverman [151, с. 111–126; 152], интересно, что Ramsay упоминает только второй способ и не упоминает первый способ, опубликованный в его ранней работе)

$$f(x) = e^{g(x)}. \quad (2.45)$$

Сплайн $g(x)$ может принимать как положительные, так и отрицательные значения, возведение экспоненты в любую степень всегда даёт положительное значение. Сам сплайн $g(x)$ может быть задан как угодно.

Недостаток данного подхода в том, что получается нелинейная оптимизационная задача, решение которой предполагается численными методами, дающими приближённое решение. Также параметры сплайна имеют плохую интерпретацию с исходной практической задачей. Другим недостатком является то, что восстанавливаемая функция получается всё ещё

«некрасивой». Заметим, что линейризация логарифмированием не всегда доступна.

3) Ещё один подход основывается на конструировании задачи квадратичного программирования с помощью явного добавления условий неотрицательности сплайна в узлах этого сплайна. Этот подход будет подробно рассмотрен ниже.

Преимущество этого подхода – в возможности получения точного математического решения. В то же время физический смысл параметров сплайна остаётся неизменным, что удобно при интерпретации результатов.

Однако есть существенный недостаток. Условие неотрицательности сплайна применяется только в узлах сплайна (конечное множество), а не к произвольной точке сплайна. Сплайн всё ещё может принять отрицательные значения между узлами. Добавление дополнительных узлов с условиями неотрицательности в них может в некоторой степени понизить вероятность этого.

Мы рассмотрим только второй и третий способ обеспечения неотрицательности (монотонности).

Экспонента от сплайна

В этом способе сплайн помещается в степень экспоненты $f(x) = e^{g(x)}$.

Случай линейризации. Начнём с простейшего случая, когда задача позволяет взять логарифм и привести задачу к линейному виду

$$\log y_i = g(t_i). \quad (2.46)$$

Однако первая $f'(x)$ и вторая производная $f''(x)$ могут быть отрицательными, поэтому линейризовать логарифмом задачу не получается (логарифм отрицательных чисел не существует). Определённый интеграл от экспоненты в степени кубического сплайна также не линейризуется. В результате линейризация может быть использована для восстановления функции только по значениям, игнорируя другую информацию. В этом случае решается оптимизационная задача, которая является частным случаем большой задачи (2.29).

$$\sum_{i=1}^{n_f} w_i^f (\log y_i - g(t_i))^2 + \alpha \int_{t_{start}}^{t_{end}} (g''(t))^2 dt \rightarrow \min. \quad (2.47)$$

Решением является ранее полученное выражение (2.38), где подставляются логарифмы $\log y_i$ вместо наблюдений y_i , а данные первых производных, вторых производных и интегралов игнорируются.

Следует заметить, что отклонения логарифмов $\log y_i - g(t_i)$, как правило, имеют намного меньшие значения, чем отклонения нелогарифмированных величин $y_i - e^{g(t_i)}$. В результате коэффициент сглаживания α следует уменьшать, причём приблизительно во столько раз, во сколько значение логарифмов $\log y_i$ меньше значений наблюдения y_i .

Нелинейный случай. В этом случае функция потерь минимизируется численными методами. Однако интеграл от экспоненты в степени кубического сплайна не может быть вычислен просто. Поэтому вынуждены отбросить и не использовать данные об интегралах. Функция потерь будет следующей

$$S(g) = \sum_{i=1}^{n_f} w_i^f (y_i - e^{g(t_i)})^2 + \mu \sum_{j=1}^{n_{df}} w_j^{df} (y_j' - (e^{g(t_j)}))')^2 + \nu \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} (y_l'' - (e^{g(t_l)}))'')^2 + \alpha \int_{t_{start}}^{t_{end}} (g''(t))^2 dt \rightarrow \min. \quad (2.48)$$

После взятия производной

$$\begin{aligned} S(g) &= \sum_{i=1}^{n_f} w_i^f (y_i - e^{g(t_i)})^2 \\ &+ \mu \sum_{j=1}^{n_{df}} w_j^{df} (y_j' - e^{g(t_j)} g'(t_j))^2 \\ &+ \nu \sum_{l=1}^{n_{d^2f}} w_l^{d^2f} \left(y_l'' - e^{g(t_l)} \left((g'(t_l))^2 + g''(t_l) \right) \right)^2 \\ &+ \alpha \int_{t_{start}}^{t_{end}} (g''(t))^2 dt \rightarrow \min. \end{aligned} \quad (2.49)$$

Для выполнения вычислений необходимо проделать следующее.

- Для заданных значений $g = (g_1, \dots, g_m)^T$ (инициализировать можно нулями) рассчитать $\gamma = R^{-1}Q^T g$ (не забывая, что индексация начинается с 2, $\gamma = (\gamma_2, \dots, \gamma_{m-1})^T$).
- Рассчитать $g(t_i)$ по формуле (2.30) в точках t_i и вычислить экспоненту от результата.
- Рассчитать $g(t_j)$ по формуле (2.30) и $g'(t_j)$ по формуле (2.31) в точках t_j , взять экспоненту от $g(t_j)$ и умножить результат на $g'(t_j)$.
- Рассчитать $g(t_l)$ по формуле (2.30), рассчитать $g'(t_l)$ по формуле (2.31) и рассчитать $g''(t_l)$ по формуле (2.32) в точках t_l , затем рассчитать $e^{g(t_l)} \left((g'(t_l))^2 + g''(t_l) \right)$.
- Рассчитать штраф по формуле $\alpha g^T K g$.

Мы не обсуждаем здесь, какие численные методы оптимизации следует использовать. Но в примерах ниже используются стандартные средства оптимизации Python (библиотека `scipy`, метод `optimize.minimize`), в которых по умолчанию используется квазиньютоновский метод BFGS (Broyden, Fletcher, Goldfarb и Shanno [153, с. 136]).

Стоит заметить, что штраф показывает меру кривизны для степени экспоненты, а не для самой восстанавливаемой функции.

Сглаживающий сплайн и квадратичное программирование

Другой подход заключается в добавлении условий неотрицательности в узлах сплайна $g(s_k) = g_k \geq 0$ непосредственно к самой оптимизационной задаче. В результате исходная задача (2.29) сводится к задаче квадратичного программирования

$$\begin{aligned} S(g) \rightarrow \min, \\ g_k \geq 0, \quad k = 1, \dots, m. \end{aligned} \tag{2.50}$$

Здесь очень кстати пришлось представление сплайна через значения и значения второй производной (value-second derivative representation). Это позволяет напрямую наложить условия неотрицательности на параметры g_k , являющиеся значениями сплайна в узлах $g(s_k) = g_k$.

Классическая задача квадратичного программирования имеет вид

$$\begin{aligned} Q(x) = c^T x + \frac{1}{2} x^T D x \rightarrow \min, \\ Ax \geq b, \\ x \geq 0, \end{aligned} \quad (2.51)$$

где x – вектор-столбец искомых значений размерностью $m \times 1$ (в наших обозначениях);

D – симметричная матрица размерностью $m \times m$;

c – вектор-столбец значений размерностью $m \times 1$;

A – матрица коэффициентов при переменных x в p ограничениях размерностью $p \times m$;

b – вектор-столбец значений размерностью $p \times 1$.

В нашем случае отсутствуют ограничения $Ax \geq b$. Необходимо выразить матрицу D и столбец c , основываясь на нашей задаче (2.37). В задаче (2.37) сумма квадратов в матричном виде, например первое слагаемое, даёт

$$\begin{aligned} (Y_f - C_f g)^T W_f (Y_f - C_f g) &= -g^T C_f^T W_f Y_f - Y_f^T W_f C_f g + g^T C_f^T W_f C_f g + Y_f^T W_f Y_f = \\ &= -2Y_f^T W_f C_f g + g^T C_f^T W_f C_f g + Y_f^T W_f Y_f, \end{aligned} \quad (2.52)$$

где $Y_f^T W_f Y_f$ не зависит от g и поэтому не участвует в оптимизации (при получении выражения (2.38) из (2.37) оно обнулялось во время взятия производной по g).

Однако имеется несколько таких сумм квадратов. Тем не менее все произведения матриц вида $C_f^T W_f Y_f$ имеют одинаковую размерность $m \times 1$, и произведения матриц вида $C_f^T W_f C_f$ имеют размерность $m \times m$, матрица K также имеет размерность $m \times m$. Поэтому оптимизационная задача (2.37) может быть записана как (дополнительно поделив на 2)

$$\begin{aligned} S(g) = -(C_f^T W_f Y_f + \mu C_{df}^T W_{df} Y_{df} + \nu C_{d^2f}^T W_{d^2f} Y_{d^2f} + \psi C_{int}^T W_{int} Y_{int})^T g + \\ + \frac{1}{2} g^T (C_f^T W_f C_f + \mu C_{df}^T W_{df} C_{df} + \nu C_{d^2f}^T W_{d^2f} C_{d^2f} + \psi C_{int}^T W_{int} C_{int} + \alpha K) g, \end{aligned} \quad (2.53)$$

$$S(g) = c^T g + \frac{1}{2} g^T D g \rightarrow \min, \quad (2.54)$$

где $c = -(C_f^T W_f Y_f + \mu C_{df}^T W_{df} Y_{df} + \nu C_{d^2f}^T W_{d^2f} Y_{d^2f} + \psi C_{int}^T W_{int} Y_{int})$,

$$D = C_f^T W_f C_f + \mu C_{df}^T W_{df} C_{df} + \nu C_{d^2f}^T W_{d^2f} C_{d^2f} + \psi C_{int}^T W_{int} C_{int} + \alpha K.$$

Выражение (2.54) имеет классическую форму задачи квадратичного программирования (2.51) с ограничениями $g \geq 0$ (или $-g \leq 0$).

Аналитическое решение задачи квадратичного программирования основывается на методе множителей Лагранжа $\lambda = (\lambda_1, \dots, \lambda_m)^T$. Производная от лагранжиана L относительно неизвестных g приравнивается к нулю, и добавляются условия теоремы Каруша-Куна-Такера.

$$L = c^T g + \frac{1}{2} g^T D g + \lambda^T (-g) \quad (2.55)$$

$$\begin{cases} \frac{\partial L}{\partial g} = (c^T)^T + Dg - (\lambda^T)^T = 0 \\ \lambda^T (-g) = 0 \end{cases} \quad (2.56)$$

$$\begin{cases} c + Dg - \lambda = 0 \\ \lambda^T g = 0 \end{cases} \quad (2.57)$$

Полученная система уравнений относится к линейной задаче о дополнителности (linear complementarity problem, LCP), которая решается методом Лемке. Идея метода схожа с симплекс-методом, с той разницей, что включаемая в базис переменная не определяется, глядя на коэффициенты строки целевой функции (такой строки нет), а включаемая переменная является дополняющей к исключаемой из базиса переменной. Подробно и наглядно о квадратичном программировании и алгоритме Лемке написано в [25], подробнее о линейной задаче о дополнителности в [154]. Стоит заметить, что задачи квадратичного программирования, помимо алгоритма Лемке, могут быть решены другими способами, например, на основе метода внутренней точки, который является итеративным методом, дающим приближённый результат.

Примечание – Линейная задача о дополнителности имеет стандартный вид $\begin{cases} w - Mz = q \\ w^T z = 0 \end{cases}$, где в наших обозначениях $w = \lambda$, $M = D$, $q = c$, и искомые неизвестные $z = g$.

В Python существует много библиотек для решения задачи квадратичного программирования, например CVXOPT, qROASES, quadprog и т.д. [155].

Ещё раз хочется обратить внимание на преимущества этого подхода. Возможно использовать все данные, включая первые и вторые производные и

определённые интегралы. Решение является точным, а не приближённым (как в градиентных методах, правда результат точного решения всё равно используется в приближении/сглаживании известных точек данных).

Однако есть и недостатки. Восстановленная функция всё ещё может принимать отрицательные значения между узлами сплайна, так как условия неотрицательности налагаются только в узлах сплайна.

Обеспечение положительности

Чтобы гарантировать неотрицательность, можно использовать описанный выше подход, когда сплайн помещается в степень экспоненты. Но он заметно видоизменяет восстанавливаемую функцию. Восстановленная функция перестаёт быть полиномом.

Другим решением может быть добавление новых узлов между двумя старыми узлами или добавление нового условия неотрицательности без добавления нового узла. Рассмотрим этот подход более подробно.

Между двумя узлами у полинома третьей степени возможна только одна экстремальная точка минимума. Для нахождения точек экстремума можно приравнять выражение (2.31) для $g'(t)$ к нулю, получив квадратное уравнение, и решить его относительно t . Но более компактным и удобным будет решение квадратного уравнения относительно Δt для $g'(s_k + \Delta t) = g'_k + g''_k \Delta t + g'''_k \Delta t^2 / 2 = 0$, где из (2.31) значение $g'_k = g'(s_k) = (g_{k+1} - g_k) / h_k - (\gamma_{k+1} + 2\gamma_k) h_k / 6$, а $g''_k = \gamma_k$, и $g'''_k = (\gamma_{k+1} - \gamma_k) / h_k$. Интересуют только действительные корни, поэтому при отрицательном $d = \gamma_k^2 - 2g'_k g'''_k$ переходим к следующей паре узлов s_k, s_{k+1} . Для $d \geq 0$ находим Δt (два значения)

$$\Delta t = (-\gamma_k \pm \sqrt{d}) / g'''_k. \quad (2.58)$$

Если Δt находится в интервале от 0 до h_k , рассчитываем значение сплайна $g(s_k + \Delta t)$ по формуле (2.30), иначе опять переходим к следующей паре узлов. Если рассчитанное значение $g(s_k + \Delta t)$ меньше нуля (или некоторого порога, который захочет установить исследователь, например

минус 0.0001), то в точке $s_k + \Delta t$ добавляем новый узел или новое условие неотрицательности без добавления нового узла. Проверив таким образом все соседние пары узлов и добавив при необходимости между ними новые узлы или условия неотрицательности, необходимо вновь произвести расчёт параметров сплайна g и γ .

Для добавления новых узлов просто изменяем последовательность $\{s_k\}$, чтобы в точке $s_k + \Delta t$ появился новый узел. После этого необходимо обновить все другие векторы и матрицы, использующиеся в расчёте (произвести все расчёты с самого начала).

Другим способом является добавление новых условий неотрицательности без добавления новых узлов. Такая идея используется в работе [156], посвящённой кубическому монотонному сплайну. Правда, в ней неотрицательность обеспечивается для первой производной, которая между узлами уже будет являться полиномом второй степени, а не третьей. В указанной работе между каждой парой узлов в точке экстремума параболы добавляется условие неотрицательности. Примечательно, что процедура не является многошаговой. Позиция экстремума выражается через ещё неизвестные параметры полинома (представленного в виде $a_k x^3 + b_k x^2 + c_k x + d_k$). В результате получается набор дополнительных нелинейных по параметрам ограничений вида $f'(-b_k/3a_k) \geq 0$, причём при условии $a_k > 0$ и $-b_k/3a_k \in [s_k, s_{k+1}]$. В указанной работе решение предлагается искать с помощью численной оптимизации. Однако подробности реализации этой численной оптимизации не поясняются. Из работы неясно, как осуществляется проверка дополнительных условий $a_k > 0$ и $-b_k/3a_k \in [s_k, s_{k+1}]$, если сами эти параметры подбираются в результате численной оптимизации. Из-за активации/деактивации этих условий во время самой оптимизации есть сомнения, что задача остаётся выпуклой и что оптимизация не застревает в локальном оптимуме. С другой стороны, численная оптимизация сама является итерационной. Если на итерациях происходит движение точки экстремума, в которой добавляется условие

неотрицательности, то это по сути эквивалентно, как если бы сама процедура была многошаговой, с проверкой и добавлением новых условий неотрицательности, где нужно.

В данной работе реализуется многошаговая процедура добавления дополнительных условий $g(s_k + \Delta t) \geq 0$ без добавления новых узлов. Расчёт значения $g(s_k + \Delta t)$ согласно (2.30) можно записать в матричном виде

$$g(s_k + \Delta t) = V_{cond}g - P_{cond}\gamma = (V_{cond} - P_{cond}R^{-1}Q^T)g, \quad (2.59)$$

где V_{cond} – строка размерностью $1 \times m$, имеющая вид

$$\begin{array}{cccccccccc} \hline 1 & 2 & \dots & k-1 & k & k+1 & k+2 & \dots & m \\ \hline 0 & 0 & \dots & 0 & \frac{h_k - \Delta t}{h_k} & \frac{\Delta t}{h_k} & 0 & \dots & 0 \\ \hline \end{array}$$

P_{cond} – строка размерностью $1 \times (m - 2)$, имеющая вид

$$\begin{array}{cccccccccc} \hline 2 & \dots & k-1 & k & k+1 & k+2 & \dots & m-1 \\ \hline 0 & \dots & 0 & \frac{\Delta t(h_k - \Delta t)(h_k + \Delta t)}{6h_k} & \frac{\Delta t(h_k - \Delta t)(2h_k - \Delta t)}{6h_k} & 0 & \dots & 0 \\ \hline \end{array}$$

Выражение $V_{cond} - P_{cond}R^{-1}Q^T$ является строкой размерностью $1 \times m$. Если имеется несколько пар узлов s_k, s_{k+1} , между которыми необходимо добавить дополнительное условие неотрицательности, то таких строчек будет несколько. Из этих строк соберём матрицу Z (пусть количество строк / дополнительных условий будет r) размерностью $r \times m$. Тогда оптимизационная задача с условиями неотрицательности в узлах и промежуточных точках запишется как

$$\begin{aligned} S(g) &= c^T g + \frac{1}{2} g^T D g \rightarrow \min, \\ -g &\leq 0, \\ -Zg &\leq 0. \end{aligned}$$

Все условия можно объединить, используя блочную матрицу $M = \begin{bmatrix} -I \\ -Z \end{bmatrix}$ размерностью $(m + r) \times m$, где I – единичная матрица размерностью $m \times m$. На практике это даже проще. Каждый раз, когда для пары узлов s_k, s_{k+1} нужно добавить дополнительное условие, к отрицательной единичной матрице дописывается строчка коэффициентов из (2.59).

В итоге получаем оптимизационную задачу

$$S(g) = c^T g + \frac{1}{2} g^T D g \rightarrow \min , \quad (2.60)$$

$$Mg \leq 0$$

которую решаем одним из методов квадратичного программирования.

Преимущество такого способа в том, что не надо пересчитывать векторы и матрицы, не создаётся дополнительных степеней свободы (которые появляются при добавлении новых узлов). Заметим, что все ограничения остаются линейными по параметрам и можно использовать аналитические методы решения задачи квадратичного программирования.

Монотонный сплайн (интегральная функция $F(t)$)

Если нас интересует интегральная функция $F(t) = \int_{-\infty}^t f(t)dt$, то достаточно адаптировать (2.33) к произвольной точке t (подставив $k = 1$ и $t_u^a = s_1$, нумеруя L от 1)

$$F(t) = \int_{s_1}^t g(t)dt = \sum_{l=1}^L \left[\frac{h_l}{2} g_{l+1} + \frac{h_l}{2} g_l - \frac{h_l^3}{24} \gamma_{l+1} - \frac{h_l^3}{24} \gamma_l \right]$$

$$- \frac{h_L^2 - (h_L^{-t})^2}{2h_L} g_{L+1} - \frac{(h_L^{+t})^2}{2h_L} g_L \quad (2.61)$$

$$+ \frac{(h_L^{+t})^2 (h_L^{-t} + h_L)^2}{24h_L} \gamma_{L+1} - \frac{(h_L^{+t})^2 ((h_L^{+t})^2 - 2h_L^2)}{24h_L} \gamma_L,$$

$$L: s_L < t \leq s_{L+1},$$

где L – номер интервала, содержащий t , $h_L = s_{L+1} - s_L$, $h_L^{-t} = t - s_L$, $h_L^{+t} = s_{L+1} - t$. Данное выражение будет совпадать с выражением в [209, с. 11], если из суммы по l вынести последний элемент суммы и упростить выражение. Выражение (2.61) является полиномом четвёртой степени.

Примеры

Программная реализация описанного способа построения монотонного/положительного сплайна реализована на языке Python в виде функции `FunctionalSmoothingSpline` и доступна на GitHub по адресу: <https://github.com/YuAKorablev/FunctionalSmoothingSpline>.

1) Восстановление функции только по её значениям. В случае линеаризации ($\log y_i = g(t_i)$) можно использовать только данные значений

самой функции (без данных производных и интегралов). Начнём с примера, когда доступны только зашумлённые данные самой функции, таблица 2.2. Пусть нас интересует проблема сглаживания (аппроксимации), а не интерполяции, поэтому выберем достаточно большой коэффициент сглаживания $\alpha = 10^2$. Количество узлов возьмём в три раза больше, чем наблюдений, $m = 21$. Без условия неотрицательности результат изображён на рисунке 2.10(а). Заметно, что функция принимает отрицательные значения на некотором интервале. Применение линеаризации ($\log y_i$) изображено на рисунке 2.10(б). Заметим, что в этом случае использован коэффициент сглаживания $\alpha = 10^2/2.3$ (так как $\log(200)$ меньше 200 приблизительно в 2.3 раза). На рисунке 2.10(в) изображён нелинеаризованный случай использования экспоненты (когда используется численная оптимизация). Рисунок 2.10(г) показывает применение подхода, основанного на квадратичном программировании.

Таблица 2.2 – Данные для восстановления функции только по её значениям

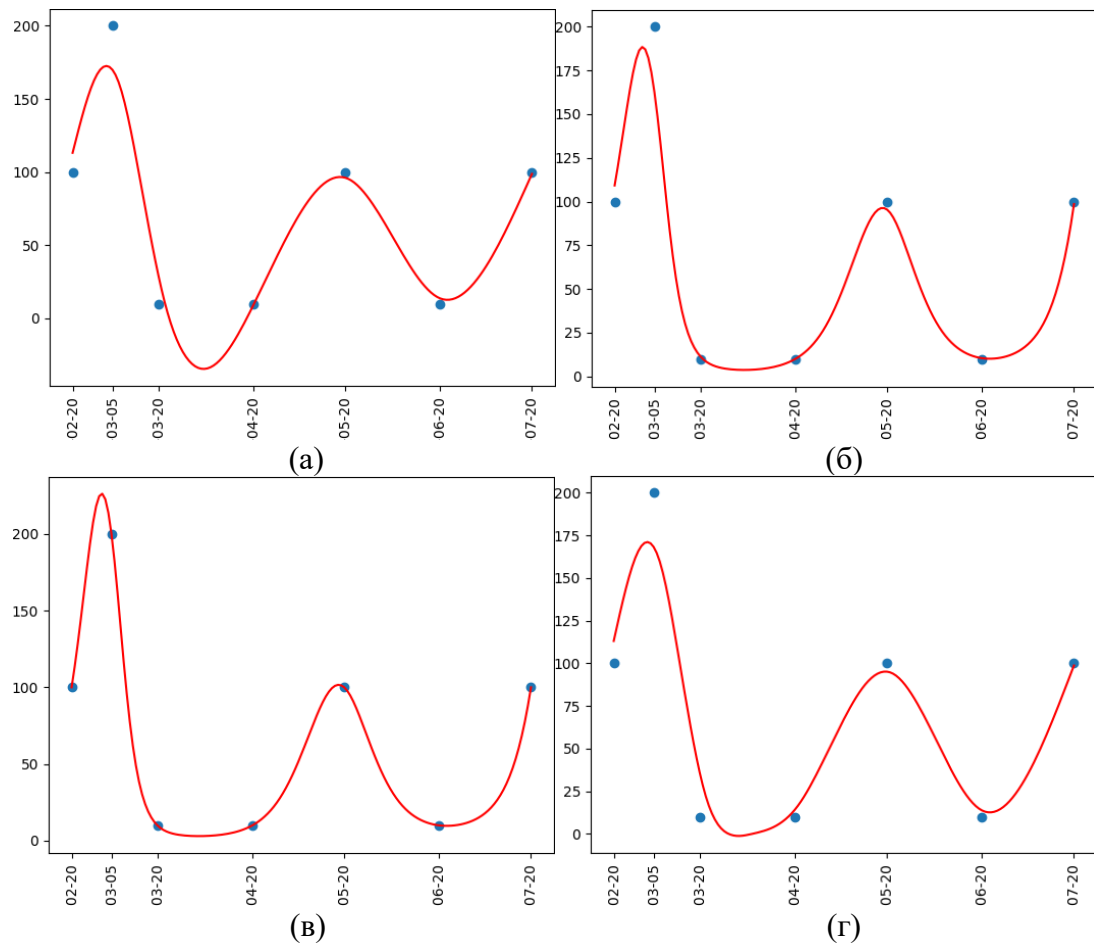
№	t_f	y_f	№	t_f	y_f
1	20.02.2021	100	5	20.05.2021	100
2	05.03.2021	200	6	20.06.2021	10
3	20.03.2021	10	7	20.07.2021	100
4	20.04.2021	10	-	-	-

Источник: составлено автором.

Можно заметить, что в случае использования экспоненты от сплайна (линеаризованный и нелинеаризованный случай) пики получаются острее. Подход на основе квадратичного программирования даёт результат, наиболее близкий к тому, который получается без добавления условий неотрицательности.

Замечание. Чтобы сказать, какой метод восстановления лучше, нам необходимо измерить, насколько отклонилась восстановленная функция от исходной. Здесь у нас есть только данные, самой исходной функции нет. Даже если бы исходная функция была, из которой отобрано лишь небольшое количество наблюдений, отклонение восстановленной функции от исходной

может быть следствием неудачно выбранных точек наблюдения, а не выбранного способа восстановления.



Источник: составлено автором.

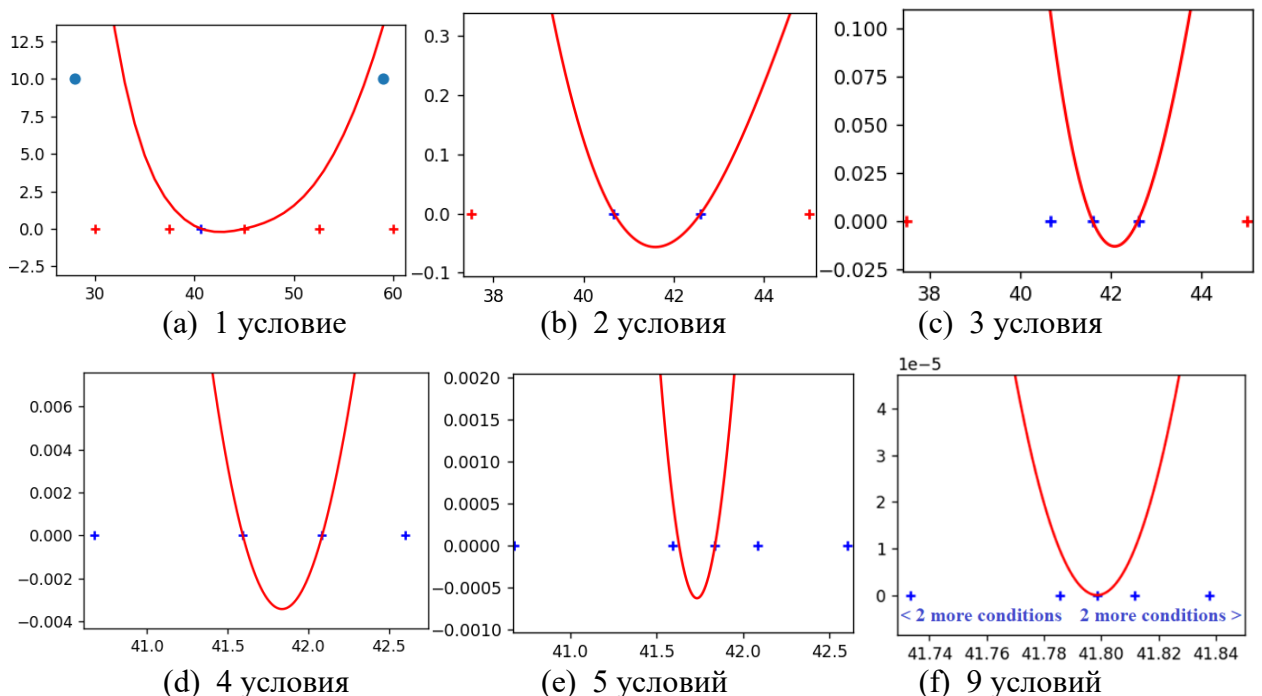
Рисунок 2.10 – Восстановление функции по значениям: а) без условий неотрицательности, б) используя логарифмы значений, в) нелинеаризованный случай с экспонентой, г) подход на основе квадратичного программирования

Однако в подходе, основанном на сведении к задаче квадратичного программирования, восстановленная функция между узлами принимает хоть незначительные, но всё же отрицательные значения. Рассмотрим два способа обеспечения положительности.

1) Добавление нового узла в точке минимума. Добавление нового узла в отрицательной точке минимума и повторный пересчёт параметров сплайна значительно уменьшает вылет функции ниже нуля, но не полностью. Между новым и старым узлом функция по-прежнему может опуститься ниже нуля. Даже после добавления 9-го узла функция опускается ниже нуля. Заметим, что при продолжении добавления новых узлов расстояние между узлами может

стать настолько малым, что алгоритм расчёта параметров сплайна может привести к возникновению ошибки вычисления, например деления на ноль (в этом примере, если не ставить ограничение по порогу, алгоритм доходит до добавления 27-го узла, после чего ломается из-за ошибки вычислений). Если использовать этот способ, то следует остановиться на добавлении не очень большого количества узлов или при достижении заданного пользователем порога, при котором выход за ноль на меньшую величину предполагается несущественным. После этого близкие к нулю отрицательные величины можно заменить на нулевые значения обычным присвоением.

2) Добавление условий неотрицательности без добавления новых узлов (рекомендуемый способ). В данном способе не создаются новые степени свободы из-за новых узлов сплайна. В этом примере при добавлении 9-го условия алгоритм успешно завершается, так как точка экстремума оказывается в самой точке нового условия неотрицательности. На рисунке 2.11 показан ход добавления новых условий без добавления узлов.



Источник: составлено автором.

Рисунок 2.11 – Добавление условий неотрицательности без добавления узлов

Заметим, что здесь результат получен при использовании точного метода Лемке для решения задачи квадратичного программирования. Если же

использовать численный метод (свхорт [155]), который основан на методе внутренней точки, то алгоритм заканчивает работать ещё раньше для обоих способов (с добавлением узлов или условий). Связано это, скорее всего, с тем, что в численном методе по умолчанию заложена не очень высокая абсолютная и относительная точность (10^{-7} и 10^{-6} соответственно) достижения минимума целевой функции. После добавления 6-го узла сплайн становится положительным между всеми узлами в обоих способах.

Однако доказательств, что количество итераций будет конечным для других входных данных, ещё нет. Можно ограничивать количество итераций добавления новых узлов или условий, вводить порог обеспечения неотрицательности. Если всё же нужна абсолютная гарантия обеспечения неотрицательности, то можно вернуться к подходу, когда сплайн помещается в степень экспоненты, однако он заметно видоизменяет восстанавливаемую функцию.

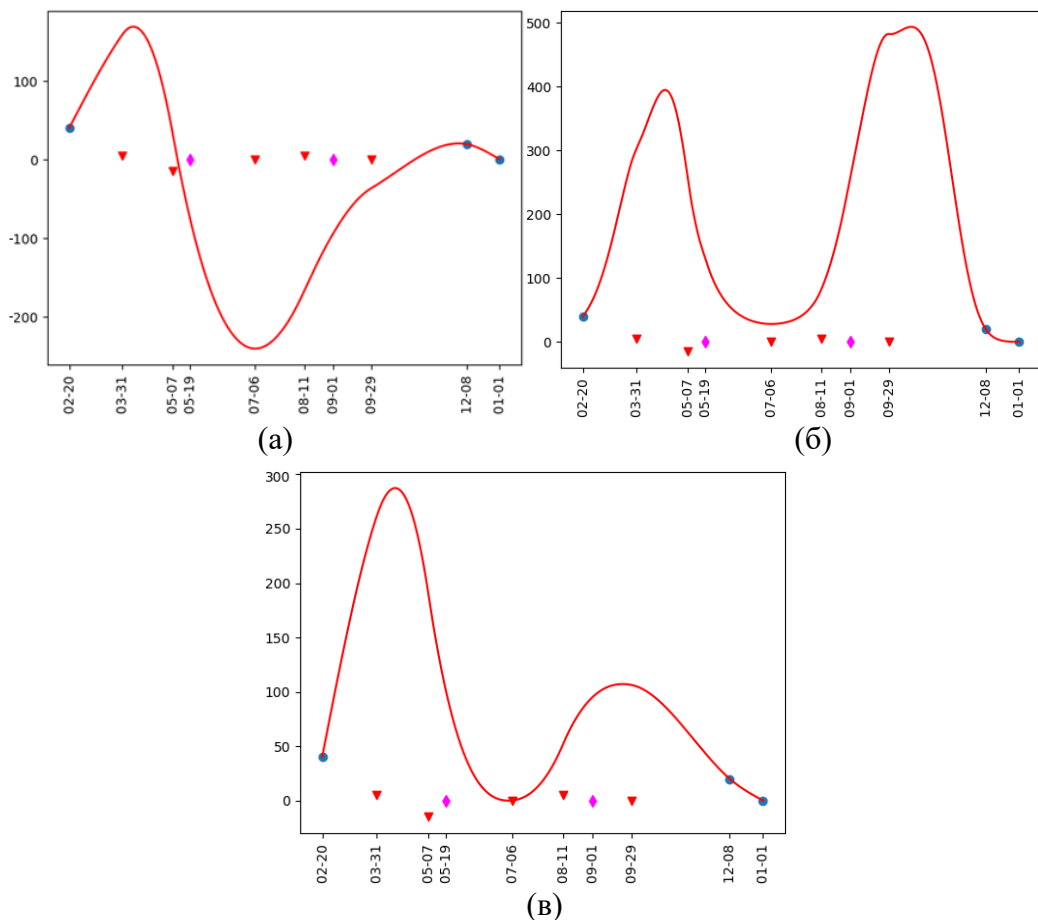
2) Восстановление функции по значениям, значениям первой и второй производной. Теперь восстановим функцию не только по зашумлённым значениям самой функции, а также по значениям первой и второй производной, данные представлены в таблице 2.3. Линеаризация уже неприменима. В этом примере используется одинаковый коэффициент сглаживания $\alpha = 10^2$. Количество узлов $m = 30$. Также уделяется большое внимание значениям первой производной (вес группы $\mu = 10$) и второй производной (вес группы $\nu = 20$). В этом примере всего 3 точки наблюдений самих значений функции, причём расположенных по краям. В середине функция определена только значениями и значениями второй производной. Без условия неотрицательности, рисунок 2.12(а), функция принимает большие отрицательные значения в середине, что может противоречить физическому смыслу. Нелинеаризованный случай взятия экспоненты от сплайна, рисунок 2.12(б), приводит к очень высоким пикам (причём повышение коэффициента сглаживания до $\alpha = 10^3$ лишь увеличивает высоту этих пиков до 500 и 600). Восстановление функции с помощью подхода на основе

квадратичного программирования даёт визуально лучше результат, но это субъективно.

Таблица 2.3 – Данные для восстановления функции по значениям, значениям первой и второй производной

t_f	y_f	t_{df}	y_{df}	t_{d^2f}	y_{d^2f}
20.02.2021	40	31.03.2021	5	19.05.2021	0
08.12.2021	20	07.05.2021	-15	01.09.2021	0
01.01.2022	0	06.07.2021	0	-	-
-	-	11.08.2021	5	-	-
-	-	29.09.2021	0	-	-

Источник: составлено автором.



Синими точками обозначены наблюдаемые значения функции. Красными треугольниками обозначены значения первой производной. Пурпурными ромбиками обозначены значения второй производной. Сплошная линия – восстановленная функция.

Источник: составлено автором.

Рисунок 2.12 – Восстановление функции: а) без условий неотрицательности, б) нелинеаризованный случай экспоненты, в) подход на основе квадратичного программирования.

3) Восстановление функции по значениям, значениям первой и второй производной и по значениям интегралов.

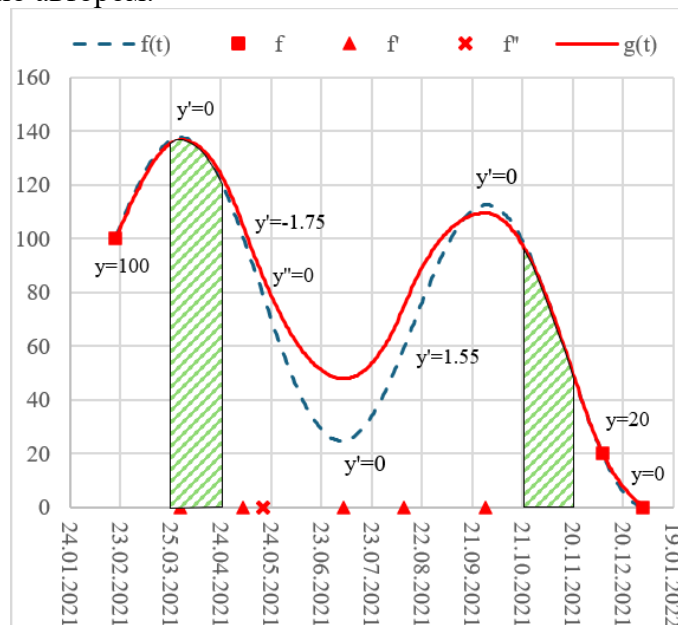
В данном примере у нас есть значения, значения первой и второй производной в определённых точках, значения определённых интегралов

функции $f(t)$ на некоторых интервалах, таблица 2.4. Так как задействуется информация об интегралах, то подход на основе экспоненты от сплайна не подходит. В этом примере исходная функция $f(t)$ была положительная, и данные позволяют восстановить функцию без условий неотрицательности. Если добавить условия неотрицательности и воспользоваться подходом на основе квадратичного программирования, то восстанавливаемая функция совпадает с функцией, восстановленной без условий неотрицательности (разница не превышает 0.03 в абсолютном значении, незаметно на рисунке). Восстановленная функция в виде сплайна $g(t)$ изображена на рисунке 2.13. Наличие исходной функции $f(t)$ позволяет нам визуальное сравнение качества восстановления. Однако ошибки восстановления связаны с недостатком данных, а не с методом.

Таблица 2.4 – Данные разных функционалов

t_f	y_f	t_{df}	y_{df}	t_{d^2f}	y_{d^2f}	t_{int}^a	t_{int}^b	Y_{int}
20.02.2021	100	31.03.2021	0	19.05.2021	0	25.03.2021	24.04.2021	4000
08.12.2021	20	07.05.2021	-1,75	-	-	21.10.2021	20.11.2021	2282
01.01.2022	0	06.07.2021	0	-	-	-	-	-
-	-	11.08.2021	1,55	-	-	-	-	-
-	-	29.09.2021	0	-	-	-	-	-

Источник: составлено автором.



Треугольниками показаны позиции наблюдений первой производной на оси X. Крестиком обозначена позиция единственного наблюдения второй производной (точки перегиба). Наблюдения определённых интегралов изображены заштрихованными областями. Квадратиками показаны наблюдения самих значений функции.

Источник: составлено автором.

Рисунок 2.13 – Восстановление функции по разным функционалам

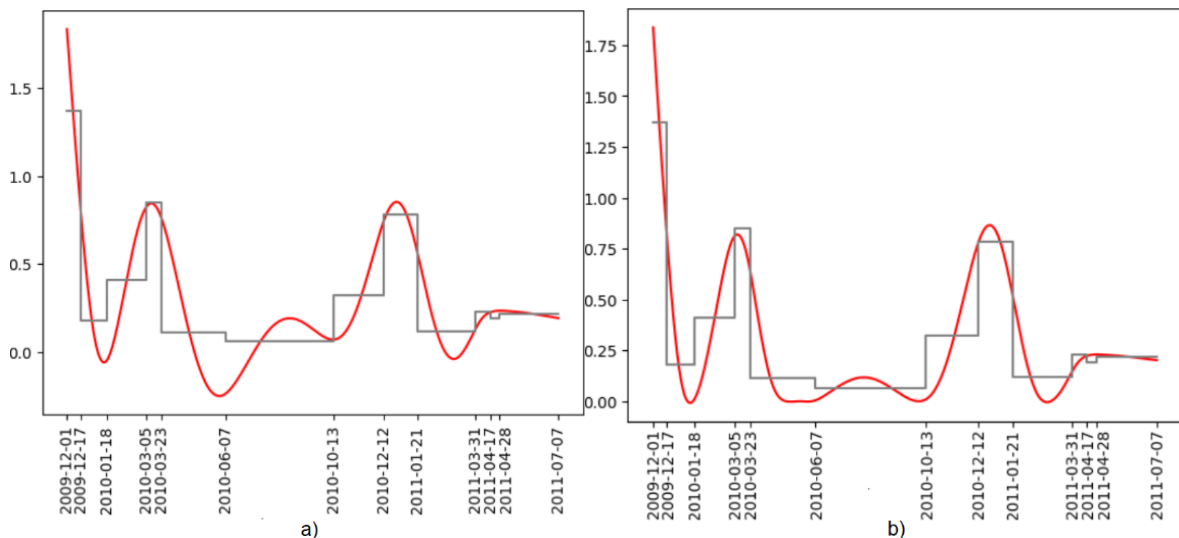
4) Практический пример монотонной сплайновой коллокации.

Имеются данные онлайн-покупок (t_i, Y_i) , где Y_i – общая сумма покупок клиента в указанную дату t_i , выраженная в долларах, таблица 2.5. По этим данным можно восстановить скорость увеличения желания совершить очередную покупку или скорость накопления средств, которые клиент подготавливает для шопинга (предполагая $Y_i = \int_{t_i}^{t_{i+1}} f(t)dt + \varepsilon_i$). Подробно пример с онлайн-магазином будет рассмотрен в главе 5. На рисунке 2.14(a) изображено восстановление функции без условий неотрицательности, а на рисунке 2.14(b) с условием неотрицательности. Интегральные функции $F(t)$ с условием и без изображены на рисунке 2.15(a) и (b).

Таблица 2.5 – Наблюдаемые с погрешностью значения определённых интегралов

t_i	Y_i	t_i	Y_i	t_i	Y_i
01.12.2009	21,95	07.06.2010	7,95	17.04.2011	2,1
17.12.2009	5,7	13.10.2010	19,3	28.04.2011	15,27
18.01.2010	18,75	12.12.2010	31,3	07.07.2011	9,2
05.03.2010	15,25	21.01.2011	8,25	-	-
23.03.2010	8,5	31.03.2011	3,9	-	-

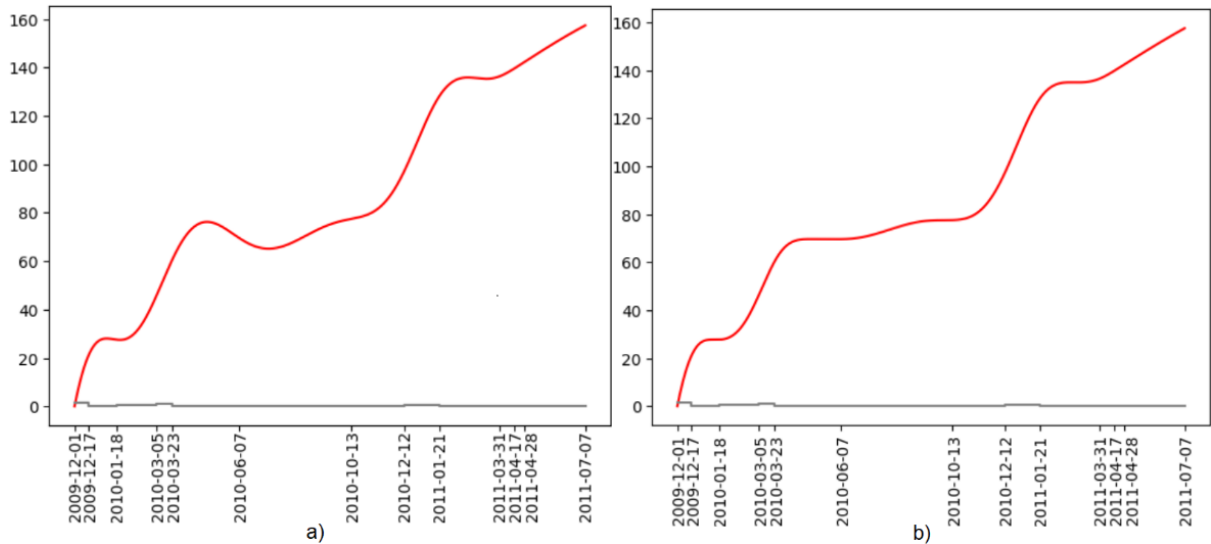
Источник: составлено автором.



Ступенчатая линия показывает среднее значение $Y_i/(t_{i+1} - t_i)$, площадь под каждой ступенькой является интегралом Y_i .

Источник: составлено автором.

Рисунок 2.14 – Восстановление функции: а) без условий неотрицательности, б) с условиями неотрицательности.



Источник: составлено автором.

Рисунок 2.15 – Интегральная функция $F(t)$: а) без условий неотрицательности, б) с условиями неотрицательности (монотонный сплайн)

5) *Дополнительный практический пример монотонной сплайновой коллокации.*

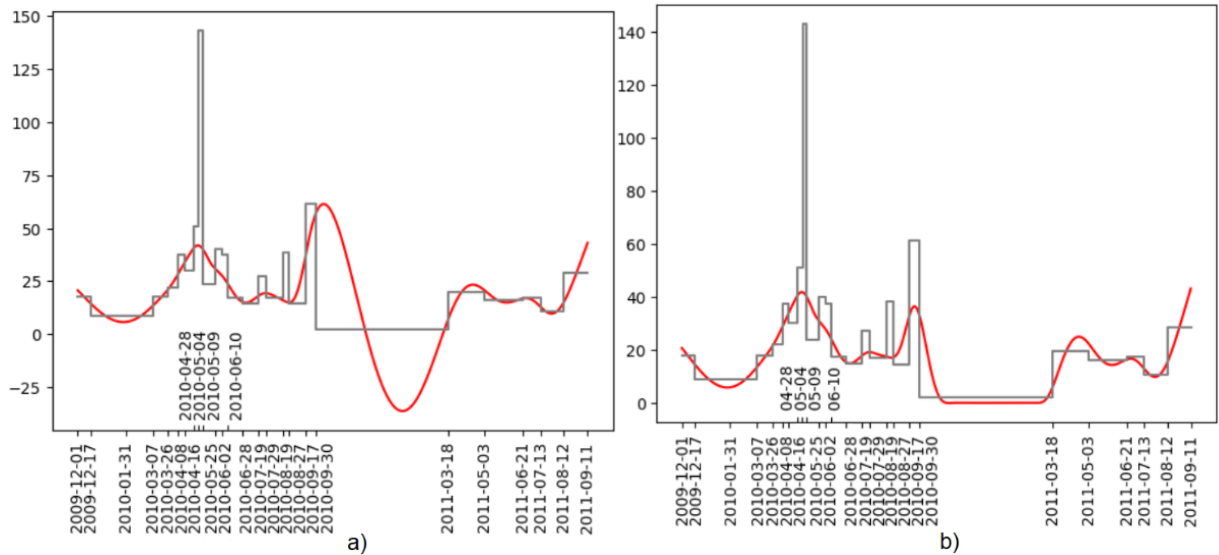
Имеются данные (t_i, Y_i) другого клиента о сумме покупок в онлайн-магазине, таблица 2.6. Аналогично на рисунке 2.16 показано восстановление функции по последовательности интегралов, а на рисунке 2.17 показана интегральная функция $F(t)$.

Замечание. Интегральная функция $F(t)$ в самый последний момент t_n не равна накопленной сумме интегралов к этому моменту $F(t_n) \neq \sum_{i=1}^{n-1} Y_n$, так как штраф на меру кривизны $\int_{S_1}^{S_m} (g''(t))^2 dt$ «перетягивает на себя» часть невязки наблюдений, заставляя функцию именно сглаживать данные наблюдений. Если нужно обеспечить равенство, то следует отключить этот штраф, положив коэффициент сглаживания $\alpha = 0$. Однако зачастую без регуляризации задача может не иметь решений. Например, для описанных выше примеров восстановления функции по последовательности интегралов, в случае если $\alpha = 0$, то матрица D , от которой берётся обратная в выражении (2.54) или которая используется в алгоритме Лемке, оказывается вычислительно сингулярной, и решение может либо не находиться, либо находиться неправильно.

Таблица 2.6 – Другие данные наблюдаемых с погрешностью интегралов

t_i	Y_i	t_i	Y_i	t_i	Y_i	t_i	Y_i
01.12.2009	285,06	28.04.2010	305,2	19.07.2010	274	03.05.2011	797,26
17.12.2009	391,44	04.05.2010	715,03	29.07.2010	361,16	21.06.2011	379,72
31.01.2010	304,47	09.05.2010	380,7	19.08.2010	307,46	13.07.2011	317,4
07.03.2010	337,74	25.05.2010	321,02	27.08.2010	303	12.08.2011	860,98
26.03.2010	286,58	02.06.2010	300,25	17.09.2010	797,14	11.09.2011	905,99
08.04.2010	297,94	10.06.2010	311,26	30.09.2010	354,03	-	-
16.04.2010	363,12	28.06.2010	308,1	18.03.2011	904,26	-	-

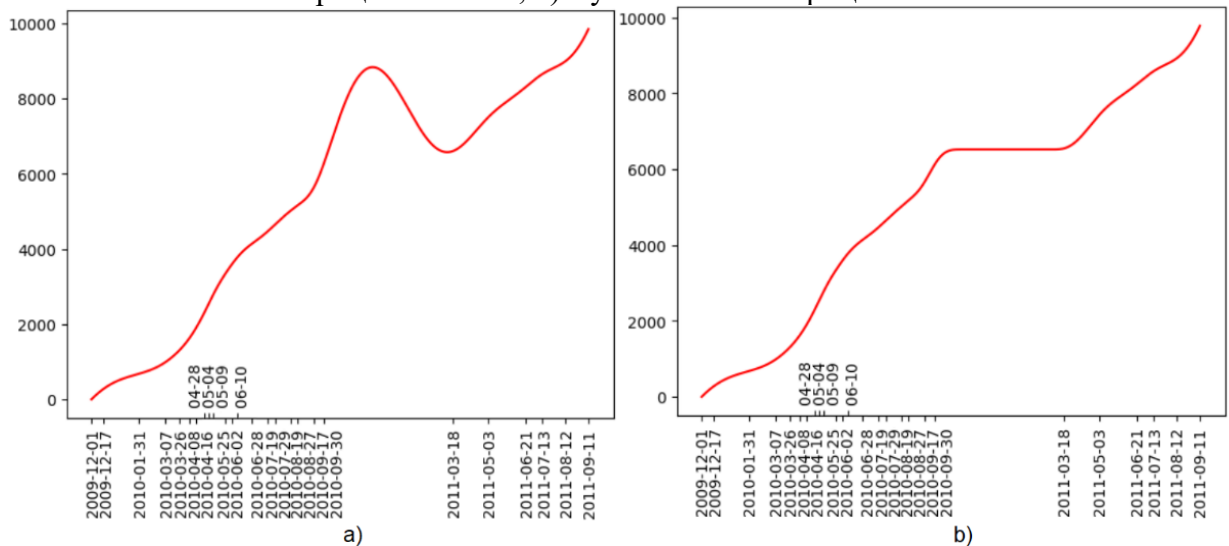
Источник: составлено автором.



Ступенчатая линия показывает среднее значение $Y_i/(t_{i+1} - t_i)$, площадь под каждой ступенькой является интегралом Y_i .

Источник: составлено автором.

Рисунок 2.16 – Восстановленная по интегралам функция: а) без условий неотрицательности, б) с условиями неотрицательности



Источник: составлено автором.

Рисунок 2.17 – Интегральная функция $F(t)$: а) без условий неотрицательности, б) с условиями неотрицательности (монотонный сплайн)

2.7 Экстраполяция параметров и получение прогноза будущего события

После того, как по выборке событий были восстановлены параметры модели механизма образования событий, прежде чем сделать прогноз будущего события, необходимо провести экстраполяцию параметров механизма на будущее.

Со стационарными параметрами, такими как в нашем примере механизма образования событий параметры P_2 и P_3 , имеющими смысл значения критического и максимального запаса, всё просто. Раз они стационарные, не меняющиеся со временем, то в будущем они будут принимать точно такие же значения, как и в настоящем (такими, какими они определились по выборке событий).

С нестационарными параметрами намного сложнее. Нестационарные параметры меняются со временем, нужно определить, какие значения они будут принимать в будущем. Конечно же, будущее неясно, и никто наверняка не скажет, какие изменения могут произойти. Но если в событиях прошлого и настоящего заложены какие-то закономерности, которые можно предположить и распространить на будущее, то следует отыскать эти закономерности.

Экстраполяция параметров, восстановленных регрессионными методами и методами машинного обучения

Что в регрессионных методах, что в методах машинного обучения – оцениваются параметры или обучается модель зависимости $f(t)$ или $f(X)$. В примере на основе регрессионных методов была полиномиальная модель $P_1(t) = a_0 + a_1t + a_2t^2 + \dots + a_mt^m$. В градиентном бустинге получается модель $f(X) = \sum_{j=1}^m \rho_j h_j(X)$, где $h_j(X)$ – модель, построенная на псевдоостатках на шаге j , в основном в виде дерева решений, ρ_j – множитель, с которым учитывается результат модели шага j , X – вектор входных признаков. В нейросетях моделью является сама обученная нейросеть,

выдающая на выходном слое вещественное значение. Если есть доверие к построенным моделям, то для экстраполяции значения параметров механизма образования событий достаточно подавать на вход обученных моделей новые значения t или сформированный вектор признаков X , соответствующий будущему времени.

Однако в рассмотренных выше примерах применения регрессионных методов и методов машинного обучения плохо получилось восстановить динамичный параметр $P_1(t)$. Особенно на краях. Поэтому, скорее всего, потребуется дополнительный поиск закономерности, как это делается для сплайновой коллокации ниже.

Экстраполяция параметров, восстановленных методами сплайновой коллокации

В отличие от регрессионных методов сглаживающие сплайны относят к непараметрическим методам, в том смысле, что не предполагается строгая форма модели. Более того, на концах исследуемого интервала сплайн (по условиям натурального сплайна) обязательно обращается в прямую линию (вторая производная равна нулю). Поэтому просто рассчитывать значение сплайна для будущих моментов времени будет ошибочно. Сплайн используется для первичного анализа. Для экстраполяции нужно выявить некоторые закономерности в изменениях со временем у этого восстановленного по событиям сплайна.

Один из подходов основан на привязывании определённых тенденций к временам года, праздничным датам и т.д. Например, в работах автора [196; 215, с. 38–42] анализировались поставки кваса в универсам, было видно, что потребление увеличивается в праздники и в жаркую погоду летом. Поэтому тенденции прошлого года переносились на будущий год, чтобы в определённые месяцы потребление кваса было соответствующим образом меньше или больше.

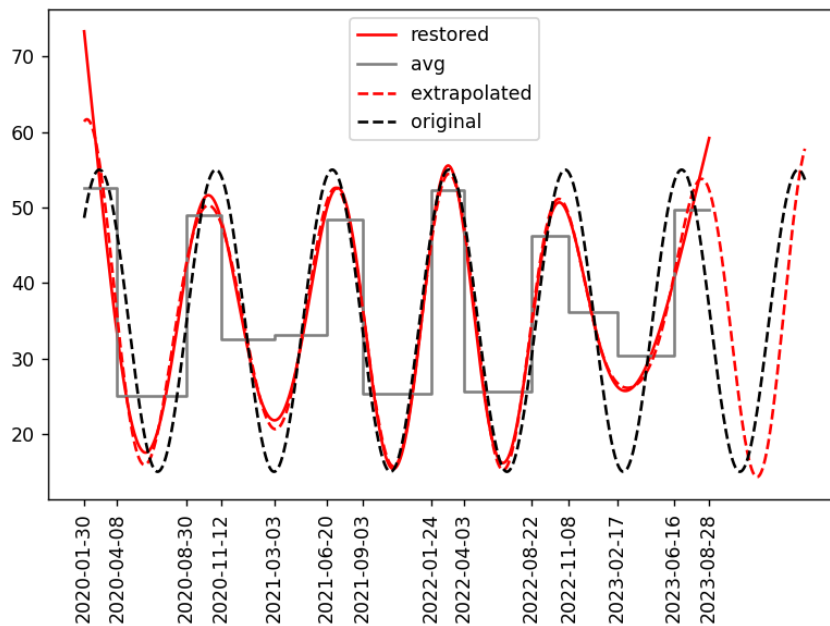
Другой подход основан именно на выявлении закономерностей в полученной восстановленной функции. Можно предполагать, что эти изменения цикличны. В работах [201; 215, с. 42–50], так как восстановленная функция не была привязана к временам года или праздничным датам и в ней наблюдались заметные циклические колебания, эта функция раскладывалась на сумму ограниченного числа гармонических функций. Разложение на гармоники может рассматриваться как один из способов экстраполяции.

Однако обычное разложение в ряд Фурье не подходит для экстраполяции, так как в нём весь рассматриваемый интервал времени предполагается периодом всей функции, и при экстраполяции это приводит к тому, что значение на конце интервала скачком становится равным значению в начале интервала (подробно об этом написано в статье [157]). Для экстраполяции хорошо подходит алгоритм Куинна и Фернандеса (Quinn-Fernandes algorithm) [158; 159]. Доходчивое объяснение можно посмотреть в источнике [160].

Алгоритм Куинна и Фернандеса эквивалентен следующему алгоритму [159, с. 53]. Из исходного ряда данных вычитается его среднее значение. Затем по нему строится периодограмма и отбирается одна самая сильная гармоника (её частота). Далее определяются амплитуды синуса и косинуса для отобранной частоты регрессионными методами по ряду данных без среднего значения. Из имеющегося ряда вычитается полученная гармоника, и для полученного ряда вновь строится периодограмма. Опять отбирается самая сильная гармоника. Определяются (корректируются) амплитуды синуса и косинуса уже для двух отобранных частот регрессионными методами и т.д. Этот алгоритм чем-то напоминает идею градиентного бустинга, когда по псевдоостаткам строятся дополнительные модели. Однако в алгоритме Куинна и Фернандеса не строится периодограмма, а используется хитрое, но простое итерационное вычисление. Считается, что алгоритм Куинна и Фернандеса из-за этого работает быстрее. В примерах ниже используется

программная реализация, взятая с MetaTrader 5 trading platform [161], адаптированная автором на язык R, а потом и на язык Python.

В итоге восстановленную функцию можно разложить на ограниченное число гармонических функций. Такое разложение в нашем примере и будет являться моделью для экстраполяции значений параметра $P_1(t)$ на будущее. Пример экстраполяции на 200 дней вперёд с помощью разложения на 10 гармоник изображен на рисунке 2.18.



Источник: составлено автором.

Рисунок 2.18 – Экстраполяция разложением на 10 гармоник

В данном примере количество гармоник могло бы равняться и одному, но заранее это знать нельзя. Количество гармоник должно быть достаточным, чтобы объяснять поведение наблюдаемой восстанавливаемой функции. В работе [159] есть рекомендации по вычислению нужного количества гармоник, но это выходит за рамки данного исследования. Десяти гармоник часто достаточно для наших целей.

Стоит обратить внимание, что любой сглаживающий сплайн на концах превращается в прямые линии. Эти прямые участки могут повлиять на процесс экстраполяции. Желательно отбросить некоторое количество наблюдений с каждой из сторон. То есть, прежде чем раскладывать функцию на гармоники,

отбрасывается n_1 точек функции (дней) слева и n_2 справа. Количество отбрасываемых точек n_1 и n_2 становится дополнительными гиперпараметрами, помимо параметра сглаживания α . Способ определения гиперпараметров будет описан немного ниже.

Это один из способов экстраполяции, который выбран для того, чтобы можно было как-то автоматизировать данный этап. Если же у исследователя имеется какая-то информация о возможных тенденциях в будущем, то экстраполяцию можно проводить, полагаясь на экспертное мнение этого исследователя. Вся ответственность за выбор метода экстраполяции параметров механизма образования событий лежит на исследователе.

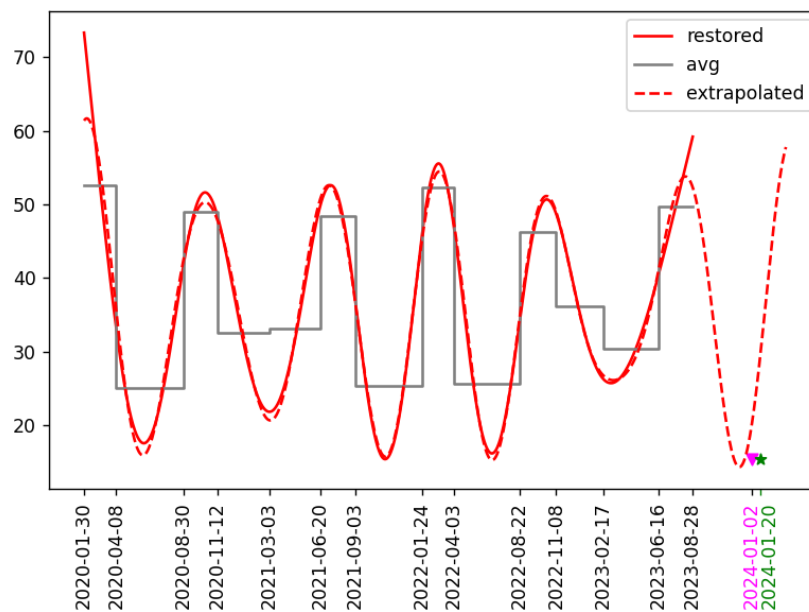
Получение прогноза будущего события

Когда все параметры механизма образования событий восстановлены и осуществлена экстраполяция значений этих параметров на будущее, то получение прогноза будущих событий происходит довольно просто. Достаточно всего лишь осуществить моделирование самого механизма образования событий с установленными значениями параметров.

У модели механизма есть внутренние переменные X , которые надо проинициализировать. Установить значения внутренних переменных такими, какими они могли бы быть в момент появления последнего события обучающей выборки. На последнем событии механизм как бы «замораживается», а при прогнозе события «размораживается». В используемом нами сквозном примере с моделью механизма, как в системе управления запасами, имеется лишь одна внутренняя переменная X , означающая текущие запасы. Согласно нашей модели механизма, в момент создания события запасы пополняются до максимального уровня, $X = P_3$, где параметр определяли по выборке событий $P_3 = n^{-1} \sum_{i=1}^n (y_i - 0.5P_1(t_i))$.

Примечание – Параметр P_3 можно определить на основе только последнего наблюдения $P_3 = y_n - 0.5P_1(t_n)$. Небольшая часть объёма последней покупки y_n откладывается в страховые запасы, которые были использованы в момент образования события t_n . Среднее значение (по случайным реализациям) использованных страховых запасов будет половина спроса в соответствующий день. Разница этих двух способов невелика, так как $P_1(t_n)$ – это спрос лишь за один день.

Установив значение внутренних переменных, осуществляем моделирование заданного механизма образования событий и получаем прогноз будущего события. В нашем сквозном примере прогноз момента времени будущего события \hat{t}_{n+1} выпадает на дату 02.01.2024 (спустя 127 дней от последнего события). Фактическое будущее событие t_{n+1} в этом примере должно было быть 20.01.2024, т.е. на 18 дней позже. Погрешность определения даты события составила 12% от интервала времени между фактическими событиями (145 дней). Прогноз и фактическое событие изображены на рисунке 2.19.



Пурпурным треугольником обозначен прогноз будущего события, зелёной звёздочкой обозначено фактическое событие.

Источник: составлено автором.

Рисунок 2.19 – Прогноз будущего события и фактическое событие

Помимо даты события, прогнозируется ещё и значение события \hat{y}_{n+1} . В примере оно определилось как 3613, когда фактическое значение y_{n+1} было 3633 (разница меньше процента, но это простейший игрушечный пример с постоянным максимумом). В главе 5 будут более сложные примеры на реальных данных.

2.8 Определение гиперпараметров

В рассматриваемом сквозном примере имеется три гиперпараметра: α – коэффициент сглаживания, n_1 и n_2 – количество отбрасываемых с начала и с конца точек сплайна перед разложением на ограниченное количество гармоник.

Для определения этих гиперпараметров разработана отдельная процедура. Идея этой процедуры мотивирована тем, как ищутся гиперпараметры при использовании методов машинного обучения с помощью алгоритмов поиска на сетке. Для поиска гиперпараметров используется глобальная оптимизация с помощью поиска на сетке и локальная оптимизация с помощью алгоритма Нелдера-Мида [162] в окрестности узлов сетки. В методах машинного обучения в качестве критерия для поиска гиперпараметров в основном используется метрика точности классификации или суммарная квадратичная ошибка. Для задачи прогнозирования событий нужна другая метрика. Такой метрикой, например, будет ошибка прогноза самого события.

Прогнозируемое событие в нашем примере характеризуется двумя показателями – датой прогноза \hat{t}_{n+1} и характеристикой события \hat{y}_{n+1} . Оценку ошибки прогноза будущего события определяем как

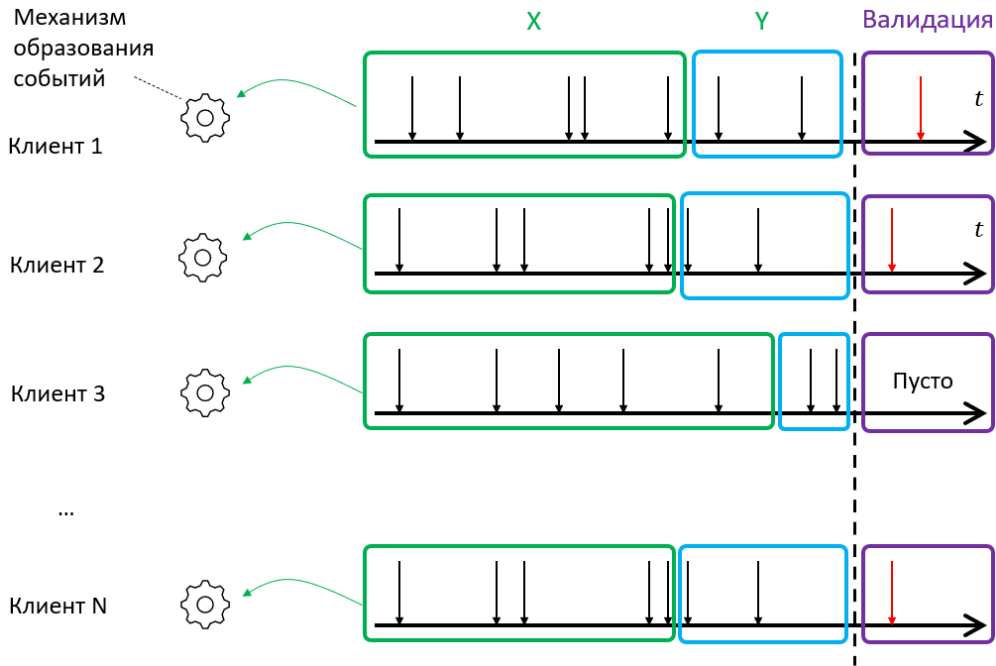
$$Score = \left| \frac{\hat{t}_{n+1} - t_{n+1}}{t_{n+1} - t_n} \right| + \mu \left| \frac{\hat{y}_{n+1} - y_{n+1}}{y_{n+1}} \right|, \quad (2.62)$$

где t_{n+1} и y_{n+1} (без крышки) – фактические значения будущего события, разница $t_{n+1} - t_n$ показывает фактическую ширину интервала между событиями. Тем самым качество прогноза выражается через относительную погрешность времени до события и соответствующего ему возвращаемого значения (объём покупки в сквозном примере). Вес μ выбирается эмпирически, например 0.1, из соображений, что в первую очередь интерес составляет дата события. Кроме суммы модулей можно использовать корень из суммы квадратов отклонений, относительных или абсолютных.

Для определения гиперпараметров в качестве фактического события (t_{n+1}, y_{n+1}) должно выступать не то событие, по которому происходит валидация всей модели. Этим событием должно быть последнее событие из обучающей выборки. «Окно» с данными нужно корректно сдвинуть на одну позицию влево. Однако выяснилось, что подгонка гиперпараметров лишь к одному событию плохо сказывается на способности предсказывать очередное событие. Следует определять гиперпараметры на основе оценки ошибки прогноза двух или более событий. Например, изначально в обучающий набор попадают события без последних трёх событий (двух событий обучающего периода и одного события, оставленного для валидации). Для заданных гиперпараметров определяется ошибка прогноза $Score1$ третьего с конца события. Затем в обучающий набор добавляется третье с конца событие, и для тех же заданных гиперпараметров делается прогноз второго с конца события и определяется ошибка прогноза $Score2$ этого события. Их средняя оценка $(Score1 + Score2)/2$ и выступает в качестве целевой функции во время оптимизации на сетке для подбора гиперпараметров. Последнее же событие используется лишь для валидации построенной модели. Конечно же, для получения прогноза этого последнего события используется выборка без этого последнего события. По аналогии с методами машинного обучения схематичное изображение использования обучающего и контролирующего набора данных X и Y приведено на рисунке 2.20. Однако на практике всё несколько сложнее, подробнее в главе 4.

В описанном примере используются два события для того, чтобы по ошибке их прогноза подбирать гиперпараметры. Конечно же, можно задействовать и большее количество событий. Но уменьшение количества событий в обучающей выборке может приводить к тому, что прогноз события будет менее точным, что будет влиять на определяемые гиперпараметры. Также вместо среднего значения погрешностей можно брать среднее квадратичное. Вместе с тем характеристики событий могут являться векторами $y_i = (y_{i,1}, y_{i,2}, \dots, y_{i,k})$. Возможные метрики оценки погрешности

прогноза событий представлены в таблице 2.7, где N – количество предсказываемых событий, w_i – веса соответствующих событий.



Источник: составлено автором.

Рисунок 2.20 – Схема обучения в новом подходе прогнозирования событий

Таблица 2.7 – Метрики оценки качества (погрешности) прогноза событий

Метрика	y_i – скаляр	$y_i = (y_{i,1}, y_{i,2}, \dots, y_{i,k})$ – вектор
mean absolute error	$\frac{1}{\sum w_i} \sum_{i=1}^N w_i (\hat{t}_i - t_i + \mu \hat{y}_i - y_i)$	$\frac{1}{\sum w_i} \sum_{i=1}^N w_i \left(\hat{t}_i - t_i + \mu \frac{1}{k} \sum_{j=1}^k \hat{y}_{i,j} - y_{i,j} \right)$
mean relative error	$\frac{1}{\sum w_i} \sum_{i=1}^N w_i \left(\left \frac{\hat{t}_i - t_i}{t_i - t_{i-1}} \right + \mu \left \frac{\hat{y}_i - y_i}{y_i} \right \right)$	$\frac{1}{\sum w_i} \sum_{i=1}^N w_i \left(\left \frac{\hat{t}_i - t_i}{t_i - t_{i-1}} \right + \mu \frac{1}{k} \sum_{j=1}^k \left \frac{\hat{y}_{i,j} - y_{i,j}}{y_{i,j}} \right \right)$
root mean squared error	$\sqrt{\frac{1}{\sum w_i} \sum_{i=1}^N w_i (\hat{t}_i - t_i + \mu \hat{y}_i - y_i)^2}$	$\sqrt{\frac{1}{\sum w_i} \sum_{i=1}^N w_i \left(\hat{t}_i - t_i + \mu \sqrt{\frac{1}{k} \sum_{j=1}^k (\hat{y}_{i,j} - y_{i,j})^2} \right)^2}$
root mean squared relative error	$\sqrt{\frac{1}{\sum w_i} \sum_{i=1}^N w_i \left(\left \frac{\hat{t}_i - t_i}{t_i - t_{i-1}} \right + \mu \left \frac{\hat{y}_i - y_i}{y_i} \right \right)^2}$	$\sqrt{\frac{1}{\sum w_i} \sum_{i=1}^N w_i \left(\left \frac{\hat{t}_i - t_i}{t_i - t_{i-1}} \right + \mu \sqrt{\frac{1}{k} \sum_{j=1}^k \left(\frac{\hat{y}_{i,j} - y_{i,j}}{y_{i,j}} \right)^2} \right)^2}$

Источник: составлено автором.

Ошибка прогноза при оптимальных значениях гиперпараметров может выступать индикатором того, насколько хорошо выбранная модель механизма образования событий подходит для получения прогноза будущего события для выбранного источника (клиента). Если эта ошибка достаточно мала, то тогда можно доверять модели и сделанному прогнозу будущего события. Если

эта ошибка велика, то, возможно, стоит поискать другую модель механизма образования событий.

2.9 Проверка адекватности модели механизма образования событий

В эконометрике при определении адекватности модели для решаемой задачи используется проверка того, что фактическое значение эндогенной переменной попадёт в соответствующий доверительный интервал, отложенный от прогнозного значения (от математического ожидания). Тогда для проверки адекватности модели, прогнозирующей события, надо построить доверительный интервал для момента времени будущего события \hat{t}_{n+1} (для значения события \hat{y}_{n+1} желательно).

Можно выделить несколько подходов по расчёту доверительного интервала.

1) Подход на основе расчёта дисперсии прогноза. Разброс прогноза момента будущего события \hat{t}_{n+1} с помощью модели механизма образования событий связан с погрешностью восстановления параметров и дополнительной погрешностью, связанной с экстраполяцией этих параметров. Если предположить, что экстраполяция параметров была проведена идеально, то всё упирается в погрешность определения параметров. В сквозном примере параметрами модели образования событий являются параметры $P_1(t)$, P_2 и P_3 . Статические параметры P_2 и P_3 могут быть определены достаточно точно, поэтому их погрешностью можно пренебречь.

При восстановлении параметра $P_1(t)$ минимизировалась сумма квадратов наблюдаемых интегралов y_i и оценок этих интегралов $\hat{y}_i = [C_{int}g]_i = \sum_{j=1}^m C_{ij}^{int} g_j$, где C_{int} определяется так же, как в (2.37) (здесь под \hat{y}_i понимается именно значение, которое получается при оценке методом МНК задачи сплайновой коллокации, а не то, что возвращает механизм образования событий). Математически можно вычислить ковариационную

матрицу для прогнозных значений \hat{y}_i^{new} в новых точках, в том числе для области, на которой происходит экстраполяция

$$\text{Cov}(\hat{y}_{int}^{new}) = C_{int}^{new} D^{-1} (C_{int}^T W_{int} W_{int}^T C_{int} \hat{\sigma}_{int}^2) D^{-1T} C_{int}^{newT} + \hat{\sigma}_u^2, \quad (2.63)$$

где C_{int}^{new} – матрица размерностью $n^{new} \times m$, которая заполняется так же, как и C_{int} в формуле (2.37), но для новых n^{new} наблюдений, $D = C_{int}^T W_{int} C_{int} + \alpha K$, $\sigma_u^2 = \sum_{i=1}^n u_i^2 / (n - m - 1)$, где $u_i = y_i - \hat{y}_i$ – разница наблюдаемых и оценённых интегралов (при обучении).

Примечание – Для расчета ковариационной матрицы используется $\text{Cov}(C^T z) = C^T \text{Cov}(z) C$, имеем $\text{Cov}(\hat{y}) = \text{Cov}(Cg + U) = C \text{Cov}(g) C^T + \hat{\sigma}_u^2$, где $\sigma_u^2 = \sum_{i=1}^n u_i^2 / (n - m - 1)$, а $u_i = y_i - \hat{y}_i$.

Если $g = D^{-1} \times (C_f^T W_f Y_f + \mu C_{df}^T W_{df} Y_{df} + \nu C_{d^2f}^T W_{d^2f} Y_{d^2f} + \psi C_{int}^T W_{int} Y_{int})$, где $D = C_f^T W_f C_f + \mu C_{df}^T W_{df} C_{df} + \nu C_{d^2f}^T W_{d^2f} C_{d^2f} + \psi C_{int}^T W_{int} C_{int} + \alpha K$, то

$$\text{Cov}(g) = D^{-1} \left(C_f^T W_f W_f^T C_f \hat{\sigma}_f^2 + C_{df}^T W_{df} W_{df}^T C_{df} \mu^2 \hat{\sigma}_{df}^2 + C_{d^2f}^T W_{d^2f} W_{d^2f}^T C_{d^2f} \nu^2 \hat{\sigma}_{d^2f}^2 + C_{int}^T W_{int} W_{int}^T C_{int} \psi^2 \hat{\sigma}_{int}^2 \right) D^{-1T}.$$

Тогда

$$\text{Cov}(\hat{y}_{int}^{new}) = C_{int}^{new} D^{-1} \left(C_f^T W_f W_f^T C_f \hat{\sigma}_f^2 + C_{df}^T W_{df} W_{df}^T C_{df} \mu^2 \hat{\sigma}_{df}^2 + C_{d^2f}^T W_{d^2f} W_{d^2f}^T C_{d^2f} \nu^2 \hat{\sigma}_{d^2f}^2 + C_{int}^T W_{int} W_{int}^T C_{int} \psi^2 \hat{\sigma}_{int}^2 \right) D^{-1T} C_{int}^{newT} + \hat{\sigma}_{int}^2.$$

Если для восстановления используются только интегралы, то $g = (C_{int}^T W_{int} C_{int} + \alpha K)^{-1} C_{int}^T W_{int} Y_{int}$, и

$$\text{Cov}(g) = D^{-1} C_{int}^T W_{int} W_{int}^T C_{int} D^{-1T} \hat{\sigma}_{y_{int}}^2,$$

$$\text{Cov}(\hat{y}_{int}) = C_{int}^{new} D^{-1} (C_{int}^T W_{int} W_{int}^T C_{int} \hat{\sigma}_{int}^2) D^{-1T} C_{int}^{newT} + \hat{\sigma}_{int}^2,$$

где $D = C_{int}^T W_{int} C_{int} + \alpha K$.

Строго говоря, новые точки \hat{t}_{n+1} , которые находятся в будущем, выходят за область определения сплайна, т.е. располагаются правее самого последнего узла s_m . Тем не менее полученные формулы можно по-прежнему использовать, зафиксировав $k = m - 1$ в формулах (2.33), (2.35), (2.36), чтобы использовался последний интервал между узлами s_{m-1} и s_m , положив $L = 0$, $t_u^a = s_m$, $t_u^b = \hat{t}_{n+1}$.

Однако здесь есть несколько существенных проблем. Во-первых, если вспомнить, что на концах сплайн ведёт себя как прямая линия, расчёт $\hat{y}_{n+1}^{new} = [C_{int}^{new} g]_{i=n+1}$ будет некорректным для точки \hat{t}_{n+1} . Расчёт ковариации в этой новой точке тоже будет некорректным (мы раскладывали на гармоники, чтобы провести экстраполяцию). Во-вторых, у нас m (количество узлов сплайна) превышает n (количество наблюдений), расчёт σ_u^2 становится невозможен. Для оценки m параметров по n наблюдениям используется регуляризация, из-за чего оценки становятся смещёнными, формула (2.63) для смещённых оценок не работает. Можно допустить, что дисперсия нового интеграла \hat{y}_{n+1} будет соответствовать средней дисперсии прогнозов обучающих интегралов.

Примечание – На самом деле на конце сплайна, если не менять формулы, расчёт интеграла $\hat{y}_{n+1}^{new} = [C_{int}^{new} g]_{i=n+1}$ даст площадь не под прямой, а под последним полиномом (определяемым узлами s_{m-1} и s_m) на новом участке. Если же учесть, что вторая и третья производные обнуляются, то $\hat{y}_{n+1}^{new} = g_m(\hat{t}_{n+1} - s_m) + g'_m(\hat{t}_{n+1} - s_m)^2/2$, где g'_m первая производная, которая перестаёт меняться после узла s_m , $g'_m = \frac{1}{h_{m-1}} g_m - \frac{1}{h_{m-1}} g_{m-1} + \frac{h_{m-1}}{6} \gamma_{m-1}$.

Другим подходом будет вмешательство в алгоритм Куинна и Фернандеса разложения функции на гармоники, в ту его часть, когда определяются амплитуды синусов и косинусов регрессионными методами для заданного набора частот. Тогда для регрессионной модели на самом последнем шаге, когда вычислена последняя гармоника, можно будет воспользоваться формулой (1.50) для определения дисперсии значений восстановленной функции в новых точках. Потом посчитать дисперсию всего интеграла \hat{y}_{n+1} как сумму дисперсий за интервал времени между последним наблюдением t_n и прогнозным моментом времени будущего события \hat{t}_{n+1} .

Зная дисперсию интеграла \hat{y}_{n+1} , можно предположить, что момент времени завершения запаса \hat{t}_{n+1} будет обладать такой же относительной дисперсией. То есть, если отклонение интеграла составляет 5%, то и отклонение времени составляет 5%. Однако даже такой подход не совсем верный.

2) Подход на основе расчёта дисперсии с помощью имитационного моделирования. В исследовании [210], посвящённом вопросу выбора оптимального коэффициента сглаживания с помощью имитационного моделирования (приблизительно на 1000 моделях), определялось, какой коэффициент сглаживания α минимизирует погрешность $R_{sqr}(\alpha)$ восстановления исходной функции $f(t)$ в виде сплайна $g(t, \alpha)$

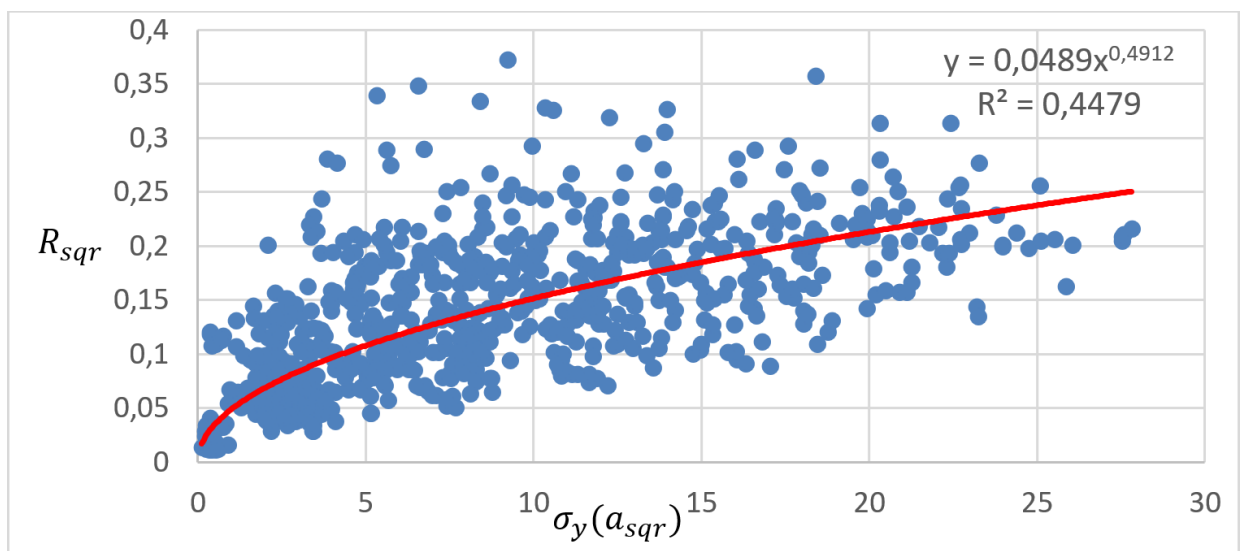
$$R_{sqr}(\alpha) = \sqrt{\frac{1}{t_n - t_1} \sum_{t=t_1}^{t_n} \left(\frac{f(t) - g(t, \alpha)}{f(t)} \right)^2}. \quad (2.64)$$

В результате были определены оптимальные коэффициенты сглаживания $\alpha^* = \arg \min_{\alpha} R_{sqr}(\alpha)$. При этих оптимальных значениях определялся уровень приглушения исходных наблюдений (в %) $\sigma_y(\alpha^*)$

$$\sigma_y(\alpha) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} \left(\frac{y_i - \hat{y}_i}{\hat{y}_i} \right)^2} \times 100\%. \quad (2.65)$$

Для некоторых экспериментов уровень приглушения наблюдений $\sigma_y(a^*)$ был близок к нулю, но погрешность восстановления истинной функции $R_{sqr}(a^*)$ при оптимальном коэффициенте сглаживания была далеко от нуля. Другими словами, остатки (невязки) в методе наименьших квадратов были очень маленькими, но восстановленная по интегралам функция $g(t, a^*)$ сильно отличалась от исходной $f(t)$.

На практике истинная функция $f(t)$ неизвестна, единственное, что можно наблюдать, это величину приглушения наблюдений $\sigma_y(\alpha)$ для выбранного коэффициента сглаживания α . Если предположить, что во время восстановления функции используется оптимальный коэффициент сглаживания a^* , то по уровню приглушения наблюдений $\sigma_y(a^*)$ можно приблизительно оценить погрешность восстановления функции $R_{sqr}(\alpha)$. Зависимость получилась следующей, рисунок 2.21.



Источник: составлено автором.

Рисунок 2.21 – Зависимость погрешности восстановления исходной функции от уровня приглушения наблюдений при оптимальных коэффициентах сглаживания

Для описания зависимости R_{sqr} от σ_y лучше всего подходит степенная модель. После логарифмирования получились следующие оценки параметров.

$$\text{Ln}(\hat{R}_{sqr}) = \begin{matrix} -3,0175 & + & 0,49116 \cdot \text{Ln}(\sigma_y) & + & \varepsilon \\ (0,02948) & (0,01414) & (0,41482) & & \end{matrix} \quad (2.66)$$

Дисперсия для погрешности \hat{R}_{sqr} восстановления функции (используется правило расчёта дисперсии для сложной функции

$$D(Y(x_1, \dots, x_n)) = \left(\frac{\partial Y}{\partial x_1}\right)^2 D(x_1) + \dots + \left(\frac{\partial Y}{\partial x_n}\right)^2 D(x_n)$$

$$D(\hat{R}_{sqr}) = D(e^{\text{Ln}(\hat{R}_{sqr})}) = (e^{\text{Ln}(\hat{R}_{sqr})})^2 \cdot D(\text{Ln}(\hat{R}_{sqr})) = (\hat{R}_{sqr})^2 \cdot D(\text{Ln}(\hat{R}_{sqr})), \quad (2.67)$$

$$D(\text{Ln}(\hat{R}_{sqr})) = 0,02948^2 + 0,01414^2 \cdot [\text{Ln}(\sigma_y)]^2 + 0,41482^2. \quad (2.68)$$

Только что была рассчитана дисперсия для относительной погрешности R_{sqr} восстановления функции $\hat{f}(t)$ (т.е. разброс разброса). Тогда дисперсия значений восстановленной функции (лучше сказать – квадрат относительных отклонений) будет (подробнее в [215, с. 156–158])

$$\begin{aligned} D(\hat{f}(t)) &= (\hat{R}_{sqr})^2 + D(\hat{R}_{sqr}) = (\hat{R}_{sqr})^2 \left(1 + D(\text{Ln}(\hat{R}_{sqr}))\right) = \\ &= (0,04892 \cdot \sigma_y^{0,49116})^2 \left(1,17294 + 0,01414^2 \cdot [\text{Ln}(\sigma_y)]^2\right). \end{aligned} \quad (2.69)$$

Дисперсия интеграла $D(\hat{y}_{n+1})$ определяется как сумма дисперсий для абсолютных значений восстановленной функции по количеству дней до следующего события. Однако у нас имеется относительное отклонение $D(\hat{f}(t))$. Тогда нам удобно так же рассчитать квадрат относительных отклонений для интеграла (а не абсолютных). Если предположить, что на интервале между событиями восстановленная функция $\hat{f}(t)$ отклоняется от исходной $f(t)$ в одну и ту же сторону на одну и ту же величину, тогда интеграл является однократным испытанием случайной величины (как отклонилась вся сумма). В этом случае квадрат относительных отклонений интеграла и функции совпадает

$$D(\hat{y}_{n+1}) = D(\hat{f}(t)). \quad (2.70)$$

Без этого предположения, т.е. при независимости отклонений $\hat{f}(t)$ от $f(t)$ на интервале между событиями, интеграл \hat{y}_{n+1} представляет собой сумму независимых величин $\hat{f}(t)$. Но каждая величина $\hat{f}(t)$ меньше самого

интеграла приблизительно во столько раз, сколько величин суммируется (количество дней). Тогда при независимости отклонений

$$D(\hat{y}_{n+1}) = D(\hat{f}(t)) / \widehat{\Delta t}, \quad (2.71)$$

где $\widehat{\Delta t} = \hat{t}_{n+1} - t_n$.

Квадрат относительного отклонения для момента времени будущего события $D(\hat{t}_{n+1})$ соответствует квадрату относительного отклонения для интеграла

$$D(\hat{t}_{n+1}) = D(\hat{y}_{n+1}). \quad (2.72)$$

При построении доверительного интервала нас интересует среднее квадратичное (относительное) отклонение

$$\sigma(\hat{t}_{n+1}) = \sqrt{D(\hat{t}_{n+1})}. \quad (2.73)$$

Например, если уровень приглушения наблюдений σ_y составит 5%, то расчёт по формулам (2.69), (2.70), (2.72), (2.73) даёт $\sigma(\hat{t}_{n+1}) = 0,11683$ или 11,7% (а при $\sigma_y = 10\%$ имеем $\sigma(\hat{t}_{n+1}) = 16,42\%$).

Так как $\sigma(\hat{t}_{n+1})$ показывает относительные отклонения, то для построения доверительного интервала в днях надо откладывать от \hat{t}_{n+1} произведение $\widehat{\Delta t} \cdot \sigma(\hat{t}_{n+1})$, где $\widehat{\Delta t} = \hat{t}_{n+1} - t_n$ – предполагаемый интервал между событиями. Доверительный интервал в 95% приблизительно будет

$$t_{n+1} \in [\hat{t}_{n+1} - 2 \cdot \widehat{\Delta t} \cdot \sigma(\hat{t}_{n+1}); \hat{t}_{n+1} + 2 \cdot \widehat{\Delta t} \cdot \sigma(\hat{t}_{n+1})]. \quad (2.74)$$

Например, при уровне приглушения наблюдений σ_y , равном 5%, доверительный интервал составляет $\pm 23,4\%$ от интервала между событиями (при $\sigma_y = 10\%$ получим $\pm 32,84\%$).

Заметим, что при большой погрешности можно получить границу доверительного интервала в прошлом. Корректнее рассчитывать погрешность не аддитивно, а мультипликативно. Так, если отклонение функции $\hat{f}(t)$ предполагается 100% в большую сторону, т.е. в 2 раза, то интервал между событиями должен сократиться тоже в 2 раза, а не на все 100% (до нуля или

уйти в прошлое при больших отклонениях). Нижняя граница доверительного интервала тогда

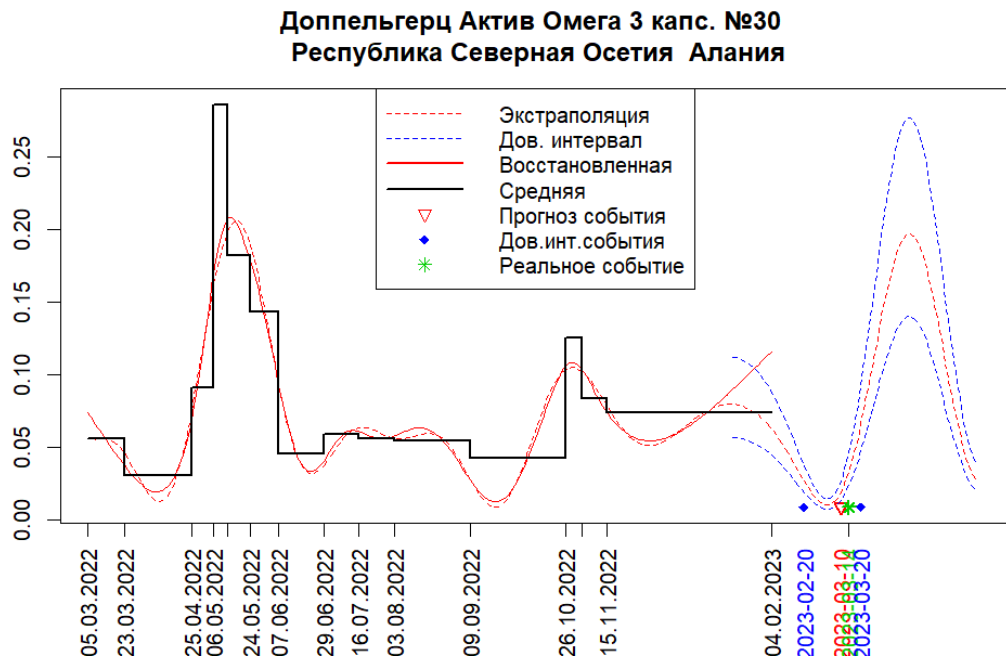
$$t_n + \frac{\widehat{\Delta t}}{1 + 2\sigma(\hat{t}_{n+1})} = \hat{t}_{n+1} - \widehat{\Delta t} \left(\frac{2\sigma(\hat{t}_{n+1})}{1 + 2\sigma(\hat{t}_{n+1})} \right). \quad (2.75)$$

Но верхнюю границу доверительного интервала лучше оставить аддитивной. Иначе, если считать по той же логике, когда отклонение функции $\hat{f}(t)$ составит 100% аддитивно в меньшую сторону, то расхода запаса не будет, знаменатель $1 - 2\sigma_{\widehat{\Delta t}}$ обратится в 0, событие произойдет через бесконечность времени. При отклонениях больше 100% верхняя граница будет в прошлом, что теряет любой смысл. Поэтому верхнюю границу следует рассчитывать аддитивно.

В итоге доверительный интервал будет

$$t_{n+1} \in \left[\hat{t}_{n+1} - \widehat{\Delta t} \cdot \left(\frac{2\sigma(\hat{t}_{n+1})}{1 + 2\sigma(\hat{t}_{n+1})} \right); \hat{t}_{n+1} + \widehat{\Delta t} \cdot 2\sigma(\hat{t}_{n+1}) \right]. \quad (2.76)$$

3) Подход на основе моделирования механизма образования событий. Идея похода достаточно проста, надо использовать в механизме образования событий восстановленную функцию $\hat{f}(t)$, заниженную или завышенную на величину соответствующей нужной доверительной вероятности (но при занижении в меньшую сторону лучше занижать «во» сколько-то раз, а не «на» сколько-то, т.е. мультипликативно, иначе при больших отклонениях можно получить отрицательные значения); величину отклонения $\hat{f}(t)$ рассчитывать на основе формулы (2.69). В результате будут получены два прогноза события, ранее и позднее, которые можно назвать границами доверительного интервала, рисунок 2.22.



Источник: составлено автором.

Рисунок 2.22 – Построение доверительного интервала, занижая и завышая параметр $P_1(t)$

Заметим, что эти три подхода расчёта доверительного интервала базируются на данных. Чем-то это похоже на то, что происходит в эконометрике при низком коэффициенте детерминации (данные не могут полностью объяснить эндогенную переменную, доверительный интервал получается достаточно большим).

4) Подход на основе ошибки прогноза *Score*. Во время подбора оптимальных значений гиперпараметров у нас была оценка ошибки прогноза событий *Score*, формула (2.62). Исследователь может сам задавать границы, при которых он будет доверять построенной модели. Например, отклонение фактического события от прогнозного не должно быть больше 30%. Заметим, что в трёх предыдущих подходах, где в зависимости от данных могла рассчитаться большая или маленькая дисперсия прогноза, доверительный интервал мог оказаться большим или маленьким для одной и той же доверительной вероятности. В этом подходе исследователь сам для себя определяет, какое отклонение считать ещё приемлемым, а какое уже нет.

Было рассмотрено три способа построения доверительного интервала и способ на основе ошибки прогноза. Если фактическое будущее событие t_{n+1}

попадает в рассчитанный доверительный интервал (2.74) или (2.76), то можно принять гипотезу об адекватности модели. Если будущее событие t_{n+1} не попадает в рассчитанный доверительный интервал, то модель признаётся неадекватной. Это говорит о том, что на одном из этапов была допущена ошибка. Ошибка могла быть допущена как на этапе экстраполяции (выбрана неправильная модель), так и на этапе восстановления функции (неправильно подобран коэффициент сглаживания или было недостаточно данных) и, конечно же, на этапе, когда выдвигалось предположение о модели механизма образования событий.

О критериях оценки качества моделей

С развитием цифровых технологий и соответствующей трансформацией системы управления в государственном секторе, общественных институтах и в частном бизнесе прогнозные модели всё чаще используются для поддержки принятия решений, прогнозирования социально-экономических процессов, оптимизации ресурсов и повышения качества предоставляемых услуг. Оценка качества прогноза по формализованным математическим критериям может требовать соблюдения гипотез о выбранном законе распределения величин, получающихся в результате случайных реализаций при неизменных условиях. Однако в реальном мире, как правило, имеется лишь одна реализация при постоянно меняющихся условиях, что зачастую делает не вполне корректным применение таких критериев. В сложившейся практике инженеры по машинному обучению прибегают к проверке качества моделей на тестовой выборке.

Внедрение моделей требует не только технической корректности, но и согласованности метрик качества прогноза со стратегическими целями организации или государства. Метрики машинного обучения (accuracy, precision, recall, AUC-ROC, MAE, RMSE и другие) используются с учётом интереса заказчика и характера последствий ошибок. В здравоохранении при диагностике важно не пропустить возможное заболевание (требуется высокий recall, малая доля ложноотрицательных результатов). В банке при выявлении

неплатежеспособных заёмщиков требуется малая доля ложноположительных результатов (низкий FPR), так как это приводит к упущенной выгоде и снижению доверия клиентов. При прогнозе налоговых поступлений ключевой метрикой может стать средняя относительная ошибка (MAPE), поскольку она напрямую связана с бюджетным планированием и распределением ресурсов.

Таким образом, выбор метрики моделей прогнозирования должен соответствовать стратегической цели: минимизация социального вреда, максимизация экономической эффективности, обеспечение справедливости или устойчивости системы. Например, для цели «повышение доступности качественного образования» для модели прогнозирования отсева школьников вместо обычной точности лучше подойдёт метрика балансирующей точности (balanced accuracy). На уровне министерств и ведомств метрики должны быть согласованы с ключевыми показателями эффективности (KPI), установленными в рамках госпрограмм или регламентов. На уровне коммерческих организаций важна интерпретируемость метрик и их перевод в денежный эквивалент (cost-sensitive learning).

При этом цели государственных и коммерческих организаций могут быть динамичны, когда внешние условия заставляют изменить текущую политику и будущие планы. Из-за этого метрики оценки прогнозов также должны быть гибкими и пересматриваемыми. Например, может потребоваться периодическое обновление стоимости ошибок.

Метрики по-своему являются инструментом перевода стратегических целей в измеримые показатели и дальнейшие действия. Корректный выбор метрик обеспечивает высокое качество, а также соответствие моделей миссии организации, интересам общества и требованиям государства. Разработка и применение метрик должны осуществляться в тесном взаимодействии между специалистами по данным, предметными экспертами, менеджерами/регуляторами.

По перечисленным причинам оценкой пригодности моделей прогнозирования событий могут являться не строго формализованные

математические метрики, а основанные на практике организаций критерии. Возможно, что даже менее точный прогноз событий будет давать больший экономический эффект по той причине, что люди будут готовиться к более раннему сроку или к более суровым последствиям грядущего события. Другими критериями могут быть некоторые социальные критерии, которые основываются на повышении счастья населения, демографии, социальной защищённости и т.д. Например, своевременный прогноз позволит объяснить происходящее и успокоить общество, поможет принять суровую реальность и перестроить мышление согласно новым наступающим условиям.

Для примеров из пятой главы было показано, что построение моделей на основе новой методологии позволило обосновать изменение спроса на биологические добавки в период COVID. Расчёты в пятой главе показывают, что увеличение точности прогноза на 1 день сокращают издержки на 0,082% от стоимости самих товаров. Повышение точности может приводит к сокращению затрат на 5%, а это в свою очередь может позволить снизить стоимость товаров, что благоприятно скажется для социума. Дополнительные критерии могут быть связаны с научной и образовательной составляющей, когда прогноз событий будет способствовать появлению новых научных идей, публикации статей по предсказанным происходящим событиям, накоплению знаний и опыта, написанию новых учебников и учебных пособий, связанных с преодолением прогнозируемых событий.

Выводы по главе 2

1) Предложен новый подход к прогнозированию экономических событий, основанный на восстановлении механизмов их образования. Под механизмом понимается детерминированная последовательность операторов над внутренними переменными и параметрами. Подход состоит из пяти этапов: 1 – подготовка данных; 2 – предположение модели механизма образования событий $Program(X, P)$, где $X = \{X_i\}$ – внутренние переменные инициализируются внутри модели, $P = \{P_i\}$ – статичные или динамичные

параметры механизма; 3 – восстановление неизвестных параметров механизма $P = \{P_i\}$ по имеющейся выборке событий $(t_i; y_i)$ аналитическими или численными методами; 4 – экстраполяция параметров механизма P на будущее; 5 – получение прогноза будущих событий $(t_i; y_i)$ с помощью моделирования механизма их образования $Program(X, P)$ с установленными значениями параметров.

2) В качестве сквозного примера рассматривается алгоритмическая модель механизма образования событий, как в модели управления запасами. Показано, что в такой модели является внутренней переменной, а что параметрами. Также дано представление механизма в математической форме. Описаны способы выбора моделей. Рассмотрены более сложные модели механизмов образования событий.

3) Рассмотрены способы восстановления неизвестных параметров P механизма образования событий с помощью регрессионных методов и 10 современных методов машинного обучения. Однако такие методы плохо справляются с восстановлением функции по последовательности интегралов, для чего нужно использовать более сложный метод.

4) Разработан метод восстановления нестационарных параметров на основе метода сплайновой коллокации восстановления функции одновременно по разным функционалам, таким как значения, значения первой и второй производной, определённые интегралы. Используется кубический сглаживающий сплайн в форме значений и значений вторых производных (*value-second derivative representation*). Рассмотрены вопросы выбора оптимального коэффициента сглаживания.

5) Разработаны способы обеспечения положительности/монотонности восстановленной по разным функционалам функции. Подробно разбираются два способа: 1) на основе взятия экспоненты от сплайна; 2) на основе сведения к задаче квадратичного программирования с помощью добавления условий неотрицательности в узлах сплайна. Получены соответствующие формулы. Во втором способе для обеспечения положительности сплайна между узлами

ищется отрицательный экстремум и добавляется либо новый узел с условием неотрицательности, либо новое условие неотрицательности без добавления нового узла. Получены формулы и представлены примеры.

6) Рассмотрены способы экстраполяции значений параметров на будущее либо с помощью регрессионных методов и методов машинного обучения, либо на основе привязывания тенденций к временам года, либо на основе разложения на ограниченное количество гармонических функций по алгоритму Куинна и Фернандеса.

7) Разработан способ определения оптимальных гиперпараметров алгоритмов, используемых в процессе восстановления и экстраполяции параметров механизма образования событий. Показано, как формировать функцию потерь для прогноза событий. Функция потерь минимизируется с помощью поиска на сетке для глобальной оптимизации и алгоритма Нелдера-Мида для локальной оптимизации в окрестности узлов сетки.

8) Разработаны способы проверки адекватности моделей механизмов образования событий на основе нескольких подходов: 1 – на основе расчёта дисперсии прогноза с помощью регрессионных методов; 2 – на основе расчёта дисперсии с помощью имитационного моделирования; 3 – на основе моделирования механизма образования событий с разными параметрами; 4 – на основе ошибки прогноза событий, полученной во время подбора оптимальных значений гиперпараметров.

Глава 3

Восстановление алгоритмических моделей механизмов образования событий с помощью численных методов

Во второй главе одной из возможных форм записи модели механизма образования событий являлась форма в виде алгоритма. Но эта запись использовалась лишь для понимания. Ориентируясь на неё, была составлена математическая форма, которая позволила сформулировать решение задачи восстановления параметров аналитическими методами. В результате во второй главе параметры находились аналитически, задействовались статистические методы и методы сплайновой коллокации. Однако не все модели могут быть приведены к простому виду, для которого найдётся аналитическое решение. Также у исследователя может не быть навыков или желания искать это аналитическое решение. В этой главе рассматриваются способы определения параметров алгоритмических моделей с помощью методов численной оптимизации, также обсуждаются способы автоматического подбора самих алгоритмических моделей.

Материалы данной главы опубликованы автором в работах [212–217].

3.1 Особенности представления модели механизма образования событий в алгоритмической форме

Механизм образования событий представляется в форме алгоритмической модели $Program(X, P)$ как последовательность разных арифметических операторов над внутренними переменными $X = (X_1, \dots, X_{N_X})$ и параметрами $P = (P_1, \dots, P_{N_P})$, где N_X и N_P – количество внутренних переменных и параметров. Внутренние переменные X изменяются только из-за действия операторов внутри модели механизма. Параметры P зависят от внешних условий независимо от того, что происходит в самом механизме. Параметры P могут быть статичными или динамичными (функцией времени, например $P_1(t)$). Параметры могут быть известны P , т.е. заданы заранее, а могут быть неизвестны, их предстоит определить. Внутренние переменные X

обязательно должны быть проинициализированы до начала функционирования механизма образования событий. Проинициализированы они могут быть либо нулем, либо значением одного из параметров P .

Помимо операторов, изменяющих значения внутренних переменных X , обязательно должен присутствовать оператор формирования события, который является ключевым во всём механизме образования событий. Событие образуется в результате выполнения определённого условия. Таким образом, в операторе формирования события присутствует операция сравнения. Сравниваются внутренние переменные X со значением других внутренних переменных X или параметров P . При наступлении заданных условий формируется событие, которое характеризуется моментом времени t'_i (где i – номер сформированного события, а штрих ' показывает, что это смоделированное событие, а не фактическое), а также некоторой информацией y'_i о событии (в сквозном примере это объём пополнения запаса). Информация о событии y'_i может быть скалярной величиной, а может быть и вектором значений. Можно допустить, что самих операторов формирования событий может быть несколько, когда в разных частях механизма происходят проверки разных условий, чтобы сформировать события.

Механизм функционирует во времени t . Начальное и конечное время задаётся извне модели исследователем. Время не является ни параметром, ни переменной. Время изменяется по шагам, например по дням. На каждом шаге выполняется заданная последовательность операторов. По достижении конечного времени будет иметься выборка событий (t'_i, y'_i) . В случае неудачно выбранных значений параметров выборка событий может оказаться пустой (не сформируется ни одного события). Как правило, начальным временем будет являться момент самого первого события t_1 в исходной выборке наблюдений. Тогда первое событие, полученное в результате функционирования механизма, будет являться уже вторым событием по отношению к исходной выборке. Моделирование оканчивается не только по

достижении конечного времени, но при генерации заданного количества событий n_{max} , для чего ведётся подсчёт образованных событий.

Примечание – Можно развить идею использования непрерывного времени, тогда механизм превращается в систему дифференциальных уравнений. Это будет похоже на модели системной динамики, но с дополнительным образованием дискретных событий

В качестве демонстрации возьмём всё тот же сквозной пример, в котором события образуются, как в моделях управления запасами. Параметрами являются $P_1(t)$ – нестационарный спрос, P_2 – критический уровень запасов, P_3 – максимальный уровень запасов. Имеется всего одна внутренняя переменная X_1 – текущие запасы. Алгоритмическая модель механизма образования событий вместе с дополнительными приготовлениями будет выглядеть следующим образом:

- 1) $t = t_{start}$; # инициализация начального времени
- 2) $n = 0$; # количество сформированных событий
- 3) $X_1 = P_3$; # инициализация текущих запасов
- 4) Пока ($t \leq t_{end}$)
- 5) $X_1 = X_1 - P_1(t)$;
- 6) Если ($X_1 \leq P_2$), то
- 7) Создать событие ($t, P_3 - X_1$);
- 8) $n = n + 1$;
- 9) Если ($n \geq n_{max}$)
- 10) Выход из цикла;
- 11) $X_1 = P_3$;
- 12) Продвинуть время t и обновить параметры ($P_1(t)$);
- 13) Конец цикла

Здесь функция «Создать событие» добавляет в выборку событие с текущим временем и значением (объёмом покупки/заказа), равным тому, сколько не хватает до максимума. Процедура «Продвинуть время и обновить параметры» продвигает время на один шаг, в данном примере – на день, обновление параметров изменяет их значение в соответствии с временем модели. На данной схеме операторов не представлены дополнительные

служебные строки, связанные с тем, как реализованы динамические параметры.

Параметры $P_1(t)$, P_2 , P_3 являются искомыми. Но чтобы механизм функционировал, необходимо задать им какие-то начальные значения, которые в результате оптимизации будут корректироваться. Заметим, что неизвестные параметры нельзя назвать экзогенными переменными (такие параметры не наблюдаются напрямую, наблюдаются только дискретные события). Неизвестные параметры можно сравнить с коэффициентами эконометрических моделей (a_0, a_1, a_2, \dots), но при этом они могут быть динамическими (т.е. $a_0(t), a_1(t), a_2(t), \dots$). Дополнительно могут присутствовать известные параметры (например ВВП или уровень инфляции). Эти параметры не надо будет определять, так как они уже известны (можно назвать экзогенными). Что неизвестные, что известные параметры будут влиять на внутренние переменные X . Хочется сделать акцент на том, что если в эконометрике или анализе данных исследователь составляет математическую модель, то в этом подходе исследователь составляет алгоритмическую модель образования событий. В эконометрике параметры определяются в основном методом наименьших квадратов, а здесь используются численные методы оптимизации (функция потерь может быть также квадратичной).

Серьёзных ограничений на представление механизма образования событий в алгоритмической форме нет. Механизм может быть описан произвольным алгоритмом. Однако этот механизм всё же должен быть детерминированной последовательностью операторов. То есть в нём не должны использоваться случайные числа. События должны образовываться в результате действия именно алгоритма, а не «броска кубика». Иначе помимо параметров механизма придётся определять параметры генератора случайных чисел. Разброс или погрешность в прогнозировании событий может быть связана с погрешностью восстановления параметров механизма образования событий, а не с тем, «что выпало на кубике». Неопределённость, конечно же,

будет присутствовать. Во-первых, сам механизм образования событий может быть недостаточно детальным, возможно, он описывает только основные тенденции. Во-вторых, сами параметры могут являться некоторыми агрегированными абстрактными характеристиками, которые в действительности формируются сложным и неизвестным способом. И когда происходит восстановление значения некоторого параметра, то как будто восстанавливается среднее значение (по реализациям) неизвестной ненаблюдаемой величины (случайного процесса). В-третьих, зачастую в реальности сами события фиксируются с некоторой погрешностью. То есть событие уже произошло, но его заметили позже. В итоге имеющаяся в распоряжении выборка может быть рассмотрена как зашумлённые данные. Восстановление параметров по зашумлённым данным событий будет давать погрешность. Механизм с такими параметрами будет давать погрешность для прогноза событий (по дисперсии параметров можно будет построить доверительный интервал для событий).

Описание динамических параметров

В отличие от статических параметров динамические параметры обозначают изменяющиеся со временем внешние условия, т.е. являются функцией времени $P_j(t)$. Такие динамические параметры задаются в виде непрерывных гладких функций, которые будем описывать кубическими сплайнами $g(t)$ (так как во второй главе восстановление параметров с помощью сплайнов показало наилучший результат). Но выражаться они будут через изменяющуюся скачками третью производную $g'''(s_k)$ в каждом узле s_k . Чтобы задать значение изменяющегося со временем параметра в виде сплайна $g(t)$, требуется:

1) задать узлы сплайна $s_1 < s_2 < \dots < s_m$ (количество узлов m и их местоположение выбирается исследователем априори, самый распространённый способ — это выбирать узлы совпадающими с наблюдениями или распределять их равноудалённо);

2) задать начальное значение, первую и вторую производную в самом первом узле $g(s_1)$, $g'(s_1)$, $g''(s_1)$;

3) задать значения третьей производной $g'''(s_k)$ во всех узлах, предшествующих моменту времени t , ($k: s_k < t$).

Примечание – Можно предположить, что на определённом уровне абстракции изменения параметров происходит непрерывно. При необходимости скачкообразные резкие изменения можно выражать операторами внутри самого механизма образования событий. Также можно строить сложные иерархические модели, когда поток дискретных событий одного механизма будет изображать изменения в параметре для вышестоящего по иерархии механизма.

Данное представление удобно тем, что не нужно определять параметры, которые на оси времени лежат «в будущем» по отношению к интересующему моменту времени t . Такая форма определения сплайна всё равно эквивалента представлению через значения и вторые производные (value-second derivative representation), которое, в свою очередь, эквивалентно классическому представлению полинома с четырьмя параметрами при степенях переменной.

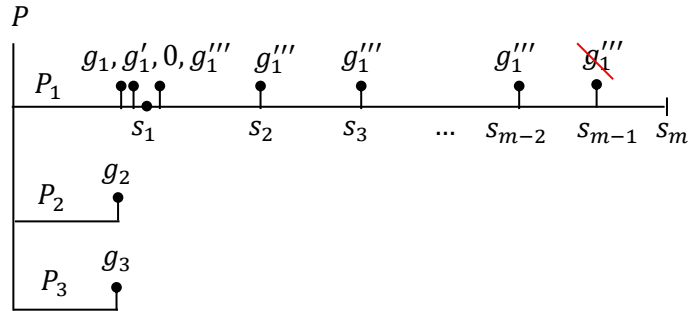
В последнем узле s_m значение третьей производной можно не определять, так как после последнего узла значение сплайна рассчитывать не нужно. Дополнительно заметим, что по условиям натурального сплайна в начале и в конце интервала вторая производная обращается в ноль. В результате вторую производную в начальном узле можно не подбирать. Также можно будет не подбирать значения третьей производной в предпоследнем узле s_{m-1} , так как это значение можно выразить через ранее определённые значения. Если $h_k = s_{k+1} - s_k$ – расстояние между узлами, $g''(s_1) = 0$, то $g''(s_m) = \sum_{k=1}^{m-1} h_k g'''(s_k) = 0$, откуда

$$g'''(s_{m-1}) = -\frac{\sum_{k=1}^{m-2} h_k g'''(s_k)}{h_{m-1}}. \quad (3.1)$$

Для сквозного примера, у которого первый параметр динамический, а второй и третий – статические, требуется определить значения, указанные в выражении (3.2).

$$P = \begin{pmatrix} P_1 = (g_1(s_1), g_1'(s_1), 0, g_1'''(s_1), g_1'''(s_2), g_1'''(s_3), \dots, g_1'''(s_{m-2})) \\ P_2 = g_2 \\ P_3 = g_3 \end{pmatrix}. \quad (3.2)$$

Схематическое изображение определяемых значений представлено на рисунке 3.1.



Источник: составлено автором.

Рисунок 3.1 – Пример схемы значений для описания параметров механизма формирования событий, P_1 – динамический параметр, P_2 и P_3 – статические параметры

В функции «Продвинуть время t и обновить параметры» реализуется расчёт нового значения для всех динамических параметров. Значения $g(t)$, $g'(t)$ и $g''(t)$ обновляются относительно значений на предыдущем шаге по следующим формулам (и в указанном порядке)

$$g(t) = g(t) + g'(t) + \frac{1}{2}g''(t) + \frac{1}{6}g'''(s_k), \quad (3.3)$$

$$g'(t) = g'(t) + g''(t) + \frac{1}{2}g'''(s_k), \quad (3.4)$$

$$g''(t) = g''(t) + g'''(s_k), \quad (3.5)$$

где k – номер узла, предшествующий моменту времени t .

Далее предстоит проводить оптимизацию и определять значения как статических, так и динамических параметров. Так как данные могут быть зашумлены, для борьбы с шумом при оптимизации используется регуляризация. К функции потерь добавляется штраф на чрезмерную гладкость (меру кривизны, шероховатость, roughness). Для кубического сплайна штраф на шероховатость рассчитывается как $\int_{s_1}^{s_m} (g''(t))^2 dt$. Этот штраф можно рассчитывать как во время функционирования механизма образования событий, суммируя квадрат второй производной, так и после его

завершения. Интегрируя квадрат величины $g''(t) = g''(s_k) + (t - s_k)g'''(s_k)$, штраф на шероховатость каждого динамического параметра определяется как

$$\int_{s_1}^{s_m} (g''(t))^2 dt = \sum_{k=1}^{m-1} \frac{(g''(s_k) + g'''(s_k)h_k)^3 - (g''(s_k))^3}{3g'''(s_k)}, \quad (3.6)$$

где $g''(s_k) = g'''(s_1)h_1 + \dots + g'''(s_{k-1})h_{k-1}$.

3.2 Определение параметров алгоритмических моделей механизмов образования событий

По окончании функционирования механизма образования событий получается выборка, которая сравнивается с имеющейся выборкой исходных данных. Обозначим моменты появления событий как t_i и t'_i для фактических наблюдений и событий, полученных в результате моделирования. Аналогично обозначим характеристики y_i и y'_i для исходных наблюдений и для событий, полученных в результате моделирования.

Функция потерь

Введём квадратичную функцию потерь, показывающую различия в двух выборках. Из-за различия как в моментах времени, так и в характеристиках событий задача получается многокритериальной.

$$S(P) = \sum_{i=2}^n (t_i - t'_i)^2 + \mu \sum_{i=2}^n (y_i - y'_i)^2 \xrightarrow{P} \min. \quad (3.7)$$

Здесь нумерация начинается со второго события, так как первое событие служит моментом времени, с которого начинается моделирование; μ – весовой коэффициент, показывающий, какая доля внимания уделяется отклонениям для моментов времени, а какая – для характеристик; n – размер выборки.

Логично перейти к безразмерным относительным отклонениям, чтобы выбор разных единиц измерения не влиял на функцию потерь. Для времени берём отношение отклонения к интервалу времени между возникновением фактических событий. Для характеристик событий берём отношение к фактическому значению.

$$S(P) = \sum_{i=2}^n \left(\frac{t_i - t'_i}{t_i - t_{i-1}} \right)^2 + \mu \sum_{i=2}^n \left(\frac{y_i - y'_i}{y_i} \right)^2 \xrightarrow{P} \min . \quad (3.8)$$

Наконец, задействуем регуляризацию (штраф на гладкость/шероховатость функции) для каждого динамического параметра Pen_j , формула (3.6). Для статических параметров этот штраф равен нулю. Дополнительно зададим свой вес для этого штрафа каждому параметру. Окончательно имеем следующую функцию потерь.

$$S(P) = \sum_{i=2}^n \left(\frac{t_i - t'_i}{t_i - t_{i-1}} \right)^2 + \mu \sum_{i=2}^n \left(\frac{y_i - y'_i}{y_i} \right)^2 + \sum_{j=1}^{N_P} \alpha_j Pen_j \xrightarrow{P} \min . \quad (3.9)$$

Здесь j – номер параметра, N_P – количество параметров, Pen_j – штраф на гладкость для параметра j , α_j – коэффициент, с которым учитывается данный штраф в общей функции потерь. Заметим: штраф не является относительной характеристикой, поэтому параметр сглаживания надо брать достаточно маленьким, например 10^{-6} .

Замечание. В случае неудачного выбора параметров может быть получена выборка событий, которая по размеру отличается от ожидаемого количества событий (или вообще может быть пустой). В этом случае можно сказать, что функция потерь обращается в бесконечность. Также моделировать механизм образования событий необходимо чуть дольше (задавать конечное время t_{end}), чем время последнего события, чтобы дать шанс сформироваться нужному количеству событий.

Оптимизация

Функция потерь $S(P)$ недифференцируемая, поэтому методы, предполагающие аналитическое вычисление производных, не годятся. Численное определение градиентов также затруднительно, так как в некоторых точках целевая функция может не существовать (возвращает бесконечность).

В имитационном моделировании одним из зарекомендовавших себя методов локальной оптимизации является алгоритм Нелдера-Мида [162]. Алгоритм Нелдера-Мида способен двигаться в направлении лучшего значения, даже если среди выбранных вершин симплекса есть вершины, в которых расчёт целевой функции невозможен. На одной итерации в большинстве случаев происходит только одно оценивание целевой функции. Тем не менее у этого метода есть существенный недостаток – иногда он сходится к неоптимальным значениям. Подробнее об этом в работах [163; 164]. Этот недостаток может быть частично ослаблен при использовании его вместе с методами глобальной оптимизации.

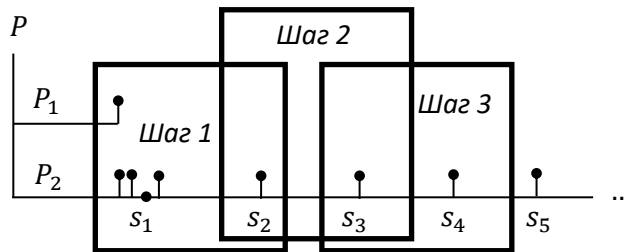
Построенная оптимизационная задача является невыпуклой и может иметь множество локальных экстремумов. Методы локальной оптимизации сходятся только к ближайшему экстремуму. Для поиска глобального экстремума существует несколько эвристических подходов. Самый простой из них – это оптимизация на сетке, когда оптимизация запускается многократно из разных начальных точек, после чего из обнаруженных локальных экстремумов выбирается наилучший. Другой подход – это генетические алгоритмы [165; 166]. В них также некоторым образом надо задать начальную популяцию (множество точек). Вероятность отбора особей из популяции при минимизации обратно пропорциональна значению целевой функции, потомки наследуют часть хромосом (параметров) от родителей, дополнительно предусмотрена процедура случайной мутации отдельных генов. Ещё один современный подход – это разбросанный поиск (scatter search, поиск по разбросу) [167; 168]. В нём основная идея заключается в том, что применяется некоторая мера разброса между точками (например, евклидово расстояние), которая используется совместно со значениями целевой функции при добавлении этих точек в популяцию. В итоге популяция для дальнейшего поиска представлена точками, которые будут одновременно обладать хорошим значением целевой функции и при этом лучше распределены по пространству.

Примечание – На данный момент разбросанный поиск и алгоритм Нелдера-Мида используются в оптимизаторе OptQuest, который встроен во многие современные средства имитационного моделирования, включая Anylogic.

Генетические алгоритмы и рассеянный поиск используют некоторую рандомизацию при определении следующих точек. В этих методах, как и в обычном методе оптимизации на сетке, происходит локальный поиск в разных стартовых точках. Из-за использования рандомизации в генетических алгоритмах и рассеянном поиске нет гарантии обнаружения глобального оптимума. Поиск на сетке явный и более понятный, не использует рандомизацию. Нахождение глобального оптимума зависит от шага сетки, который выбирает исследователь. Более того, исходные данные и некоторые представления о механизме образования событий могут подсказать, какой шаг сетки следует выбрать по каждой переменной. В итоге для определения параметров процесса образования событий используется оптимизация на сетке и метод Нелдера-Мида для поиска экстремума внутри ячейки сетки. Если во время локальной оптимизации происходит выход за границы ячейки, то алгоритм прекращается и поиск начинается уже из следующей ячейки.

Однако, если каждое значение перебирать на сетке, то размерность задачи получается огромной. Будем подбирать на сетке значения для такого количества узлов сплайна, чтобы получить всего несколько событий, а не все сразу. Получается многошаговый алгоритм. На первом шаге подбираются неизвестные значения на N событий вперёд, для всех таких параметров запускается оптимизация на сетке. При оптимизации нужно добиться того, чтобы первые события, образовавшиеся моделью механизма, были близки к первым событиям исходной выборки. На следующем шаге происходит сдвиг «скользящего окна» на одну позицию, на одно событие, и для него подбираются параметры для N событий. Если $N = 2$, то на первом шаге подбираются на сетке параметры, отвечающие за образование первого и второго события (так как время первого события является отправной точкой, то первое образованное событие на самом деле будет вторым событием в обучающей выборке), на втором шаге подбираются на сетке параметры,

отвечающие за образование второго и третьего события. На рисунке 5.16 изображены значения параметров, перебираемые на сетке на каждом новом шаге для случая $N = 2$ (при условии, что узлы сетки s_k совпадают с наблюдениями).



Источник: составлено автором.

Рисунок 3.2 – Подбор значений на $N = 2$ события вперёд, на первом шаге полностью перебираются на сетке комбинации 5 значений, на втором – комбинации 2 значений, на третьем – тоже комбинации 2 значений

Значения, покинувшие скользящее окно, не подбираются на сетке, но используются в локальной оптимизации. То есть в локальной оптимизации используются все значения до некоторого события. Глобальная оптимизация с помощью поиска на сетке запускает локальную оптимизацию из центра каждой ячейки, но количество ячеек определяется сеткой только для вошедших в скользящее окно значений. Предполагается, что определённые ранее и не вошедшие в скользящее окно значения лежат в окрестности глобального оптимума (используется предположение – значения, соответствующие событию, зависят незначительно от значений, удалённых более, чем на N событий). Таким образом, глобальная оптимизация помогает найти правильный локальный оптимум с помощью поиска на сетке, а далее он только корректируется с помощью локальной оптимизации. Предполагается, что это должно помочь избежать «эффекта бабочки», когда малые расхождения в начале приводят к огромным расхождениям в конце.

Желательно выбирать N (ширину окна событий) как можно больше, но это сильно влияет на объём вычислений. Нужно найти некоторый баланс между временем выполнения всех расчётов и точностью определяемых параметров. Объём вычислений для самого первого шага может быть самым большим, так как на нём определяется больше всего параметров. Для второго

шага объём вычислений может быть наименьшим. Далее с ростом номера шага объём вычислений будет увеличиваться, так как количество оптимизируемых значений будет линейно возрастать (также время моделирования механизма образования событий будет линейно увеличиваться). Описанный приём с использованием скользящего окна событий позволяет обойти проблему экспоненциального роста объёма вычислений.

Пример определения параметров процесса, заданного в алгоритмической форме

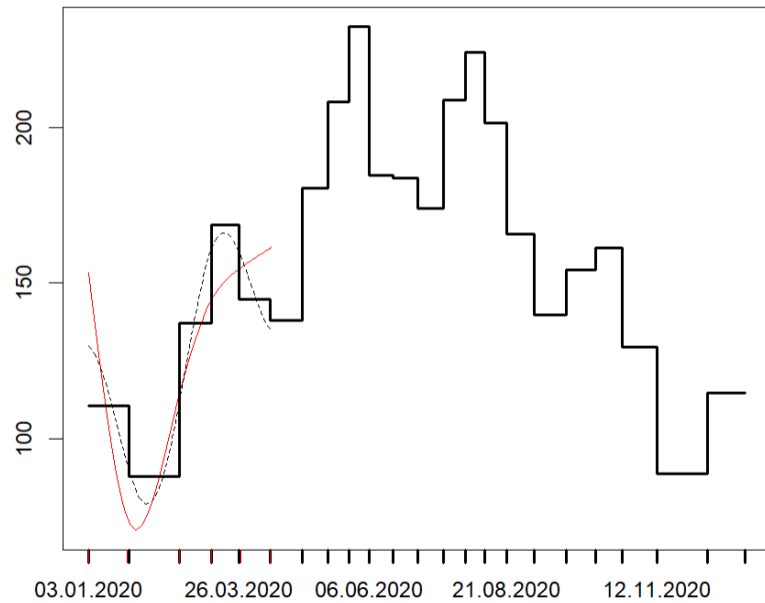
Рассмотрим работу метода на сквозном примере, в котором события образуются, как в моделях управления запасами. Нашей целью будет восстановить значения параметров $P_1(t)$, P_2 , P_3 (спрос, критический уровень, максимальный уровень), располагая только выборкой событий, таблица 3.1. Так же как и в математическом способе восстановления параметров, можно предположить $P_2 = 0$ и не определять его (так как он спутан с параметром P_3). Параметры $P_1(t)$ и P_3 определяем с помощью описанного подхода глобальной и локальной оптимизации.

Таблица 3.1 – Выборка событий

t_i	y_i	t_i	y_i	t_i	y_i	t_i	y_i
03.01.2020	2429	30.04.2020	2526	17.07.2020	2508	09.10.2020	2418
25.01.2020	2461	14.05.2020	2500	29.07.2020	2466	24.10.2020	2457
22.02.2020	2465	26.05.2020	2556	09.08.2020	2418	12.11.2020	2478
11.03.2020	2528	06.06.2020	2402	21.08.2020	2484	10.12.2020	2409
26.03.2020	2462	19.06.2020	2574	05.09.2020	2517	31.12.2020	2478
12.04.2020	2481	03.07.2020	2437	23.09.2020	2467	-	-

Источник: составлено автором.

Выделим больший вес для определения даты событий t_i по сравнению с определением характеристик событий y_i (объём покупок). В функции потерь (3.9) для отклонений характеристик y_i возьмём вес $\mu = 0.1$. Коэффициент регуляризации возьмём $\alpha = 10^{-6}$. Зададим ширину скользящего окна событий $N = 5$. Результат выполнения первого шага алгоритма показан на рисунке 3.3 и в таблицах 3.2 и 3.3.



Гладкая сплошная линия показывает восстановленное значение скорости потребления $P_1(t)$ (шт./день), пунктирная линия показывает искомую функцию потребления, заложенную в модель, ступенчатая линия показывает среднее потребление $y_i/(t_{i+1} - t_i)$, площадь под каждой ступенькой будет объемом покупки y_i .

Источник: составлено автором.

Рисунок 3.3 – Первый шаг алгоритма, нестационарный спрос $P_1(t)$

Таблица 3.2 – Значения параметров после первого шага алгоритма

Параметр	Значение
Начальная скорость потребления	$P_1(s_1) = 153,222$
Первая производная скорости потребления	$\dot{P}_1(s_1) = -4,8323$
Третья производная скорости потребления в узле s_1	$\ddot{P}_1(s_1) = 1,4825 \cdot 10^{-2}$
Третья производная скорости потребления в узле s_2	$\ddot{P}_1(s_2) = -1,4202 \cdot 10^{-2}$
Третья производная скорости потребления в узле s_3	$\ddot{P}_1(s_3) = 5,6984 \cdot 10^{-5}$
Третья производная скорости потребления в узле s_4	$\ddot{P}_1(s_4) = 4,0556 \cdot 10^{-3}$
Третья производная скорости потребления в узле s_5	$\ddot{P}_1(s_5)$ выражается через предыдущие значения
Максимальный запас (истинное значение 2400)	$P_3 = 2371,628$

Источник: составлено автором.

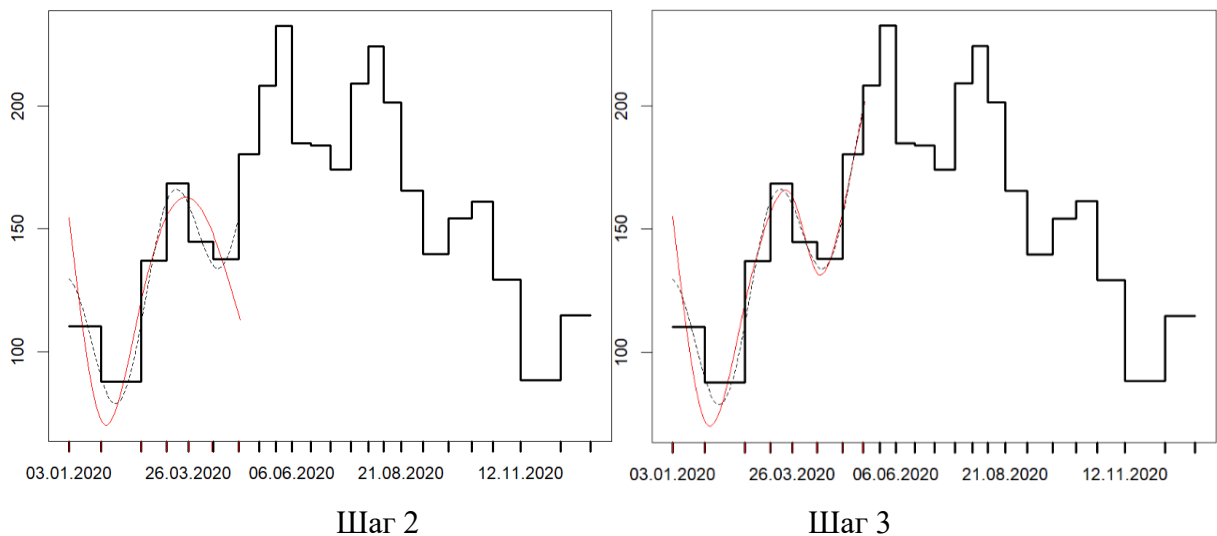
Таблица 3.3 – Полученные события после первого шага алгоритма

Прогноз t_i	Факт t_i	Прогноз y_i	Факт y_i
24.01.2020	25.01.2020	2386,468	2461
22.02.2020	22.02.2020	2485,890	2465
11.03.2020	11.03.2020	2378,407	2528
27.03.2020	26.03.2020	2413,642	2462
12.04.2020	12.04.2020	2532,705	2481

Источник: составлено автором.

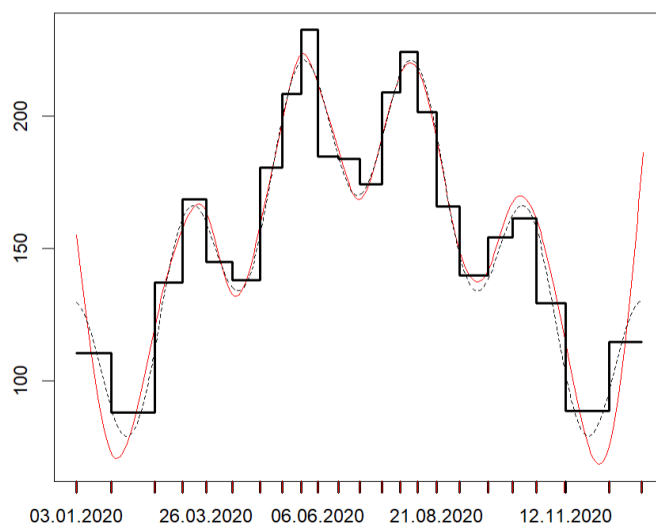
Заметим, что при выборе ширины скользящего окна $N = 4$ (вместо 5) результаты оказываются хуже, об этом подробнее в работе [215, с. 193–196].

На втором шаге скользящее окно событий сдвигается вправо, и для $N = 5$ событий осуществляется подбор параметров, не забывая о значениях параметров, определённых для самого первого сформированного события. Так, значение максимального запаса изменилось с ранее определённого 2371,628 на 2410,712. На третьем и последующем шаге скользящее окно вновь сдвигается на одну позицию, и попавшие в это скользящее окно значения перебираются на сетке, а не попавшие лишь корректируются. Алгоритм останавливается после 18 шага, когда будут подобраны все значения параметров, необходимые для формирования всех 22 событий. Результат работы алгоритма после шагов 2 и 3 изображён на рисунке 3.4, а за все 18 шагов – на рисунке 3.5. Таблица 3.4 показывает сформированные механизмом события.



Источник: составлено автором.

Рисунок 3.4 – Результат работы алгоритма после 2 и 3 шагов



Источник: составлено автором.

Рисунок 3.5 – Результат работы алгоритма после всех 18 шагов

Таблица 3.4 – Полученные события после 18 шага алгоритма

-	Дата					
Исходные	Полное совпадение в 21 из 22 событий только для события с датой 12.11.2020, процесс определил событие на день раньше (11.11.2020)					
Процесс						
-	Значения (объём покупки)					
Исходные	2429	2461	2465	2528	2462	2481
Процесс	-	2468.564	2472.071	2547.122	2464.828	2494.400
Исходные	2526	2500	2556	2402	2574	2437
Процесс	2545,614	2511,836	2566,986	2411,377	2585,020	2445,382
Исходные	2508	2466	2418	2484	2517	2467
Процесс	2498,655	2470,175	2406,726	2462,082	2515,078	2523,396
Исходные	2418	2457	2478	2409	2478	-
Процесс	2489,236	2509,046	2522,387	2406,723	2575,255	-

Источник: составлено автором.

В результате применения описанного метода получилось восстановить параметры механизма образования событий, который был задан алгоритмической моделью. Программная реализация на языке R опубликована в работе [213].

3.3 Экстраполяция восстановленных параметров

Восстановление параметров механизма образования событий, заданного алгоритмической моделью, с помощью описанного алгоритмического подхода – это лишь полдела. Для успешного прогноза надо ещё правильно произвести экстраполяцию значений параметров на будущее. Во второй главе уже обсуждался этот вопрос. Стационарные параметры со временем не меняются. Динамические параметры здесь также заданы

кубическими сплайнами, просто в другой параметризации (через начальные значения и значения третьей производной в узлах сплайна). По-хорошему, к вопросу экстраполяции динамических параметров надо привлекать экспертов, владеющих информацией о том, как меняются глобальные тенденции. Иначе нужно опираться на полученные результаты о динамике изменения в прошлом.

Так же как во второй главе, предполагается проводить экстраполяцию с помощью разложения на ограниченное число гармонических функций с помощью алгоритма Куинна и Фернандеса [158; 159]. Так же как и ранее, появляются два гиперпараметра n_1 и n_2 , задающих, сколько точек надо отбросить слева и справа у восстановленного сплайна. Напомним, что у нашего сплайна, даже в этот раз, вторая производная на концах обращается в ноль и сплайн ведёт себя как прямая линия за границами своей области определения. Отбросив n_1 и n_2 точек с начала и конца сплайна, раскладываем сплайн на сумму гармоник, после чего экстраполируем их на будущее. Определять эти гиперпараметры можно так же, как во второй главе, с помощью ещё одной оптимизации, подбирая гиперпараметры таким образом, чтобы прогноз нескольких последних событий был как можно ближе к фактическим последним событиям (как определяли $Score1$ и $Score2$ по формуле (2.62)).

На самом деле есть ещё один параметр α_j , это множитель, с которым учитывается штраф на меру кривизны восстанавливаемого динамического параметра j . По-хорошему, его тоже надо определять. Но это потребует перезапуск всей описанной выше процедуры поиска параметров алгоритмическими методами, что может быть крайне трудоёмким (в математическом решении основная задача сводилась к матричным операциям, которые не требовали много времени). Поэтому параметр α_j рекомендуется фиксировать на определённом уровне, способствующем достижению заданной степени кривизны функции. В итоге предлагается

выполнить один раз восстановление параметров с заданным значением α_j , а затем на основе полученной функции динамического параметра определять n_1 и n_2 , что не требует большого объёма вычислений.

Заметим, что средняя ошибка прогноза $(Score1 + Score2)/2$ (или среднее квадратичное), полученная на основе расчёта ошибки прогноза последних двух или большего числа событий, может быть использована как индикатор того, насколько выбранная модель механизма образования событий подходит для получения уже будущего события. Если эта ошибка мала, то можно доверять модели, а если эта ошибка большая, то, возможно, стоит попробовать другую модель.

3.4 Методы автоматического подбора структуры алгоритмических моделей механизмов образования событий

Если требуется подобрать не только параметры модели P , но и саму модель механизма образования событий $Program(X, P)$, то вместо решения оптимизационной задачи $S(P) \rightarrow \min$ необходимо решать оптимизационную задачу

$$S(Program, P) \rightarrow \min, \quad (3.10)$$

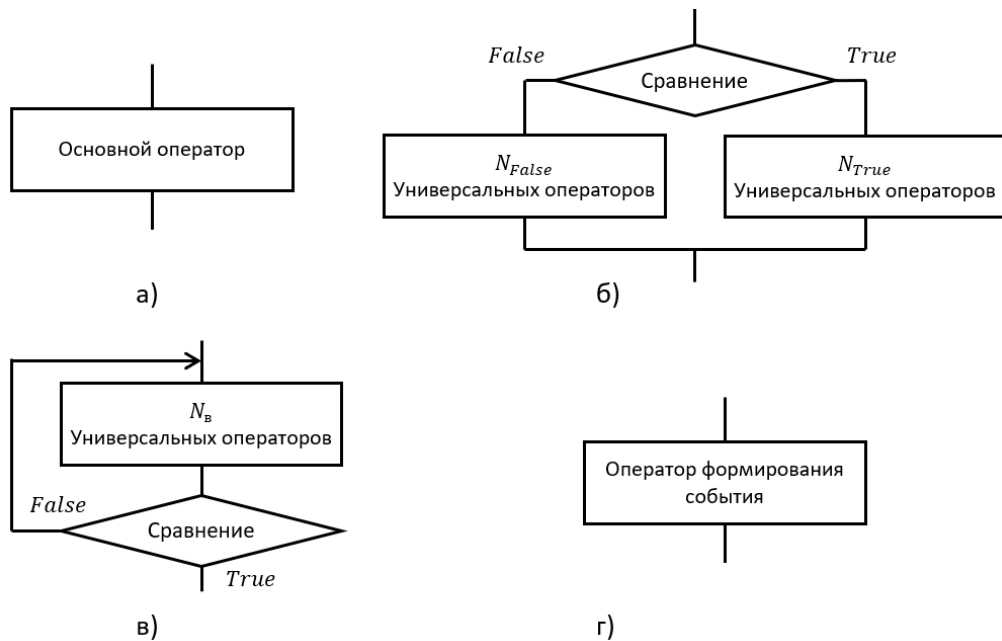
где $Program$ – составленная из операторов модель механизма образования событий. Можно предложить несколько подходов в решении этой задачи.

Комбинаторный подход

В этом подходе предполагается перебирать все возможные конструкции из операторов для составления модели механизма образования событий $Program$ при заданном ограничении на количество операторов, количество внутренних переменных и параметров. Для каждой модели $Program$ происходит попытка восстановления параметров P по имеющейся выборке событий с использованием критерия (3.9). Для большинства неправильно составленных моделей эта попытка завершится на первом же шаге, так как модель не будет способна образовать нужное количество

событий. Но среди моделей, которые способны образовывать нужное количество событий, предстоит выбрать наилучшую.

Модель *Program* предлагается конструировать рекурсивным образом из операторов, которых назовём универсальными. Каждый универсальный оператор может быть одного из четырёх видов: основной оператор, оператор ветвления, оператор цикла и оператор формирования события. Внутри оператора ветвления и цикла могут располагаться другие универсальные операторы, которые, в свою очередь, опять могут быть одного из четырёх видов, рисунок 3.6. Таким образом может быть получена достаточно сложная вложенная рекурсивная структура операторов.



а) основной оператор; б) оператор ветвления; в) оператор цикла; г) оператор формирования события.

Источник: составлено автором.

Рисунок 3.6 – Четыре вида универсального оператора

Основной оператор предназначен лишь для выполнения одной из арифметических операций между парой операндов с обязательным сохранением результата во внутреннюю переменную. Операндами могут быть внутренние переменные и параметры. В операторах ветвления и цикла обязательно осуществляется операция сравнения двух операндов. Внутри оператора ветвления может быть N_{true} и N_{false} , а внутри оператора цикла N_b универсальных операторов. Оператор формирования события, как и основной

оператор, осуществляет одну из арифметических операций над операндами, но вместо сохранения результата во внутреннюю переменную формирует новое событие (t'_i, y'_i) , значением y'_i которого является полученный результат. Введём ограничение, что оператор формирования события может находиться только внутри оператора ветвления.

Ограничением на количество операторов для модели *Program* является максимальное количество основных операторов N_{opers} и количество сравнений N_{if} (количество операторов ветвления и циклов). Рекурсивным образом перебираются все возможные конструкции модели из этих операторов. Например, при ограничении в $N_{opers} = 3$ и $N_{if} = 1$ возможно всего 30 различных схем. При $N_{opers} = 3$ и $N_{if} = 2$ – уже 450. При $N_{opers} = 4$ и $N_{if} = 1$ – всего 70 комбинаций схем, при $N_{opers} = 4$ и $N_{if} = 2$ – уже 1638 комбинаций схем. При $N_{opers} = 5$ и $N_{if} = 1$ – всего 140 комбинаций схем, при $N_{opers} = 5$ и $N_{if} = 2$ – уже 4761 комбинаций схем.

Для каждой фиксированной схемы перебираются комбинации арифметических операций и операндов. Ограничением здесь является количество внутренних переменных N_X и параметров N_P . Количество возможных комбинаций арифметических операций и операндов зависит от того, из каких операторов составлена схема. Например, для схемы всего из $N_{opers} = 3$ основных операторов и $N_{if} = 1$ сравнения, если количество переменных $N_X = 2$ и количество параметров тоже $N_P = 2$, получается 10 892 880 комбинаций операторов/операндов. Но это количество всех возможных комбинаций, включая те, которые дублируют логику модели. То есть без отсеечения таких комбинаций, в которых происходит, например, перестановка операций сложения, перестановка операндов в умножении, запись в одну переменную и проверка условия с этой переменной или запись в другую переменную и проверка условия с другой переменной и т.д. Реальное количество действительно различных комбинаций (в том смысле, что логика модели становится другой) намного меньше. Но такая фильтрация

повторяющихся по смыслу моделей может быть непростой задачей и является темой дальнейших исследований. Дополнительно стоит сказать, что поиск наилучшей модели легко поддаётся распараллеливанию. Некорректно построенные модели не смогут дать выборки событий, поэтому подбор параметров для них закончится на первом шаге. Дополнительно можно сделать так, что если после первого шага восстановления параметров погрешность определения событий превосходит заданный порог, то можно не выполнять второй шаг и переходить к другой модели (например, если относительная погрешность больше 50%).

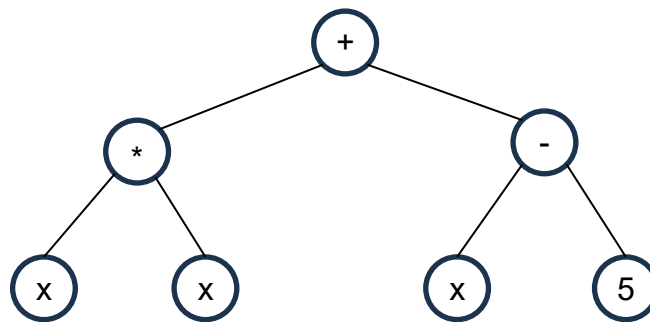
В работе [214] представлена реализация на языке R рекурсивного алгоритма автоматического перебора модели механизма образования событий *Program* (пока ещё без фильтрации повторяющихся моделей). В этой программной реализации иногда (при неудачно выбранных моделях процесса) за 5 минут получается исследовать ~ 730 комбинаций операторов (~ 2.5 комбинаций в секунду), но если модель процесса выбрана удачно, то весь поиск значений параметров этого процесса занимает около 10 секунд (в случае, если все параметры статические, если же параметры динамические, то время может быть намного больше).

Кроме комбинаторного подхода, есть и другой, правда, больше эвристический подход.

Генетическое программирование

Согласно работе [169], ещё в 1963 году предпринимались попытки научить машины программировать самих себя [170], хотя первые попытки научить компьютер писать компьютерные программы были ещё в 1958 году. Основополагающей считается работа Джона Р. Коза [171] 1992 года, в которой для этого направления исследований было закреплено название «генетическое программирование» и были изложены основные принципы подхода. Достаточно компактно и ясно основы генетического программирования излагаются в небольшой книге [172].

Основная идея генетического программирования совпадает с идеей генетических алгоритмов, когда в результате операций скрещивания и случайных мутаций получается более подходящий экземпляр компьютерной программы. Сначала надо инициализировать начальную популяцию программ. Программы чаще всего представляются деревьями, например рисунок 3.7, где в терминальных вершинах располагаются входные переменные или константы (терминалы), а в узлах дерева располагаются разные функции (бывают линейные и графовые представления [169, с. 113–118]). Функции могут быть совершенно разными, не только простейшие арифметические, но и логические операторы, тригонометрические функции, операторы ветвления и цикла [172, с. 21], да и совсем произвольные (но которые уже сконструированы).



Источник: составлено автором по материалам [169; 172].
Рисунок 3.7 – Представление программы $x^2 + x - 5$ в виде дерева

Инициализация начальной популяции программ может происходить несколькими способами. Или методом «роста», когда в каждом узле случайно выбирается либо терминальное значение, либо функция, пока не будет превышена максимальная глубина дерева, или «полным» способом, когда до достижения максимальной глубины могут выбираться только функции, а на самом нижнем уровне только терминальные значения, или их комбинацией (называемой «ramped half-and-half»), когда для каждого из фиксированных размеров деревьев (до максимального) в половине случаев дерево строится по методу роста, а в другой половине случаев полным способом.

Прежде чем производить операцию скрещивания, необходимо отобрать потенциальных родителей. Для этого существуют разные подходы.

Например, отбирать случайно по вероятностям, пропорциональным обратной величине ошибки соответствия, которую даёт программа. Или сортировать программы по этой ошибке и заданное количество лучших назначать родителями. Существуют и другие подходы [169, с. 126–133]. Однако при оценке «качества» каждой программы следует использовать «защищённые» операторы, которые обрабатывали бы возможные вычислительные ошибки вроде деления на ноль, корня из отрицательной величины, логарифма нуля и т.д., возвращая заранее заданное значение [172, с. 22–23].

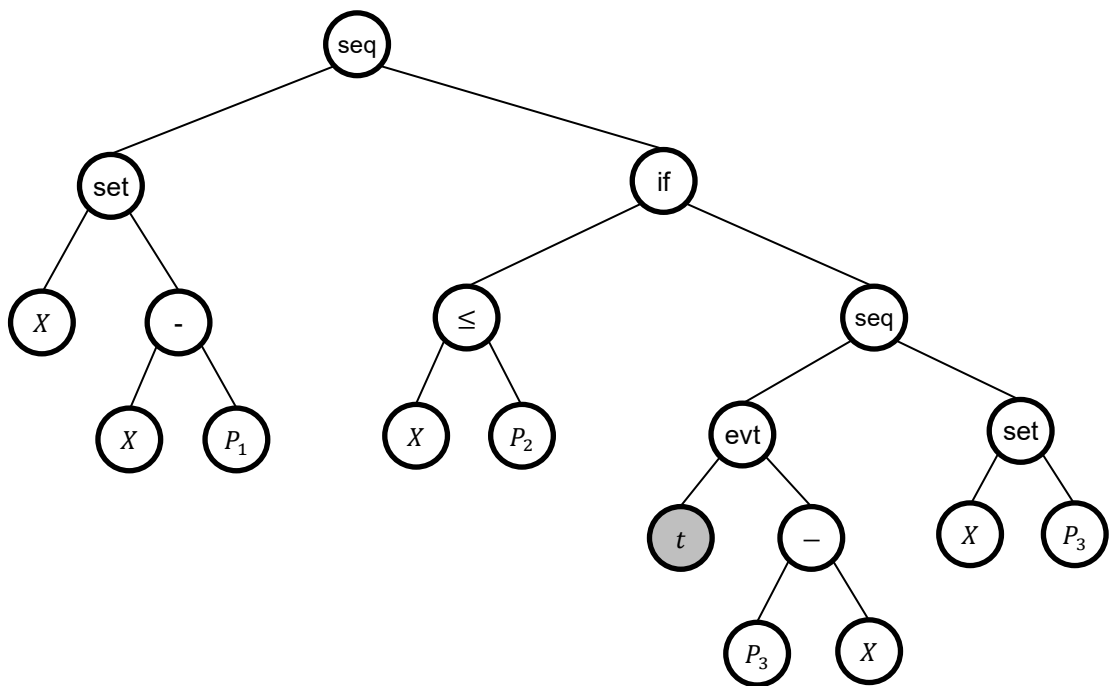
Операция скрещивания предполагает случайный выбор узла (точки пересечения) в деревьях обоих родителей, после чего их потомки обмениваются поддеревьями, начинающимися в этом узле. В итоге поддерево одного из родителей может оказаться совсем в другом месте в дереве потомка. Однако существует и «гомологичный» способ обмена поддеревьями, когда поддерево занимает то же место в дереве потомка [172, с. 45–46]. Операция скрещивания выполняется не для всех особей, а приблизительно для 90%. Для оставшихся 10% выполняется обычная репродукция особей [172, с. 30].

Наконец, в редких случаях (для 1% популяции) возможна мутация. Во время мутации выбирается случайное поддерево и заменяется на случайно сгенерированное поддерево, которое может быть другой глубины, или можно потребовать, чтобы поддерево было той же структуры, но с другими функциями и терминальными значениями [172, с. 42–43].

Условием остановки генетического алгоритма является либо достижение требуемой точности (погрешности), либо достижение максимального количества выполненных циклов эволюции. Наилучшая программа из популяции возвращается в качестве результата выполнения алгоритма.

Описанный алгоритм генетического программирования можно адаптировать для подбора алгоритмических моделей механизмов образования событий. Для этого необходимо в качестве терминалов использовать совокупность внутренних переменных $X = (X_1, \dots, X_{N_x})$ и параметров

$P = (P_1, \dots, P_{N_P})$, где N_X – количество внутренних переменных, N_P – количество параметров. Среди возможных функций обязательно должен быть оператор формирования события. Дерево программы для сквозного примера изображено на рисунке 3.8, где *set* – оператор присвоения, *seq* – оператор последовательности (по сути как оператор сложения, только не возвращает значения), *evt* – оператор формирования событий (t не является терминалом, если событие сформировано, оно всегда привязывается к текущему времени). Роль генетического программирования в том, чтобы в результате скрещивания, мутаций и отбора составилось такое дерево.



Источник: составлено автором.

Рисунок 3.8 – Модель механизма сквозного примера в виде дерева

Заметим: здесь изображено дерево для тела цикла программы, описанной в предыдущем разделе. Нет инициализации, оператора цикла (Пока $(t \leq t_{end})$) и оператора продвижения времени и обновления параметров. Для инициализации внутренних переменных (нулем или одним из параметров) можно обойтись без генетического программирования. Функционирование механизма во времени всегда программируется циклом с продвижением времени и обновлением динамических параметров после каждой итерации.

Поэтому для генетического программирования можно оставить только подбор тела цикла.

Значения параметров P будут определяться, как в предыдущем разделе, от генетического алгоритма требуется только построение самой схемы операторов. Качество построенной программы *Program* будем оценивать во время определения параметров P по формуле (3.8).

Однако у генетического программирования есть определённые недостатки, которые нужно иметь в виду. Во-первых, нет гарантии, что все возможные комбинации операторов, из которых может быть построена модель механизма, будут проверены. Генетическое программирование полагается на случайные числа, поэтому есть вероятность, что определённая комбинация операторов не разыграется при ограниченном числе циклов эволюции. Во-вторых, во время операции скрещивания могут образовываться повторные комбинации операторов, которые были уже получены и отброшены на предыдущих шагах. Тем самым частично алгоритм будет работать вхолостую. Быстродействие что генетических алгоритмов, что генетического программирования может быть далеко не самым лучшим. Тем не менее генетическое программирование продолжает активно развиваться.

В результате существует принципиальная возможность исследовать совершенно неизвестные события с помощью подбора алгоритмической модели механизма образования событий. При таком подходе исследователь не строит модель механизма, опираясь на некоторую известную информацию, а алгоритм сам подберёт наилучшую модель, после чего исследователь может заняться интерпретацией этой модели. Такой подход требует дальнейших исследований, необходимо подключение широкого круга исследователей.

Выводы по главе 3

1) Разработан метод восстановления параметров механизмов образования событий, заданных в виде произвольных алгоритмических моделей. Описаны особенности представления моделей механизмов.

Динамические параметры представляются в виде кубического сплайна, который кодируется через начальные условия в самом первом узле и значения третьей производной в последующих узлах. Параметры обновляются каждый шаг времени. Функция потерь состоит из суммы относительных отклонений моментов времени и значений прогнозируемых событий от фактических, плюс штраф на гладкость динамических параметров. Оптимизация функции потерь основывается на алгоритме Нелдера-Мида и поиска на сетке. Используется специальный приём, позволяющий снизить размерность получаемой задачи. Представлен пример метода восстановления параметров алгоритмической модели механизма образования событий.

2) Разработан комбинаторный способ подбора самих алгоритмических моделей механизмов образования событий. Модель составляется рекурсивно из универсальных операторов четырёх видов, которые, в свою очередь, опять могут содержать универсальные операторы. Задавая ограничение на количество операторов и сравнений, перебираются все возможные комбинации. Задавая ограничение на количество внутренних переменных и параметров, перебираются все возможные комбинации операндов в операторах.

3) Разработан способ подбора самих алгоритмических моделей механизмов образования событий на основе генетического программирования. Для адаптации генетического программирования к подбору моделей механизмов образования событий предлагается в качестве терминальных листов использовать внутренние переменные и параметры механизма. Среди возможных функций, используемых в узлах дерева, обязательно должен быть оператор формирования события.

Глава 4

Разработка информационных технологий для прогнозирования экономических событий

Помимо разработки методологии прогнозирования дискретных экономических событий, для активного распространения и внедрения результатов исследований, а также для становления и укрепления зарождающегося нового направления исследований, которое будем называть эвентометрикой, необходимо позаботиться о создании соответствующих технологий. Наиболее известными в настоящее время являются технологии машинного обучения и анализа данных, реализованные на языке Python в виде очень популярной библиотеки `scikit-learn` (`sklearn`) [173]. Сейчас в каждом учебном заведении страны практически на каждом направлении подготовки в том или ином виде преподаётся анализ данных в Python с использованием библиотеки `sklearn`. Создание библиотеки для прогнозирования экономических событий на распространённом языке Python, с похожим и привычным интерфейсом, как в `sklearn`, будет способствовать активному распространению новой методологии в научных кругах, внедрению в практическую деятельность на всех уровнях, быстрому обучению специалистов и служащих. Помимо этого, для активного распространения методологии среди компаний бизнеса будет полезным реализовать соответствующие технологии в качестве компонентов на платформе для бизнес-аналитики, такой как `Loginom` [174]. Платформа `Loginom` изучается студентами направления бизнес-информатики более чем в 118 учебных заведениях (на момент написания данного текста). Клиентами `Loginom Company` являются крупнейшие организации разных профилей, такие как МТС, t2, Норникель, X5 Retail Group, ПИК, ДОМ.РФ и др.

Примечание – Кроме библиотеки `scikit-learn` (`sklearn`) существуют и другие библиотеки с похожим интерфейсом, такие как `PyTorch`, `Keras`, `TensorFlow`. Но они больше посвящены глубоким нейронным сетям.

`Loginom` – отечественная аналитическая low-code (мало программирования) платформа, которая позволяет проводить анализ данных любого уровня сложности без программирования [180]. `Loginom`

позиционирует себя в основном как визуальное ETL (extract-transform-load) средство, но активно развивается и снабжается инструментами анализа и визуализации данных.

Ниже реализуется аналитический метод прогнозирования событий и автоматизированный способ подбора гиперпараметров, описанный в главе 2. Численный метод восстановления параметров механизма образования событий и автоматизированный комбинаторный способ подбора самих моделей реализован на языке R и опубликован в работах [213; 214].

4.1 Разработка библиотеки `eventometrics` для прогнозирования событий на языке Python

Название библиотеки задано английским словом `eventometrics`, которого ещё нет в обороте (что логично, так как направления эвентометрика ещё не существовало до этого). У данного направления видится потенциал, сравнимый с успехами состоявшейся эконометрики.

Для того, чтобы интерфейс новой библиотеки был похож на интерфейс библиотеки `sklearn`, основные функции будут также называться похожим образом: функция `fit` – для обучения модели, функция `predict` – для получения прогноза. Заметим: в методах регрессии или классификации на вход функции `predict` подаётся набор данных, например `model.predict(X)`, для которых надо получить прогноз в соответствии с обученной моделью. Но в нашем случае набор данных используется только для обучения модели, прогноз события осуществляется относительно самого последнего события в обучающей выборке. Тем самым никакие входные данные `X` для прогноза не задаются. Задаются только параметры, о чём будет ниже.

Обычно в библиотеке `sklearn` обучающий набор данных подаётся на вход только в момент вызова функции `fit`. Это кажется небольшим упущением. В нашей библиотеке реализовано так, что обучающий набор данных можно задавать как в момент создания модели (инициализации), так и в момент вызова функции `fit`. Если набор данных был задан при создании модели, то функция `fit` может вызываться без необходимости указания набора данных.

Если же набор данных задан, то он перезапишет тот, который был задан при инициализации.

Ещё одно важное замечание. В отличие от методов машинного обучения, которые используют связку `fit – predict`, в нашем новом подходе необходимо использовать связку `fit – extrapolate – predict`. Напомним, что в предлагаемом подходе восстанавливаются (обучаются) параметры модели механизма образования событий по имеющейся выборке событий, т.е. определяются значения параметров в прошлом. Для получения прогноза события нужно задать значения параметров для будущего периода времени. Таким образом, обязательно нужен этап экстраполяции параметров. Пока в самой первой версии библиотеки реализован способ экстраполяции, основанный на описанном выше разложении на ограниченное количество гармоник по алгоритму Куинна-Фернандеса.

Оптимизация вычислений. Сам чистый Python является достаточно медленным скриптовым языком. Для ускорения вычислений можно использовать разные средства. Одним из способов ускорения вычислений является использование Cython [174]. Cython – это статический компилятор, который позволяет заранее скомпилировать код и во много раз повысить скорость вычислений. Cython позволяет ускорить чистый код на Python, но он также позволяет писать оптимизированный код на подобном одноимённом языке Cython, который является некоторой смесью языка Python и C/C++. Компиляция чистого языка Python проигрывает в скорости, если бы код писался на языке Cython. Что чистый Python код, что код на языке Cython транслируется в оптимизированный код C/C++ и компилируется в виде модулей расширения Python. Расширения выполняются со скоростью компилированного кода на языке C, при этом обеспечивается возможность непосредственного использования из кода на Python [175]. Функции, написанные на языке Cython, после компиляции работают до 150 раз быстрее, чем функции на чистом Python без предварительной компиляции. Подробнее в документации [176].

Однако нет необходимости всю библиотеку писать на языке Cython. Например, в библиотеке `sklearn` используется гибридный подход, когда все классы, их поля, инициализация, отдельные блоки кода на верхнем уровне пишутся на чистом коде Python, и только критические функции, требующие большого объёма вычислений, пишутся на языке Cython. Например, объявление класса, весь ввод данных, проверка на корректность, некоторые предварительные приготовления написаны на чистом Python, и только какие-нибудь методы оптимизации, применяющиеся в обучении модели, написаны на Cython. В нашей библиотеке также используется гибридный подход. Все объявления классов, атрибуты, методы написаны на чистом Python, но внутри методов происходит обращение к оптимизированным функциям. Такой гибридный подход обеспечивает высокую гибкость, позволяет вносить изменения в исходный код на Python, не затрагивая оптимизированные участки кода.

Дополнительной оптимизацией является проверка и пропуск повторных вычислений одного и того же блока кода. Дело в том, что в процессе обучения, особенно в процессе подбора гиперпараметров, функция обучения `fit` будет многократно вызываться. Часть операций, например перерасчёт большинства матриц, не нужно осуществлять, если поменялся лишь какой-нибудь гиперпараметр, не связанный с этими матрицами.

Описание библиотеки

Первый модуль имеет имя файла `CapacityModels.py` и содержит модели, основанные на том механизме, когда события образуются вследствие исчерпания некоторого запаса. Внутри этого модуля объявлен первый класс, который называется `SimpleCapacity`.

```
class eventometrics.CapacityModels.SimpleCapacity(t=None, Y=None,
weights=None, knots_number=None, alpha=10**5, knots=None, x=None,
x_step=1, x_num=None, ALL_Positive=True, Positive_abs_tol=0.0001,
Positive_rel_tol=1e-6, max_recalculate_positive_iterations=4,
extrapolator=None, skip_input_parameters_validation=False)
```

Параметры конструктора:

t, *Y*, *weights* – входная обучающая выборка событий и их весов, заданных в виде одномерных вещественных (*double*) массивов (поддержка таблиц *pandas* будет реализована позже), по умолчанию не задана (*None*). Предполагается, что время *t* указывается в днях.

knots_number – количество узлов сплайна, если не задано, то определяется на основе данных в момент подготовки матриц, соответствует количеству наблюдений. Если задано *knots*, то *knots_number* игнорируется.

alpha – коэффициент сглаживания, по умолчанию $\alpha=10^5$.

knots – одномерный массив узлов сплайна, по умолчанию не задан.

x – массив координат, в которых рассчитывается сплайн, по умолчанию не задан, если не задан, то определяется как *x_num* равноудалённых точек или как точки с шагом *x_step* между самым первым и последним узлом сплайна.

x_step – шаг, с которым берутся точки для расчёта в них значений сплайна, по умолчанию равен 1 (предполагается, что время событий указано в днях).

All_Positive – указывает, следует ли восстанавливать сплайн как положительный, по умолчанию *True*.

Positive_abs_tol, *Positive_rel_tol* – абсолютная и относительная (к максимальному значению) точность достижения положительности сплайна, по умолчанию 10^{-4} и 10^{-6} .

max_recalculate_positive_iterations – ограничение на количество итераций при обеспечении положительности сплайна, по умолчанию 4.

extrapolator – экстраполятор, т.е. объект, осуществляющий экстраполяцию динамических параметров (скорости потребления), будет описан ниже, по умолчанию не задан.

skip_input_parameters_validation – указывает, можно ли пропустить валидацию входного набора данных, чтобы сэкономить немного времени. Используется в основном в служебных целях.

Атрибуты объекта:

h , Q , R , inv_R , t_Q , $inv_R_dot_QT$, K , W , V , P , C , t_C , $t_C_dot_W_dot_C$, c , D , g , $gamma$, $gamma2$ – массивы и матрицы, включая промежуточные: h , Q , R , R^{-1} , Q^T , $R^{-1}Q^T$, K , W , V , P , C , C^T , C^TWC , c , D , g , γ , описанные выше, $\gamma2$ – тот же вектор γ , дополненный нулями слева и справа, чтобы выровнять размерность. Определяются только после обучения модели.

Consumption_rate – восстановленная скорость расхода ресурса, одномерный массив, рассчитанный в точках x .

max_Storage – восстановленный максимальный размер запаса.

x_fut, **y_fut** – экстраполированные значения x и **Consumption_rate**.

Координаты в **x_fut** идут с шагом $x[1]-x[0]$.

y_last – значение экстраполированной функции в последнем узле (до начала будущего периода).

Методы объекта:

validate_input(t, Y, weights=None) – проверяет корректность входных данных и устанавливает соответствующие внутренние переменные (на данный момент реализована только проверка и конвертация в numpy массивы).

prepare_matrix() – подготавливает все массивы и матрицы, необходимые для начала обучения, вычисления подробно описаны во второй главе.

calculate_positive_spline_params() – рассчитывает параметры сплайна g , $gamma$, $gamma2$ при условии неотрицательности в узлах сплайна, задействуя оптимизированную с помощью cython функцию реализации алгоритма Лемке Lemke_cython. Подробное описание представлено во второй главе. Для ознакомления с реализацией можно посмотреть исходный код, который скачивается при загрузке библиотеки.

Про метод **fit** поговорим подробнее.

```
fit(t=None, Y=None, weights=None, alpha=None, All_Positive=None,
Positive_abs_tol=None, Positive_rel_tol=None,
max_recalculate_positive_iterations=None, **kwargs)
```

Параметры:

t, **Y**, **weights** – входные данные, которые могут быть пропущены, если они были заданы при инициализации самого объекта класса. Если при создании объекта входные данные также не задавались, то их следует задать в этой функции **fit**.

alpha – коэффициент сглаживания, который, если задан, перезаписывает значение, заданное при инициализации объекта.

All_Positive – указывает, следует ли восстанавливать сплайн как положительный, если задан, то перезаписывает соответствующее значение.

Positive_abs_tol, **Positive_rel_tol** – абсолютная и относительная (к максимальному значению) точность достижения положительности сплайна, если заданы, то перезаписывают соответствующие значения.

max_recalculate_positive_iterations – ограничение на количество итераций при обеспечении положительности сплайна, если задано, то перезаписывает соответствующее значение.

kwargs – другие ключевые аргументы, которые могут подставляться в функцию (на данном моменте игнорируются).

Рассмотрим код функции **fit**.

```
if t is not None and Y is not None:
    self.validate_input(t, Y, weights)
    self.have_data = True
    self.matrix_ready = False
if not self.matrix_ready:
    self.prepare_matrix()
    self.matrix_ready = True
    self.matrix_D_ready = False
if alpha is not None and alpha != self.alpha:
    assert alpha >= 0, "alpha should be not negative"
    self.alpha = alpha
    self.matrix_D_ready = False
if not self.matrix_D_ready:
    self.D = self.t_C_dot_W_dot_C + self.alpha * self.K
    self.matrix_D_ready = True
    self.spline_params_ready = False
if All_Positive is not None and All_Positive != self.All_Positive:
    self.All_Positive = All_Positive
    self.spline_params_ready = False
if Positive_abs_tol is not None and self.Positive_abs_tol != Positive_abs_tol:
    self.Positive_abs_tol = Positive_abs_tol
    self.spline_params_ready = False
```

```

if Positive_rel_tol is not None and self.Positive_rel_tol != Positive_rel_tol:
    self.Positive_rel_tol = Positive_rel_tol
    self.spline_params_ready = False
if max_recalculate_positive_iterations is not None
and self.max_recalculate_positive_iterations != max_recalculate_positive_iterations:
    self.max_recalculate_positive_iterations = max_recalculate_positive_iterations
    self.spline_params_ready = False
if not self.spline_params_ready:
    if self.All_Positive:
        self.calculate_positive_spline_params()
        self.spline_params_ready = True
    else:
        self.g = np.linalg.solve(self.D, self.c)
        self.gamma = self.inv_R_dot_QT @ self.g
        self.gamma2 = np.ascontiguousarray(np.append(np.append([0], self.gamma), 0))
        self.spline_params_ready = True
        self.Events_mechanism_parameters_ready = False
if not self.Events_mechanism_parameters_ready:

    # calculate Resource consumption rate
    self.Consumption_rate = eval_spline_cython(self.x, self.knots, self.h,
                                                self.g, self.gamma2)

    self.parameters_extrapolated = False

    # calculate max Storage
    self.max_Storage = np.sum(self.Y)
    y_ti = eval_spline_cython(self.t, self.knots, self.h, self.g, self.gamma2)
    self.max_Storage -= np.sum(y_ti)/2
    self.max_Storage /= self.n
    self.Events_mechanism_parameters_ready = True

```

В первом блоке, если задаются данные t или Y , то также делается пометка, что матрицы не готовы и их надо пересчитать. Далее, если матрицы не готовы, то происходит подготовка матриц и делается соответствующая пометка, при этом отмечается, что матрица D не готова. Когда второй раз будет вызвана функция `fit` без изменения входных данных, этап перерасчёта матриц будет пропущен. Если задавался параметр сглаживания α и он отличен от ранее заданного, то старое значение перезаписывается и делается пометка, что матрицу D также надо пересчитать (матрица D зависит от α). Далее, если матрица D была не готова, то она рассчитывается и ставится пометка, что надо пересчитать параметры сплайна g и γ . Затем идёт 4 блока проверки изменения параметров восстановления положительного сплайна, если хотя бы одна выполняется, то параметры сплайна также нужно будет пересчитать. Далее, если параметры сплайна не были ранее рассчитаны, то параметры сплайна рассчитываются либо с помощью вызова функции `calculate_positive_spline_params`, либо с помощью простых матричных операций, после чего делается пометка, что параметры сплайна готовы.

Наконец происходит расчёт скорости расхода ресурса как значений сплайна во всех точках x с помощью оптимизированной функции `eval_spline_cython`, а также расчёт максимального запаса простейшим способом, делается пометка о том, что параметры механизма образования событий готовы. Повторный запуск функции `fit` будет пропускать те блоки кода, аргументы которых не поменялись и пересчитывать которые не нужно. Это даёт значительное ускорение повторного вызова функции, который может быть нужен при подборе гиперпараметров, о чём подробно будет сказано ниже.

Следующим идёт метод `extrapolate_parameters`.

`extrapolate_parameters(force=False, extrapolator=None, **kwargs)`

Параметры:

force – указывает, следует ли принудительно выполнять расчёты, даже если параметры не изменились.

extrapolator – экстраполятор, т.е. объект, осуществляющий экстраполяцию параметров, можно не задавать, подробнее ниже.

kwargs – другие ключевые аргументы, которые будут подставляться при вызове функции заданного экстраполятора.

Код данной функции достаточно компактный, поэтому приведём её.

```
if force or not self.parameters_extrapolated or
    self.extrapolator.check_need_re_extrapolate(**kwargs):
    assert self.Events_mechanism_parameters_ready, "Parameters not ready to be
                                                    extrapolated. Call fit method first"
    if extrapolator is not None:
        self.extrapolator = extrapolator
    if self.extrapolator is None:
        self.extrapolator = QuinnFernandesExtrapolator()
    self.x_fut, self.y_fut, self.y_last = self.extrapolator.extrapolate(
        x=self.x,
        y=self.Consumption_rate,
        x_step=self.x_step, **kwargs)
    self.parameters_extrapolated = True
```

В этой функции проверяется, надо ли выполнять экстраполяцию. Для этого проверяется, не выполнена ли экстраполяция ранее или поменялись ли аргументы. Последнее проверяется с помощью функции самого объекта класса экстраполятора `check_need_re_extrapolate`. Класс описан в другом модуле и будет обсуждаться ниже. Если ни при инициализации и ни при

вызове этой функции экстраполятор так и не был задан, то в качестве экстраполятора выбирается экстраполяция с помощью разложения на ограниченное количество гармоник с помощью алгоритма Куинна-Фернандеса. Наконец запускается метод `extrapolate` объекта экстраполятора, который возвращает значения x и y (не путать с Y) на будущем периоде, а также значение экстраполированной функции в последнем узле (до начала будущего периода).

Наконец метод `predict`.

```
predict(n_predict=None, n_events=1, **kwargs)
```

Параметры:

n_predict – максимальное количество единиц времени работы механизма образования событий, если не задано, то берётся из свойств экстраполятора, а если тот не задан, то устанавливается равным 200.

n_events – количество событий, которые пытается предсказать модель механизма образования событий, по умолчанию 1.

kwargs – другие ключевые аргументы, которые будут подставляться в функцию экстраполяции параметров, если та ещё не вызывалась.

Код функции следующий.

```
def predict(self, n_predict=None, n_events=1, **kwargs):
    if n_predict is None:
        if self.extrapolator is not None:
            n_predict = self.extrapolator.n_predict
        else:
            n_predict = 200
    if not self.parameters_extrapolated:
        self.extrapolate_parameters(n_predict=n_predict, **kwargs)
    elif len(self.x_fut) != n_predict:
        warnings.warn('Because n_predict is different from length(x_fut) calculate
            extrapolation again')
        self.extrapolate_parameters(n_predict=n_predict, **kwargs)
    if n_events > 1:
        t_predict = []
        Y_predict = []
    Storage = self.Y[-1]
    Storage -= self.y_last/2 # Correction
    num_events = 0
    i = 0
    while i < n_predict:
        Storage -= self.y_fut[i]
        if Storage < 0:
            if n_events > 1:
                t_predict.append(self.x_fut[i])
                Y_predict.append(self.max_Storage - Storage)
```

```

        Storage = self.max_Storage
        num_events += 1
        if num_events >= n_events:
            return t_predict, Y_predict
        else:
            return self.x_fut[i], self.max_Storage - Storage
    i += 1
if num_events > 0:
    return t_predict, Y_predict
else:
    return None, None

```

В самом начале определяется величина `n_predict`, если она не была задана. Затем, если экстраполяция параметров ещё не проводилась или если количество единиц времени, на которое ранее осуществлялась экстраполяция, отличается от `n_predict`, то заново вызывается функция экстраполяции параметров. Наконец, моделируется сам механизм образования событий с установленными значениями параметров, в результате которого получается прогноз дискретных событий. Заметим, что механизм образования событий может завершиться, так и не сформировав нужного количества событий.

Вторым модулем является файл с именем **Extrapolators.py**, который содержит описание классов для реализации разных способов экстраполяции параметров. Внутри этого модуля описан класс `QuinnFernandesExtrapolator`.

```

class eventometrics.Extrapolators.QuinnFernandesExtrapolator(n1=0,
n2=0, n_predict=None, Nharm=10, FreqTOL=1e-6, MaxIterations=10000)

```

Параметры:

n1, n2 – количество отбрасываемых точек слева и справа поданного для экстраполяции временного ряда, по умолчанию ноль.

n_predict – количество единиц времени, на которое необходимо экстраполировать значения поданного временного ряда, по умолчанию не задано.

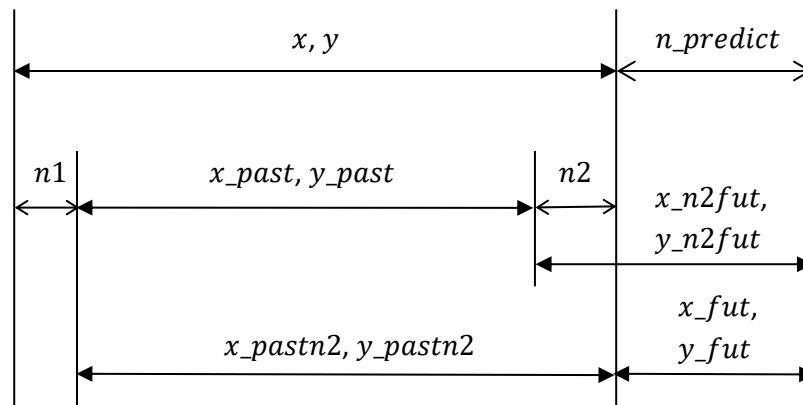
Nharm – количество гармоник, на которые раскладывается поданный временной ряд, по умолчанию 10.

FreqTOL – точность вычисления частот, по умолчанию 10^{-6} .

MaxIterations – максимальное количество итераций при расчёте частот, по умолчанию 10 000.

Атрибуты:

x_past, y_past, x_n2fut, y_n2fut, x_pastn2, y_pastn2, x_fut, y_fut – одномерные массивы вещественных чисел, получающиеся после отбрасывания $n1$ и $n2$ точек, а также экстраполяции входных значений x и y , соответствующие разным интервалам, согласно рисунку 4.1. Данные массивы удобно использовать при построении графиков.



Источник: составлено автором.

Рисунок 4.1 – Соотношение подготавливаемых массивов с входными массивами x и y

Методов класса всего два:

check_need_re_extrapolate($n1=None$, $n2=None$, $n_predict=None$, $Nharm=None$, $FreqTOL=None$, $MaxIterations=None$, ****kwargs**) – возвращает True, если хотя бы один из аргументов задан и отличается от ранее используемых, что свидетельствует о том, что надо заново проводить экстраполяцию.

extrapolate(x , y , $n1=None$, $n2=None$, $n_predict=None$, $x_step=None$, $Nharm=None$, $FreqTOL=None$, $MaxIterations=None$, ****kwargs**) – производит экстраполяцию временных рядов x и y на $n_predict$ единиц времени вперёд с помощью алгоритма Куинна-Фернандеса. Параметры те же, что и при объявлении класса. Все параметры кроме x и y могут быть не указаны, в этом случае используются значения, заданные при инициализации объекта. Внутри задействуется оптимизированная функция `Quinn_Fernandes_cython`, куда передаются все нужные параметры. В результате вызова подготавливаются

соответствующие массивы, как на рисунке 5.1. Метод возвращает x_{fut} , y_{fut} и последнее значение массива y_{pastn2} . Подробности реализации смотрите в исходных кодах библиотеки.

Примеры использования

Прежде всего надо установить библиотеку. Библиотека уже загружена на официальный репозиторий PyPI, поэтому достаточно выполнить команду (в консоли самого Python)

```
pip install eventometrics.
```

Далее подготавливаем входные данные. Пусть в файле с данными `simple.csv` имеется два столбца с заголовками `t` и `y` (как в таблице 2.1). Читаем данные из файла следующим образом.

```
import numpy as np
import pandas as pd

filename = "../eventometrics/data/simple.csv"
discard_events = 3 # заменить 3 на 2 или 1

MyData = pd.read_csv(filename, sep = ";", decimal=',')
print(MyData)

t = pd.to_datetime(MyData.t.dropna(), format='%d.%m.%Y')
t_start = min(t)
t = np.array([(x-t_start).days for x in t], dtype = np.double)
Y = MyData.y.dropna().to_numpy(dtype=np.float64)
n = len(t)

# ===== Убрать discard_events последних событий =====
print("Last event = ", (t_start+pd.Timedelta(t[n - discard_events - 1], "d")).date())
print("Next event = ", (t_start + pd.Timedelta(t[n - discard_events], "d")).date() )
T_future_real = t[n - discard_events]
T_future_real_date = (t_start + pd.Timedelta(T_future_real, "d")).date()
Y_future_real = Y[n - discard_events]

t = t[0:(n - discard_events)]
Y = Y[0:(n - discard_events)]
n = len(t)
```

В этом фрагменте кода из файла `simple.csv` с помощью библиотеки `Pandas` (которую тоже надо установить) считывается набор данных, содержащий столбцы `t` и `y`. Далее время событий `t` преобразуется из формата дат в обычные вещественные значения, как разница в днях относительно даты самого первого события. Переменные `t` и `Y` преобразуются в `numpy` массивы. Затем отбрасывается какое-то количество последних событий, чтобы их потом предсказывать. На экран выводится вся выборка событий, информация о

последнем событии и об ожидающемся истинном событии, которое надо предсказать, рисунок 4.2.

	t	y
0	30.01.2020	3628.994024
1	08.04.2020	3615.590775
2	30.08.2020	3625.529960
	• • •	
13	28.08.2023	3620.066928
14	20.01.2024	3633.927626
15	29.03.2024	3605.976363
16	15.08.2024	3624.141314
	Last event =	2023-08-28
	Next event =	2024-01-20

Источник: составлено автором.

Рисунок 4.2 – Результат после чтения данных

Далее создаём модель, например, таким образом.

```
from eventometrics.CapacityModels import SimpleCapacity
from eventometrics.Extrapolators import QuinnFernandesExtrapolator

model = SimpleCapacity(t, Y,
                       knots_number=n*3,
                       alpha=10**5,
                       max_recalculate_positive_iterations=1,
                       extrapolator=QuinnFernandesExtrapolator(n1=0, n2=0, Nharm=7)
                       )
```

Здесь данные t и Y задаются при создании модели, но это было необязательно, так как их можно было задать позже. Также в качестве экстраполятора создаётся объект `QuinnFernandesExtrapolator` с некоторыми значениями параметров $n1$, $n2$ и $Nharm$.

Само обучение можно выполнить функцией `fit` без аргументов.

```
model.fit()
```

После обучения можно вывести на экран восстановленные значения параметров механизма образования событий, рисунок 4.3.

```
>>> model.Consumption_rate
array([[73.31534038, 72.70089436, 72.08647533, ..., 58.66963572,
        58.93155278, 59.19347415], shape=(1307,))
>>> model.max_Storage
np.float64(3597.777822528977)
```

Источник: составлено автором.

Рисунок 4.3 – Вывод на экран восстановленных параметров механизма образования событий

Для получения прогноза указываем горизонт планирования и сколько событий нужно предсказать. Заметим, что экстраполяция параметров запускается в данном случае самой функцией `predict`.

```
t_pred, y_pred = model.predict(n_predict=200, n_events=1)
```

Результатом прогноза будут два вещественных числа, количество дней относительно первого события и предполагаемый объём покупки. Чтобы в более читаемом формате посмотреть на результат и сравнить его с ожидаемым значением, выполняем строки кода.

```
t_pred_date = (t_start + pd.Timedelta(round(t_pred), "d")).date()
print("t_pred_date = ", t_pred_date, " y_pred = ", y_pred)
print("T_real_date = ", T_future_real_date, " y_real = ", Y_future_real)
error = t_pred - T_future_real
dt = T_future_real - t[-1]
print("dt = ", dt)
rel_error = error / dt
print("error = ", error, " rel_error = ", rel_error)
```

Результатом будут следующие строки.

```
t_pred_date = 2024-01-04  y_pred = 3602.6799381853707
T_real_date = 2024-01-20  y_real = 3633.927626
dt = 145.0
error = -16.0  rel_error = -0.1103448275862069
```

В результате при истинном интервале между последним и будущим событием в 145 дней ошибка прогноза составила 16 дней или 11%.

Экстраполяцию параметров можно было выполнить явно.

```
model.extrapolate_parameters(n_predict=300, n1=10, n2=15, Nharm=8)
```

После этого опять выполнить прогнозирование и вывести результаты.

```
t_pred, y_pred = model.predict(n_events=3)
t_pred_dates = [(t_start + pd.Timedelta(round(t_p), "d")).date() for t_p in t_pred]
for date, y_p in zip(t_pred_dates, y_pred):
    print("t = ", date, " Y = ", y_p)
```

Результаты будут

```
t = 2024-01-08  Y = 3608.6123640419473
t = 2024-03-28  Y = 3615.118289359484
```

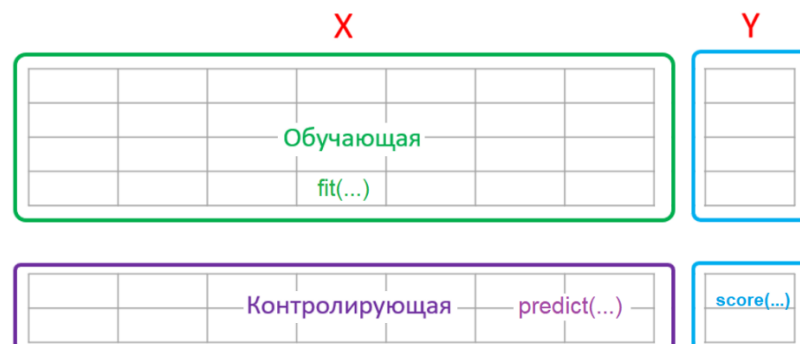
Заметим: мы просили дать прогноз трёх событий, а получили только два события, так как 300 дней было недостаточно для получения последнего события. Если задать параметр `n_predict` равным 500, то получаем и третье событие. Однако, чем дальше временная перспектива, тем больше может быть ошибка прогноза.

В этом простом примере все гиперпараметры были заданы вручную. Вообще прогноз может очень сильно зависеть от таких гиперпараметров, как α , $n1$, $n2$ (от других параметров он тоже может зависеть, но в меньшей степени). Подбор гиперпараметров вручную может быть изнурительным. Для автоматического подбора гиперпараметров в библиотеке была разработана специальная процедура.

Поиск оптимальных гиперпараметров

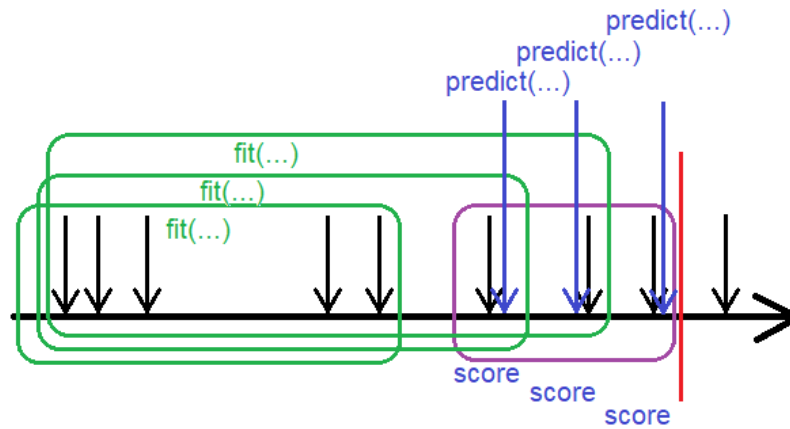
В библиотеке `sklearn` поиск гиперпараметров в основном осуществляется с помощью функции `GridSearchCV` [177], которая строит сетку комбинаций гиперпараметров и запускает обучение и предсказание из каждого узла этой сетки, оценивает качество предсказания и выбирает лучший результат. Поиск гиперпараметров в библиотеке `eventometrics` хоть и следует этому пути, но с очень серьёзными отличиями.

Во-первых, набор данных для валидации задействуется совсем иначе. Валидационный набор в библиотеке `sklearn` может выбираться случайно или последовательно из набора данных. У нас только последовательно. В `sklearn` модель оценивается по всей обучающейся выборке, прогноз получается для каждой записи контролирующей (валидационной) выборки, после чего прогноз сравнивается с истинными значениями, рисунок 4.4. У нас же модель обучается многократно: сперва по выборке без событий, попавших в валидационный набор, получается прогноз одного события, затем модель снова обучается, но по выборке, куда включается очередное событие, и так далее, рисунок 4.5.



Источник: составлено автором.

Рисунок 4.4 – Логика использования набора данных при валидации в `sklearn`



Источник: составлено автором.

Рисунок 4.5 – Логика использования набора данных при валидации в eventometrics

Давать прогноз сразу всех будущих событий на большую временную перспективу неправильно, это гарантированно приведёт к высоким ошибками. В машинном обучении в валидационном наборе могут находиться наблюдения, относящиеся к одному и тому же периоду времени. У нас же каждое очередное событие соответствует следующему периоду времени. Наша задача – подобрать гиперпараметры, при которых обучение, а также экстраполяция позволяют впоследствии хорошо предсказывать будущие события.

Другой отличительной особенностью является использование локальной оптимизации из каждой середины ячейки сетки. В sklearn эффективность гиперпараметров оценивается только в узлах сетки, которую задает пользователь. Промежуточные значения между узлами сетки никак не проверяются. В библиотеке eventometrics запускается локальный поиск, основанный на специальным образом модифицированном алгоритме Нелдера-Мида из середины (не из узла) каждой ячейки сетки. Тем самым проверяются многие числовые значения гиперпараметров.

Про локальный поиск надо рассказать отдельно. От классического алгоритма Нелдера-Мида его отличает то, что он умеет работать с целочисленными переменными, при этом корректно осуществляет проверку условий сходимости. Заметим, что категориальные переменные не изменяются при локальной оптимизации, значения категориальных

переменных определяются только сеткой параметров. В локальном поиске предусмотрена остановка при выходе за границы ячейки. Причём для возможного случая, когда оптимум может лежать рядом с границей ячейки, используется перекрытие нескольких соседних ячеек, чтобы залезание за границу не приводило к немедленной остановке и была возможность вернуться обратно. Если же ячейка сетки была на краю, т.е. некоторые границы ячейки совпадают с некоторыми глобальными границами всей сетки, то в случае, когда оптимум лежит на грани или за гранью глобальной границы, возможно сокращение размерности симплекса, чтобы оптимум продолжал искаться только на этой грани. Кроме этого, используются дополнительные модификации классического алгоритма Нелдера-Мида. Одна из них – это отражение не только самой худшей точки от центра противоположной грани симплекса, но и второй самой худшей (предпоследней) точки от противоположной ей грани. Другая модификация, центр грани, противоположный самой лучшей точке, отражается относительно центра грани, противоположной второй худшей точке, и туда помещается худшая точка. Ещё одна модификация связана с тем, что запрещается на несколько итераций проверка операции простого отражения и растяжения после операций внутреннего отражения и глобального сжатия. По проведённым тестам первая и вторая модификация немного уменьшают вероятность застрять в неоптимальной точке. Последняя модификация немного сокращает общее количество вызовов оптимизируемой функции по сравнению с классическим методом Нелдера-Мида (включая проверку дополнительных отражений). Подробнее ознакомиться с реализацией локального поиска можно, обратившись к исходным кодам библиотеки, доступной по адресу <https://pypi.org/project/eventometrics/#files>.

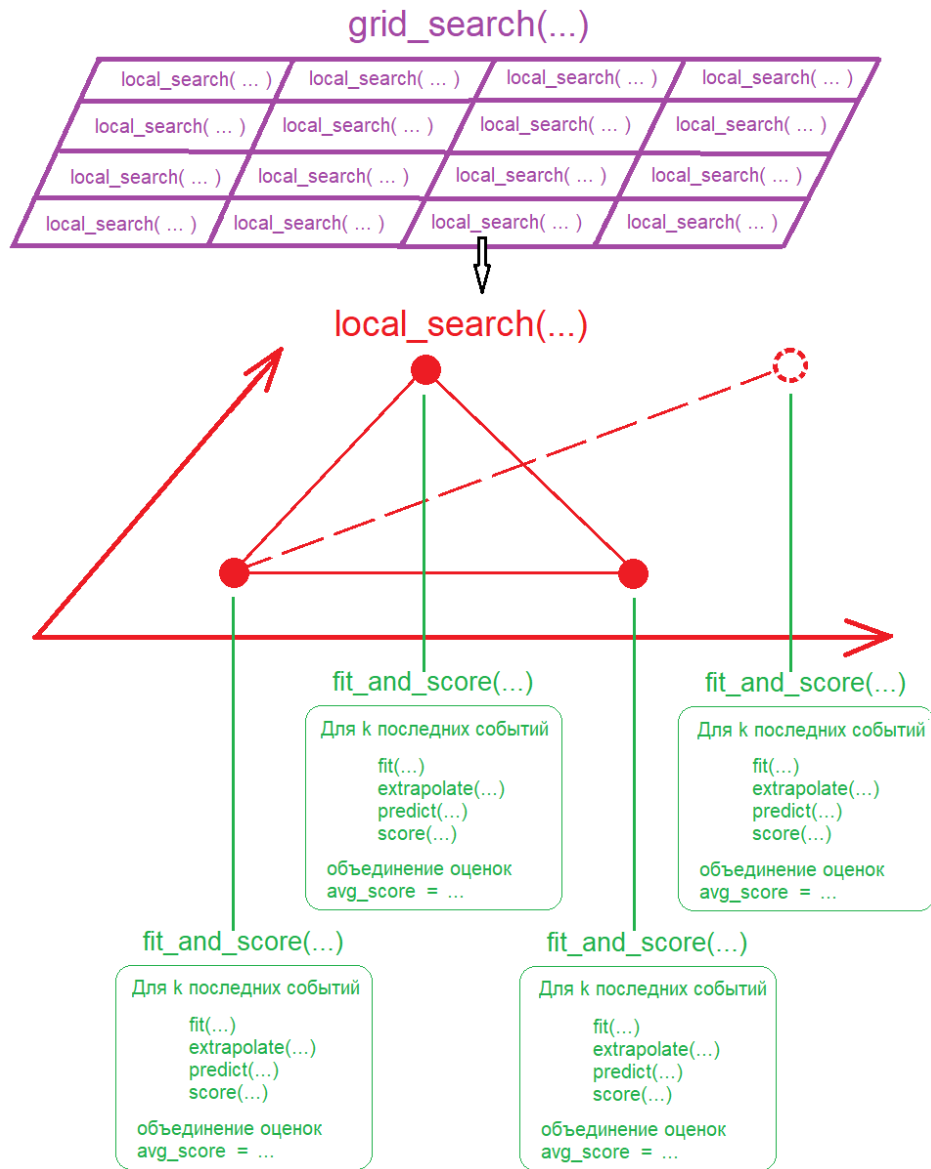
Реализация эффективного поиска оптимальных гиперпараметров была непростой задачей. Причины в том, что многие циклы оказываются вложенными друг в друга, рисунок 4.6. Самых ячеек сетки множество. Локальный поиск может требовать большое число итераций. Для оценки

качества одной комбинации параметров требуется несколько раз обучить модель, экстраполировать параметры, предсказать следующее событие, посчитать метрику качества предсказания, то есть функция `fit_and_score` выполняет действия, показанные на рисунке 4.5.

Для достижения высокой эффективности и быстродействия были предприняты следующие шаги. Во-первых, на самом верхнем уровне используются параллельные вычисления. Локальная оптимизация для каждой ячейки сетки запускается и выполняется в отдельных процессах параллельно. В самой локальной оптимизации были предприняты описанные выше шаги. Это остановка при выходе за границы ячейки, запрет на несколько шагов отражений и растяжений после внутренних отражений или сжатий, а также дополнительные приёмы, связанные с отражениями других точек. Функцию `fit_and_score` рассмотрим ниже подробно, в ней тоже применяются приёмы для ускорения вычислений. Также при обучении самой модели в функции `fit` применяются описанные выше приемы для пропуска повторных вычислений. Благодаря всему этому подбор гиперпараметров происходит достаточно быстро.

Ещё одним отличием от библиотеки `sklearn` являются сами метрики оценки качества прогноза. Машинное обучение в `sklearn` не предсказывает событий, а занимается классификацией, регрессией или кластеризацией. Метрики для оценки качества предсказания событий серьёзно отличаются. Во второй главе кратко о них уже говорили, ниже они будут приведены более подробно.

Поиск оптимальных гиперпараметров реализован в отдельном модуле `Search` с помощью класса `GridSearchCV`, обсудим его более подробно.



Источник: составлено автором.

Рисунок 4.6 – Структура вложенных циклов при поиске оптимальных гиперпараметров

Описание модуля Search

Внутри этого модуля объявлены несколько функций вне класса и сам класс GridSearchCV.

```
class eventometrics.Search.GridSearchCV(estimator, param_grid,
excluded_from_local_search=None, scorer=None, extra_scorer=None,
n_jobs=-2, refit=True, num_val_events=3, n_predict=200,
mix_weight=0.1, is_int=None, simplex_start_size=0.05, overlap=1,
verbose=1)
```

Параметры:

estimator – непосредственно сама модель, для которой подбираются гиперпараметры.

param_grid – сетка гиперпараметров, представленная в виде словаря, где ключи – это имена гиперпараметров, а значением является список возможных значений этого гиперпараметра.

excluded_from_local_search – список имён гиперпараметров, исключённых из локальной оптимизации, по умолчанию не задан. Нечисловые гиперпараметры автоматически исключаются из локального поиска, указывать их нет необходимости.

scorer – функция для расчёта качества прогноза событий, либо строка (одна из "mae", "mre", "rmse", "rmsre", "ae", "re", подробности ниже), либо вызываемая функция. По умолчанию не задана, что соответствует "mre".

extra_scorer – дополнительная функция оценки качества прогноза, либо строка (одна из "mean" или "rms"), либо вызываемая функция. Имеет эффект, только когда ***scorer*** "ae" или "re". По умолчанию не задана.

n_jobs – количество одновременно работающих задач, 1 означает однопоточное выполнение, -1 означает задействование всех доступных ядер процессора, -2 означает задействование всех доступных ядер процессора, кроме одного. По умолчанию -2.

refit – True или False, означает, нужно ли после поиска гиперпараметров обучать модель на всём наборе данных, по умолчанию True.

num_val_events – количество последних событий, на основе которых происходит поиск лучшей комбинации гиперпараметров, по умолчанию 3.

n_predict – количество единиц времени, за которое делается попытка получить каждое из предсказываемых событий, по умолчанию 200.

mix_weight – весовой коэффициент для оценки качества прогноза события, показывает, с каким весом учитывается ошибка прогноза Y , по умолчанию 0.1 (ошибка прогноза даты t всегда учитывается с весом 1, к этой ошибке прибавляется ошибка прогноза Y , умноженная на ***mix_weight***).

is_int – список имён целочисленных гиперпараметров, по умолчанию не задан.

simplex_start_size – начальный размер симплекса в алгоритме Нелдера-Мида, показывает ширину по каждой из осей относительно ширины ячейки сети, по умолчанию 0.05 (относительно стартовой точки строятся ещё n точек, где n – размерность исследуемого пространства, каждая точка получена приращением 0.05 от ширины ячейки по одной из осей).

overlap – перекрытие ячеек сетки, по умолчанию 1, при значении 0 нет перекрытия. Используется, чтобы находить оптимумы, расположенные близко к краям ячеек сетки.

verbose – контролирует полноту вывода информации на экран в процессе поиска гиперпараметров. На данный момент используются только значения 0 или 1. При значении 1 выводится вся информация, при значении 0 никакой информации на экран не выводится.

Атрибуты:

best_params – лучшая найденная комбинация параметров, представляет собой словарь, где в качестве ключей используются имена гиперпараметров.

best_value – наилучшее значение качества прогноза событий.

best_estimator – в случае, если *refit=True*, то обученная на полном обучающем наборе событий модель, иначе None.

Методы класса:

prepare_parameters_lists() – служебная функция, которая подготавливает и возвращает список параметров, списки границ и глобальных границ для всех ячеек сетки, которые используются во время распараллеливания задачи глобальной оптимизации.

fit(t=None, Y=None, weights=None, **kwargs) – основной метод, осуществляющий поиск оптимальных гиперпараметров. Параметры *t*, *Y*, *weights* можно не задавать, если они были заданы у модели, указанной как *estimator* во время инициализации. Внутри функции происходит вызов

функции `prepare_parameters_lists` для подготовки всех ячеек сетки. Затем запускается параллельное выполнение локального поиска из каждой ячейки, сортировка списка результатов оптимизации и выбор оптимальной комбинации гиперпараметров. Если свойство *refit* было установлено `True`, то обучает копию исходной модели на всём наборе данных с лучшими найденными гиперпараметрами. Функция реализована в духе аналогичной функции библиотеки `sklearn`, но совсем с другими подготовленными данными и целевой функцией для оптимизации. Функция достаточно объёмна, поэтому не будем приводить её код. Единственное замечание, что во время параллельного выполнения каждый процесс работает независимо со своей копией объекта модели. Ознакомиться с реализацией можно, посмотрев исходный код библиотеки.

Сама вызываемая для параллельного выполнения функция `start_local_search` объявлена в модуле `Search` вне класса. В этой функции происходит разделение гиперпараметров на те, которые надо оптимизировать и которые не надо оптимизировать во время локального поиска. После этого запускается локальная оптимизация только для оптимизируемых гиперпараметров. Реализация локального поиска с помощью алгоритма Нелдера-Мида вынесена в отдельный модуль `optimization`, она слишком объёмна, не будем её рассматривать. В качестве целевой функции для локального поиска указывается метод `fit_and_score`, который объявлен в модуле `CapacityModels` как метод самой модели `SimpleCapacity` (возможно, будет перенесён в другой модуль). Метод `fit_and_score` рассмотрим более подробно.

```
eventometrics.CapacityModels.fit_and_score(Parameters, scorer,  
extra_scorer=None, scalars=False, num_val_events=1, **kwargs)
```

Параметры:

Parameters – гиперпараметры, подбираемые во время локальной оптимизации, представленные в виде словаря как ключ-значение.

scorer – функция для оценки качества прогноза событий.

extra_scorer – дополнительная функция для оценки качества прогноза событий, может не задаваться.

scalars – свойство True или False, показывающее, нужно ли оценку качества прогноза событий производить индивидуально или же всем списком сразу. Зависит от того, какая функция ***scorer*** выбиралась при инициализации поиска на сетке (True, если выбиралась "ae", "re" или задавалась собственная функция). По умолчанию False.

num_val_events – количество событий, отводимых для валидации, по умолчанию 1.

kwargs – другие параметры, подставляемые в функции обучения, экстраполяции, прогнозирования или оценки событий.

Рассмотрим подробнее реализацию функции `fit_and_score`.

```

if not self.child_models_initialized:
    child_models = []
    for i in range(num_val_events):
        child_model = clone(self)
        new_n = self.n - num_val_events + i
        if self.weights is not None:
            ch_weights = self.weights[0:new_n-1]
        else:
            ch_weights = None
        child_model.validate_input(self.t[0:new_n], self.Y[0:new_n], ch_weights)
        child_models.append(child_model)
    self.child_models = child_models
    self.child_models_initialized = True

if scalars:
    Scores = np.zeros(num_val_events)
else:
    t_true_arr = self.t[self.n - num_val_events:]
    t_lag_true = self.t[self.n - num_val_events - 1]
    t_predict_arr = np.zeros(num_val_events)
    Y_true_arr = self.Y[self.n - num_val_events:]
    Y_predict_arr = []

for ind, ch_model in enumerate(self.child_models):
    ch_model.fit(**Parameters, **kwargs)
    ch_model.extrapolate_parameters(force=True, **Parameters, **kwargs)
    t_predict, Y_predict = ch_model.predict(**Parameters, **kwargs)

    if t_predict is None or Y_predict is None:
        return np.inf
    if scalars:
        ind_true = self.n - num_val_events + ind
        score = scorer(t_true=self.t[ind_true],
                      t_lag_true=ch_model.t[-1],
                      t_pred=t_predict,
                      Y_true=self.Y[ind_true],
                      Y_pred=Y_predict,
                      mix_weight=kwargs["mix_weight"]
                      )
        if score == np.nan or score == np.inf:
            return np.inf
        Scores[ind] = score

```

```

else:
    t_predict_arr[ind] = t_predict
    Y_predict_arr.append(Y_predict)
if scalars:
    score = extra_scorer(Scores, weights=kwargs["sample_weights"]) else:
    score = scorer(t_true=t_true_arr,
                  t_lag_true=t_lag_true,
                  t_pred=t_predict_arr,
                  Y_true=Y_true_arr,
                  Y_pred=Y_predict_arr,
                  **kwargs)
if score == np.nan:
    return np.inf
return score

```

Напомним, что метод `fit_and_score` вызывается многократно во время локальной оптимизации с разными значениями гиперпараметров. Также в этом методе прогнозируется несколько событий, каждый раз включая очередное событие в обучающую выборку, см. рисунок 4.5 выше. В первом блоке кода для ускорения вычислений при самом первом вызове метода `fit_and_score` создаётся столько копий исходной модели, сколько событий требуется спрогнозировать. Каждой копии модели задаются свои обучающие данные, в зависимости от того, какое событие эта модель будет прогнозировать. Благодаря этому каждый раз при вызове функции `fit_and_score` для прогнозирования каждого события будет использоваться своя отдельная дочерняя модель. Во втором блоке подготавливаются списки и массивы для сохранения и сравнения результатов (если `scalars` было `True`, то подготавливается массив для сохранения значений для их последующего усреднения). Наконец, в третьем блоке происходит обучение каждой дочерней модели, экстраполяция её параметров, предсказание с помощью неё очередного события и оценка качества прогноза события с помощью вызова функции `scorer` (если `scalars` было `True`, иначе сохранение прогноза события в список, а оценка качества прогноза выполняется после получения всех событий). Также, если `scalars` было `True`, то оценки качества прогноза каждого отдельного события усредняются с помощью функции `extra_scorer`. Разделение оценки качества прогноза событий на индивидуальное или групповое сделано для того, чтобы можно было досрочно выйти из цикла, если какое-то из событий не получится спрогнозировать, тем самым получится

сэкономить немного времени. При групповой оценке придётся в начале дождаться получения прогноза всех событий.

Функции оценки качества прогноза событий объявлены в модуле `Metrics`. Среди них абсолютные и относительные средние или среднеквадратичные ошибки. Например, функция для расчёта среднеквадратичной относительной ошибки выглядит следующим образом.

```
eventometrics.Metrics.root_mean_squared_relative_error(t_true,
t_lag_true, t_pred, Y_true, Y_pred, mix_weight=0.1,
sample_weights=None, **kwargs)
```

Параметры:

t_true – истинное время возникновения событий, скаляр или список / одномерный массив.

t_lag_true – истинное время возникновения последнего события в обучающей выборке, всегда скаляр.

t_pred – предсказанное время возникновения событий, скаляр или список / одномерный массив.

Y_true – истинные характеристики событий. Если каждое событие имеет только одну характеристику, то может являться скаляром или списком / одномерным массивом (для оценки многих событий). Если каждое событие имеет больше одной характеристики, то может являться одномерным или двумерным списком/массивом.

Y_pred – предсказанные характеристики событий, скаляр, одномерный или двумерный список/массив, в зависимости от того, сколько событий и сколько характеристик каждое из них несёт.

mix_weight – весовой коэффициент, с которым смешиваются ошибки прогноза времени возникновения t и ошибки прогноза характеристик Y . По умолчанию 0.1.

sample_weights – веса наблюдений, по умолчанию не заданы.

kwargs – остальные параметры, которые могут передаваться (только для совместимости).

Код функции следующий.

```

if invalid_parameters(t_pred, Y_pred):
    return np.inf
if np.ndim(t_true)==0:
    t_div = max(abs(t_true - t_lag_true) , eps)
    if np.ndim(Y_true)==0:
        Y_div = max(abs(Y_true), eps)
        return abs(t_true - t_pred)/t_div + mix_weight * abs(Y_true - Y_pred)/Y_div
    else:
        Y_div = abs(Y_true)
        Y_div[Y_div == 0] = eps
        return abs(t_true - t_pred)/t_div + mix_weight *
            np.sqrt(np.mean((np.abs(Y_true - Y_pred)/Y_div)**2))
else:
    t_div = np.zeros(len(t_true))
    t_div[1:] = t_true[1:] - t_true[0:-1]
    t_div[0] = t_true[0] - t_lag_true
    t_div = np.abs(t_div)
    t_div[t_div == 0] = eps
    Y_div = abs(Y_true)
    Y_div[Y_div == 0] = eps
    if len(Y_true.shape) == 1:
        if sample_weights is None:
            return np.sqrt( np.sum((np.abs(t_true - t_pred) / t_div + mix_weight *
                abs(Y_true - Y_pred) / Y_div)**2)/len(t_true))
        else:
            return np.sqrt(np.sum(( np.abs(t_true - t_pred) / t_div + mix_weight *
                abs(Y_true - Y_pred)/Y_div)**2 * sample_weights) / np.sum(sample_weights))
    else:
        if sample_weights is None:
            return np.sqrt(np.sum( (np.abs(t_true - t_pred) / t_div + mix_weight *
                np.sqrt(np.mean( ( (Y_true - Y_pred) / Y_div )**2, axis=1) ) )**2 )/len(t_true))
        else:
            return np.sqrt(np.sum( (np.abs(t_true - t_pred) / t_div + mix_weight *
                np.sqrt(np.mean( ( (Y_true - Y_pred) / Y_div )**2, axis=1) ) )**2 * sample_weights
                )/np.sum(sample_weights))

```

В начале проверяется, что предсказанные события не содержат нечисловые данные, иначе возвращается `inf` (бесконечная ошибка). Затем в зависимости от того, сколько передано для оценки событий и сколько характеристик имеют события, происходит расчёт средней квадратичной относительной ошибки. Для того, чтобы избежать возможную ошибку деления на ноль, к знаменателю прибавляется малая величина (наименьшее значение для `float`).

Функции расчёта других метрик реализованы аналогичным образом.

Пример поиска оптимальных гиперпараметров

Применение поиска на сетке очень похоже на то, как это делается в библиотеке `sklearn`, с небольшими отличиями в параметрах. Так же как в примере выше, вначале создаём модель, можно не указывать параметры, так как они всё равно подбираются.

```

from eventometrics.CapacityModels import SimpleCapacity
from eventometrics.Extrapolators import QuinnFernandesExtrapolator
model = SimpleCapacity(t, Y,
                       knots_number=m,
                       extrapolator=QuinnFernandesExtrapolator()
)

```

Задаём сетку гиперпараметров следующим образом.

```

param_grid = {"alpha": [0, 10**3, 10**4, 10**5, 10**6, 10**7, np.inf],
              "n1": [0, 10, 20, 30],
              "n2": [0, 10, 20, 30]
             }

```

Создаём объект `GridSearchCV` для поиска на сетке, указывая оптимизируемую модель, сетку гиперпараметров и при желании другие параметры, например, как сделано ниже. После этого запускаем сам поиск на сетке с помощью вызова функции `fit`.

```

from eventometrics.Search import GridSearchCV

gs = GridSearchCV(estimator= model,
                  param_grid=param_grid,
                  scorer="re",
                  extra_scorer="rms",
                  n_jobs=-2,
                  refit=True,
                  num_val_events=3,
                  n_predict=200,
                  mix_weight=0.1,
                  is_int=["n1", "n2"],
                  verbose=1
                  )

gs.fit()

```

Информация о запуске и завершении локальной оптимизации для каждой ячейки сетки выводится на экран при `verbose`, отличным от нуля, рисунок 4.7. Заметим, что при многопоточном выполнении (при `n_job`, отличным от единицы) сообщения о старте одной задачи и завершении другой могут быть перемешаны.

Посмотреть лучшие найденные значения гиперпараметров и целевой функции можно, обратившись к атрибутам `best_params` и `best_value`.

```

print("best_params = ", gs.best_params)
print("best_value = ", gs.best_value)

```

Результаты будут

```

best_params = {'alpha': 5303216.032479669, 'n1': 7, 'n2': 25}
best_value = 0.04690435938458174

```

Заметим, что в данном примере `best_value` показывает среднюю квадратичную относительную ошибку для прогноза трёх событий всего ~4.7%

(среднее квадратичное усреднение для смеси относительной ошибки прогноза даты t и 0.1 от относительной ошибки прогноза характеристики Y).

```
Starting grid search for 54 candidates
[1/54] START {'alpha': 500.0, 'n1': 5, 'n2': 5}
[2/54] START {'alpha': 500.0, 'n1': 5, 'n2': 15}
[3/54] START {'alpha': 500.0, 'n1': 5, 'n2': 25}
[4/54] START {'alpha': 500.0, 'n1': 15, 'n2': 5}
[5/54] START {'alpha': 500.0, 'n1': 15, 'n2': 15}
[6/54] START {'alpha': 500.0, 'n1': 15, 'n2': 25}
[7/54] START {'alpha': 500.0, 'n1': 25, 'n2': 5}
[8/54] START {'alpha': 500.0, 'n1': 25, 'n2': 15}
[9/54] START {'alpha': 500.0, 'n1': 25, 'n2': 25}
[10/54] START {'alpha': 5500.0, 'n1': 5, 'n2': 5}
...
[32/54] START {'alpha': 550000.0, 'n1': 15, 'n2': 15}
[9/54] End : score = 0.11605475509417654 , converged = True , best parameters = {'alpha': 477.7777777777778, 'n1': 27, 'n2': 26} ...
[33/54] START {'alpha': 550000.0, 'n1': 15, 'n2': 25}
[6/54] End : score = 0.09269478079811026 , converged = True , best parameters = {'alpha': 452.7777777777778, 'n1': 16, 'n2': 25} ...
[1/54] End : score = 0.12197449455345165 , converged = True , best parameters = {'alpha': 507.40740740740733, 'n1': 6, 'n2': 5} ...
[34/54] START {'alpha': 550000.0, 'n1': 25, 'n2': 5}
[35/54] START {'alpha': 550000.0, 'n1': 25, 'n2': 15}
[18/54] End : score = 0.11605368809300402 , converged = True , best parameters = {'alpha': 5300.0, 'n1': 27, 'n2': 26} ...
[36/54] START {'alpha': 550000.0, 'n1': 25, 'n2': 25}
[8/54] End : score = 0.10726272012249848 , converged = True , best parameters = {'alpha': 447.993827160494, 'n1': 30, 'n2': 21} ...
[15/54] End : score = 0.0895346405104444 , converged = True , best parameters = {'alpha': 512.45046486816, 'n1': 14, 'n2': 19} ...
...
[50/54] End : score = 0.08257626359181718 , converged = True , best parameters = {'alpha': 63026855.03348533, 'n1': 21, 'n2': 26} ...
[45/54] End : score = 0.11131161881423567 , converged = True , best parameters = {'alpha': 5661149.788691994, 'n1': 26, 'n2': 25} ...
[54/54] End : score = 0.10475256645223151 , converged = True , best parameters = {'alpha': 83592412.04128245, 'n1': 26, 'n2': 27} ...
[53/54] End : score = 0.10893426531929006 , converged = True , best parameters = {'alpha': 105550293.44420114, 'n1': 24, 'n2': 16} ...
```

Источник: составлено автором.

Рисунок 4.7 – Информация о поиске гиперпараметров с помощью поиска на сетке и локальной оптимизации

В случае, если параметр `refit` был `True` (по умолчанию), то можно сразу пользоваться моделью, обученной на всём наборе данных с лучшей комбинацией гиперпараметров, обратившись к `best_estimator`. Например, можно сразу получить прогноз будущего события, вызвав метод `predict`.

```
t_pred, y_pred = gs.best_estimator.predict(n_predict=200)
```

После этого вывести информацию о полученном прогнозе будущего события.

```
t_pred_date = (t_start + pd.Timedelta(round(t_pred), "d")).date()
print("t_pred_date = ", t_pred_date, " y_pred = ", y_pred)
print("T_real_date = ", T_future_real_date, " y_real = ", Y_future_real)
error = t_pred - T_future_real
dt = T_future_real - t[-1]
print("dt = ", dt)
rel_error = error / (T_future_real - t[-1])
print("error = ", error, " rel_error = ", rel_error)
```

Результатом будет

```
t_pred_date = 2024-01-06 y_pred = 3607.4136617740464
T_real_date = 2024-01-20 y_real = 3633.927626
dt = 145.0
error = -14.0 rel_error = -0.09655172413793103
```

Видим, что хоть на трёх предыдущих событиях ошибка была всего лишь 4.7%, ошибка для будущего события составила 9.65%.

Можно подбирать и другие гиперпараметры, например гиперпараметр `Nharm`, который задействуется при экстраполяции в методе Куинна-Фернандеса.

```
param_grid = {"alpha": [0, 10**3, 10**4, 10**5, 10**6, 10**7, np.inf],
              "n1": [0, 10, 20, 30],
              "n2": [0, 10, 20, 30],
              "Nharm": [9, 10, 11, 12]
             }
gs = GridSearchCV(estimator= model,
                  param_grid=param_grid,
                  excluded_from_local_search="Nharm",
                  scorer="re",
                  extra_scorer="rms",
                  n_jobs=-2,
                  refit=True,
                  num_val_events=3,
                  n_predict=200,
                  mix_weight=0.1,
                  is_int=["n1", "n2"],
                  verbose=1
                 )
gs.fit()
```

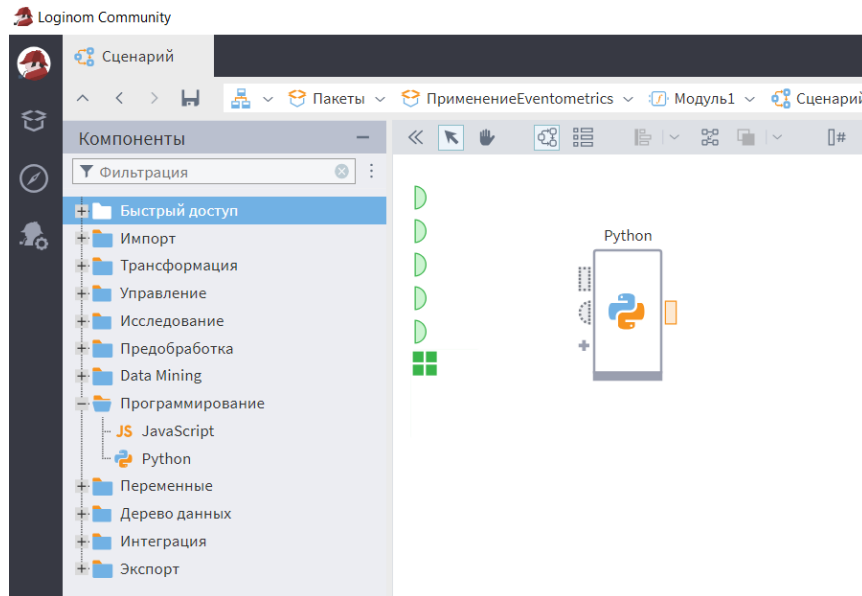
Надо обратить внимание, что тут гиперпараметр `Nharm` был добавлен в исключение при локальной оптимизации, так как все возможные для него значения уже указаны в сетке параметров. Всего будет 216 стартовых позиций для локального поиска. Результат будет следующим.

```
best_params = {'alpha': 4967111.74938632, 'n1': 6, 'n2': 24, 'Nharm': 12}
best_value = 0.03819599480134542
t_pred_date = 2024-01-07 y_pred = 3610.2314451089956
T_real_date = 2024-01-20 y_real = 3633.927626
dt = 145.0
error = -13.0 rel_error = -0.0896551724137931
```

На трёх предшествующих событиях среднеквадратичная относительная ошибка прогноза составила всего 3.8%. Но при прогнозе будущего события получилось выиграть лишь один день в точности, и ошибка сократилась до 8.9%. Конечно же, могут встречаться случаи переобучения, когда предыдущие события предсказываются хорошо, а будущие события плохо. Качество прогноза будущего события может зависеть не только от гиперпараметров, но также от самого выбора способа экстраполяции параметров и, конечно же, от самой модели механизма образования событий.

4.2 Создание компонента в Loginom для прогнозирования событий

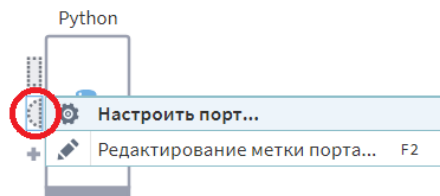
Благодаря разработанной библиотеке eventometrics, написанной на языке Python, не составит труда создать свой компонент для Loginom [174]. Для этого из компонентов (раздел программирование) перетягиваем в свой сценарий узел «Python», рисунок 4.8.



Источник: составлено автором.

Рисунок 4.8 – Начало создания собственного компонента в Loginom

Нажимаем правой кнопкой на значок входных переменных узла и выбираем «Настроить порт...», рисунок 4.9.



Источник: составлено автором.

Рисунок 4.9 – Настройка порта входных переменных

В открывшемся окне нажимаем на плюс и добавляем две переменные, `n_events` и `n_predict`. Указываем тип данных – Целый. Задаём значения 1 и 200. Должно получиться, как на рисунке 4.10. При желании можно будет добавить и остальные переменные, но ограничимся определением их в самом коде.

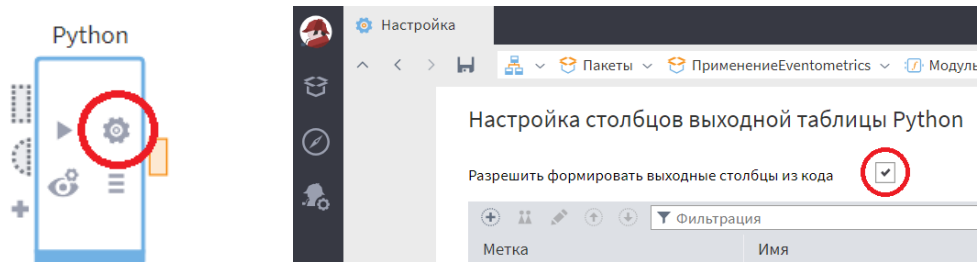
Настройка переменных

Метка	Имя	Назначение	Значение	
12 n_events	n_events	Не задано	1	
12 n_predict	n_predict	Не задано	200	

Источник: составлено автором.

Рисунок 4.10 – Создание входных переменных

Далее заходим в настройки узла, пропускаем настройку входных столбцов, но в настройке выходных столбцов ставим галку «Разрешить формировать выходные столбцы из кода», рисунок 4.11.



Источник: составлено автором.

Рисунок 4.11 – Настройка выходных столбцов

Наконец задаём код на языке Python для данного компонента.

```
import builtin_data
from builtin_data import InputTable, InputTables, InputVariables, OutputTable,
                        DataType, DataKind, UsageType
from builtin_pandas_utils import to_data_frame, prepare_compatible_table, fill_table

if InputTable:
    input_frame = to_data_frame(InputTable)

# ===== Take data from Input table =====
import numpy as np
import pandas as pd

t = pd.to_datetime(input_frame.t.dropna(), format='%d.%m.%Y')
t_start = min(t)
t = np.array([(x-t_start).days for x in t], dtype = np.double)
Y = input_frame.y.dropna().to_numpy(dtype=np.double)
n = len(t)

# ===== Take variables =====
```

```

if InputVariables.Count == 0:
    n_events = 1
    n_predict = 200
else:
    n_events = InputVariables.Items["n_events"].Value
    n_predict = InputVariables.Items["n_predict"].Value

# ===== Use model =====

from eventometrics.CapacityModels import SimpleCapacity
from eventometrics.Extrapolators import QuinnFernandesExtrapolator
from eventometrics.Search import GridSearchCV

model = SimpleCapacity(t, Y,
                      knots_number=3*n,
                      extrapolator=QuinnFernandesExtrapolator()
                      )
param_grid = {"alpha": [0, 10**3, 10**4, 10**5, 10**6, 10**7, np.inf],
              "n1": [0, 10, 20, 30, 100],
              "n2": [0, 10, 20, 30, 100]
              }
gs = GridSearchCV(estimator=model,
                  param_grid=param_grid,
                  scorer="re",
                  extra_scorer="rms",
                  n_jobs=-2,
                  refit=True,
                  num_val_events=3,
                  n_predict=n_predict,
                  mix_weight=0.1,
                  is_int=["n1", "n2"],
                  verbose=0
                  )

gs.fit()

t_pred, y_pred = gs.best_estimator.predict(n_predict=n_predict, n_events=n_events)

# ===== Output prediction as table =====

if np.ndim(t_pred)==0:
    if t_pred is not None:
        t_pred_date = [(t_start + pd.Timedelta(round(t_pred), "d")).date()]
    else:
        t_pred_date = [None]
    y_pred = [y_pred]
else:
    t_pred_date = [(t_start + pd.Timedelta(round(t_p), "d")).date() for t_p in t_pred]

if n_events == 1:
    df = {"t":t_pred_date, "y":y_pred}
else:
    df = {}
    for i in range(n_events):
        if i<len(t_pred_date):
            df["t"+str(i+1)] = [t_pred_date[i]]
            df["y"+str(i+1)] = [y_pred[i]]
        else:
            df["t"+str(i+1)] = [None]
            df["y"+str(i+1)] = [None]

output_frame = pd.DataFrame(df)

# ===== Output dataframe as Loginom table =====

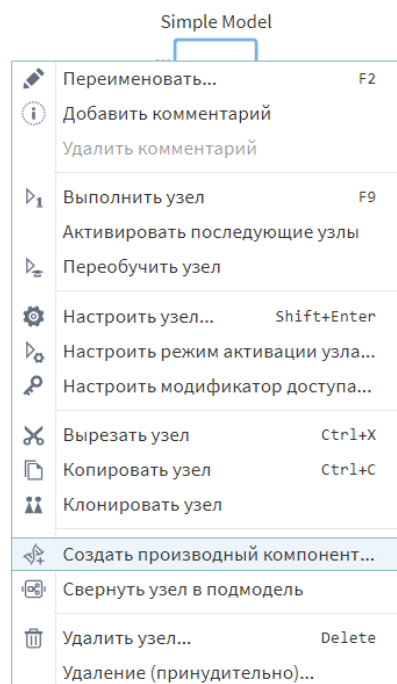
if isinstance(OutputTable, builtin_data.ConfigurableOutputTableClass):
    prepare_compatible_table(OutputTable, output_frame, with_index=False)
fill_table(OutputTable, output_frame, with_index=False)

```

В первом блоке кода идёт стандартный код узла, который обеспечивает преобразование входной таблицы в дата фрейм (таблица) `pandas`, чтобы с ней можно было работать дальше. Во втором блоке кода из этой таблицы извлекаем время `t` и характеристики `Y` событий. В третьем блоке преобразуем переданные в узел переменные в переменные Python. Наконец, в четвёртом блоке кода используем созданную библиотеку `eventometrics` для предсказания событий. В пятом блоке кода подготавливается выходной дата фрейм. В случае, если было спрогнозировано несколько событий, то они помещаются в отдельных столбцах (не строках). Всего подготавливается одна строка. Сделано это для того, чтобы можно было потом с помощью узла управления «Цикл» применять созданный нами узел для массовой обработки данных. Таким образом, прогнозы событий для разных клиентов будут располагаться в отдельных строках. Сами события будут располагаться в столбцах `t1, y1, t2, y2, t3, y3, ...`. Предполагается, что пользователя будет интересовать в первую очередь прогноз только одного события. В заключительном шестом блоке кода выходной дата фрейм `pandas` преобразуется в выходную таблицу `Loginom`.

На последней вкладке настроек узла задаём какое-нибудь название, например `Simple Model`, и нажимаем «сохранить» (на дискету) справа. На этом настройка узла Python закончена.

Чтобы создать из узла производный компонент, щёлкаем по нему правой кнопкой и в контекстном меню выбираем «Создать производный компонент...», рисунок 4.12. В появившемся окне указываем метку, например `Predict Events`, и щёлкаем далее. На этом создание собственного компонента в `Loginom` закончено, и его теперь можно использовать так же, как другие стандартные узлы.



Источник: составлено автором.

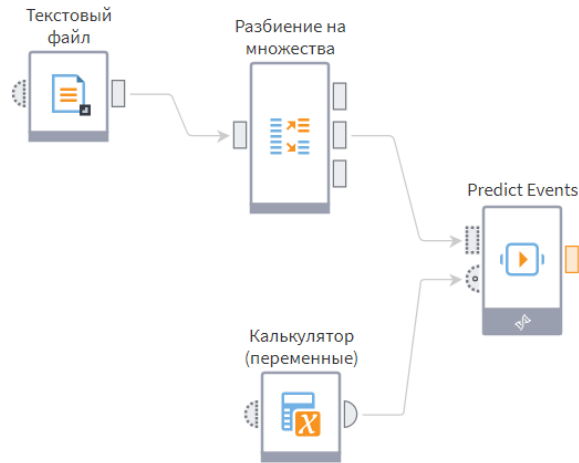
Рисунок 4.12 – Создание производного компонента

Пример применения собственного компонента в Logiom для прогнозирования событий

Прежде чем пользоваться нашим настроенным узлом, нам необходимо установить все используемые библиотеки Python, а именно pandas и eventometrics. Для установки библиотек необходимо открыть обычную командную строку (в windows в строке поиска ввести cmd). После этого ввести последовательно две команды (опция --upgrade обеспечивает обновление библиотеки до последней актуальной версии, в случае если библиотека уже была установлена, без этой опции новая версия библиотеки не установится).

```
pip install pandas --upgrade
pip install eventometrics --upgrade
```

Для демонстрации прогнозирования событий в Logiom с помощью созданного нами компонента перенесём узлы «Текстовый файл», «Разбиение на множества», «Калькулятор (переменные)» и сам наш компонент «Predict Events» (такую метку мы задали при создании производного компонента), который находится в разделе Производные компоненты. Соединяем все узлы, как показано на рисунке 4.13



Источник: составлено автором.

Рисунок 4.13 – Расположение и соединение узлов

В качестве примера будем использовать тот же набор данных, что и в нашем сквозном примере. В настройках узла «Текстовый файл» указываем путь к файлу, рисунок 4.14, и на следующей вкладке для настройки форматов импорта нажимаем «Определить автоматически», рисунок 4.15.

Импорт из текстового файла

Имя файла / URL	data/simple.csv		
Информация о файле	Нет	Кодовая страница	UTF-8 (65001)
Заголовок в первой строке	<input checked="" type="checkbox"/>	Пропустить строк	0

```
t;y
30.01.2020;3628,994024
08.04.2020;3615,590775
...
```

Источник: составлено автором.

Рисунок 4.14 – Загрузка файла

В настройках узла «Разбиение на множества» задаём, какое количество событий у нас будет использоваться для обучения, а какое – для последующего сравнения (не зависит от того, сколько событий потом пожелаем прогнозировать). Обязательно метод сэмплинга выбираем Последовательный, рисунок 4.16.

Настройка форматов импорта

Десятичный разделитель

Разделитель столбцов
Формат даты

Считать последовательные разделители одним
Разделитель даты

Ограничитель строк
Разделитель времени

Пусто
Истина

Переменный тип
Ложь

Обновить все | Определить типы данных | Кол-во строк для анализа: 25 | Исходные данные | Результат

Поля	31 t	9.0 y
Имя	t	y
Метка	t	y
Тип данных	31 Дата/Время	9.0 Вещественный
Вид данных	<input checked="" type="radio"/> Непрерывный	<input checked="" type="radio"/> Непрерывный
Использовать	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1	30.01.2020, 00:00	3 628,99
2	08.04.2020, 00:00	3 615,59
3	30.08.2020, 00:00	3 625,53
4	12.11.2020, 00:00	3 606,92

Источник: составлено автором.
Рисунок 4.15 – Настройка форматов импорта

Разбиение на множества

Состояние входа: [Активировать](#)

Общее число записей:

Множество	Способ	% Размер в процентах	Размер в строках
Обучающее	<input checked="" type="checkbox"/>	50	14
Тестовое	<input checked="" type="checkbox"/>	50	3
Итого:		Не определено	17

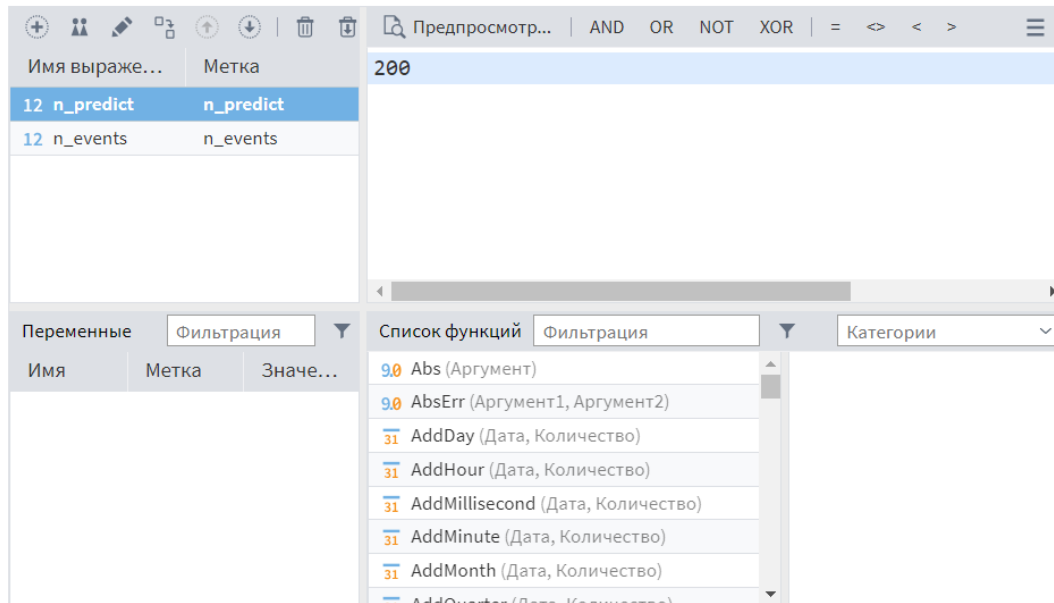
Метод сэмплинга:

Источник: составлено автором.
Рисунок 4.16 – Настройка узла «Разбиение на множества»

В узле «Калькулятор (переменные)» нажимаем на плюс и создаём две переменные `n_predict` и `n_events` со значениями 200 и 1 соответственно, тип переменных задаём как Целые, рисунок 4.17. Эти переменные будут использоваться для управления нашим компонентом.

Примечание – Вместо узла «Калькулятор (переменные)» можно было настраивать переменные пользователя на левой панели, но потом от них связи идут через всё пространство, зачастую перечёркивая имеющиеся блоки, что неэстетично, использование калькулятора переменных кажется более подходящим.

Редактирование выражений



Источник: составлено автором.

Рисунок 4.17 – Создание управляющих переменных `n_predict` и `n_events` в узле «Калькулятор (переменные)»

Наконец, кликаем по треугольничку (Выполнить узел) в нашем компоненте Predict Events. После выполнения узла посмотреть результат можно, кликнув два раза по выходному порту нашего компонента. В результате видим предсказанное событие, рисунок 4.18.

Выходной набор данных			
#	ab t	90 y	
1	2024-01-06	3 607,41	

Источник: составлено автором.

Рисунок 4.18 – Результат работы компонента

Если требуется предсказать несколько событий, то задаём `n_event`, например, равным 2, видим такой результат, рисунок 4.19.

Выходной набор данных				
#	ab t1	90 y1	ab t2	ab y2
1	2024-01-06	3 607,41	<null>	<null>

Источник: составлено автором.

Рисунок 4.19 – Попытка предсказать два события

Вместо второго события можно увидеть `null`, что означает, что за 200 дней оно не успело образоваться. Но если увеличить `n_predict` до 300, то получаем искомое второе событие, рисунок 4.20.

Выходной набор данных				
#	ab t1	90 y1	ab t2	90 y2
1	2024-01-06	3 607,41	2024-03-26	3 643,57

Источник: составлено автором.

Рисунок 4.20 – Предсказание двух событий

Опять заметим, что прогноз на большую временную перспективу может обладать плохой точностью.

Созданный нами компонент можно также использовать в цикле, например, для предсказания будущих событий по множеству разных клиентов. Для этого следует использовать элементы Группировка, чтобы получить список клиентов, Цикл, чтобы пробегать по клиентам, Фильтр строк, чтобы фильтровать транзакции по клиентам, и наш компонент Predict Events для предсказания будущих транзакций. Возможно, понадобятся другие дополнительные приготовления для обработки данных.

Выводы по главе 4

1) Разработана библиотека eventometrics на языке Python для прогнозирования событий. Библиотека состоит из модулей CapacityModels, Extrapolators, Search, Metrics, а также модуля optimization. Основные вычисления оптимизированы с помощью Cython. Реализован поиск оптимальных гиперпараметров с помощью поиска на сетке и локальной оптимизации. Библиотека имеет интерфейс, похожий на интерфейс библиотеки sklearn. Благодаря разработанной библиотеке значительно упрощается работа исследователей и аналитиков, занимающихся прогнозированием экономических событий.

2) Реализован компонент для платформы бизнес-аналитики Loginom. Компонент использует созданную в Python библиотеку eventometrics. Подробно описан процесс создания компонента. Показан пример использования компонента для прогноза будущих событий. Создание компонента обеспечивает возможность использования результатов

исследования без написания программного кода путём интеграции с визуальными средствами платформы бизнес-аналитики.

Глава 5

Апробация и сравнение нового подхода с существующими методами

5.1 Апробация разработанных методик

Разработанная методология была апробирована на данных о поставках биологически активных добавок российского представительства компании ООО «Квайссер Фарма» в аптечные пункты в разных регионах Российской Федерации. Использовались данные самого ходового товара – Doppelherz Актив Омега-3, который выпускается в разных упаковках, содержащих от 30 до 120 пилюль, рисунок 5.1.

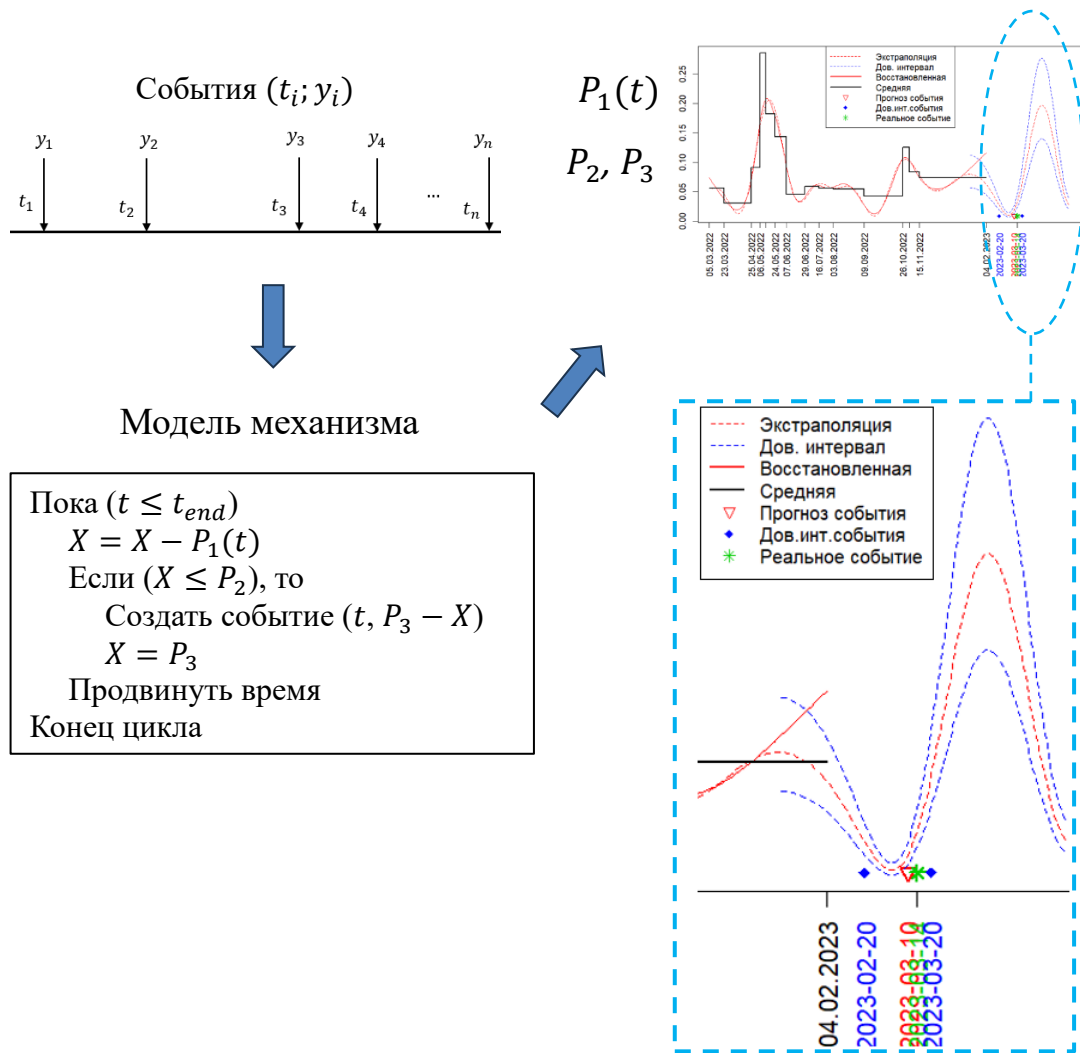


Источник: составлено автором по официальным изображениям.

Рисунок 5.1 – Биологически активные добавки компании ООО «Квайссер Фарма» – Doppelherz Актив Омега-3 в разных упаковках

В приводимых ниже примерах применялась простейшая модель, как в сквозном примере. При апробации использовался программный код на языке R (библиотека на Python была написана позже на основе ранее составленного кода R). Динамические параметры экстраполировались с помощью разложения на сумму ограниченного количества гармонических функций. Гиперпараметры определялись вручную, так как средства автоматизации ещё не были созданы. Подробности реализации на языке R представлены в работе [217, с. 203–248].

Во всех примерах использовалась одна и та же схема для прогнозирования событий, рисунок 5.2.



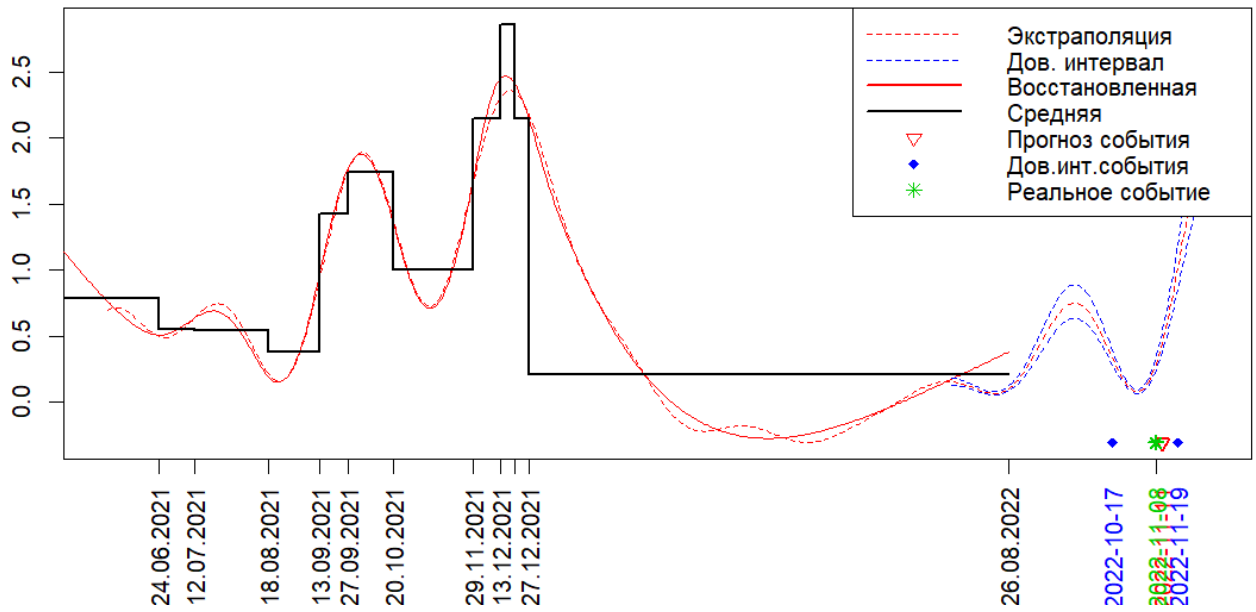
Источник: составлено автором.

Рисунок 5.2 – Схема прогнозирования событий и построение доверительного интервала

Доверительный интервал строился способом, когда получается раннее и позднее время наступления событий с помощью моделирования механизма с параметром $P_1(t)$, заниженным и завышенным на величину нужной доверительной вероятности. Дисперсия для параметра $P_1(t)$, определялась на основе формулы (2.69) в зависимости от величины приглушения исходных данных при заданном коэффициенте сглаживания.

Отдельные примеры представлены на рисунках 5.3, 5.4, 5.5, 5.6. В некоторых примерах имеется совсем незначительное количество событий, тем не менее по ним получается восстановить параметры механизма и получить прогноз.

**Доппельгерц Актив Омега 3 капс. №30
Калининградская область**

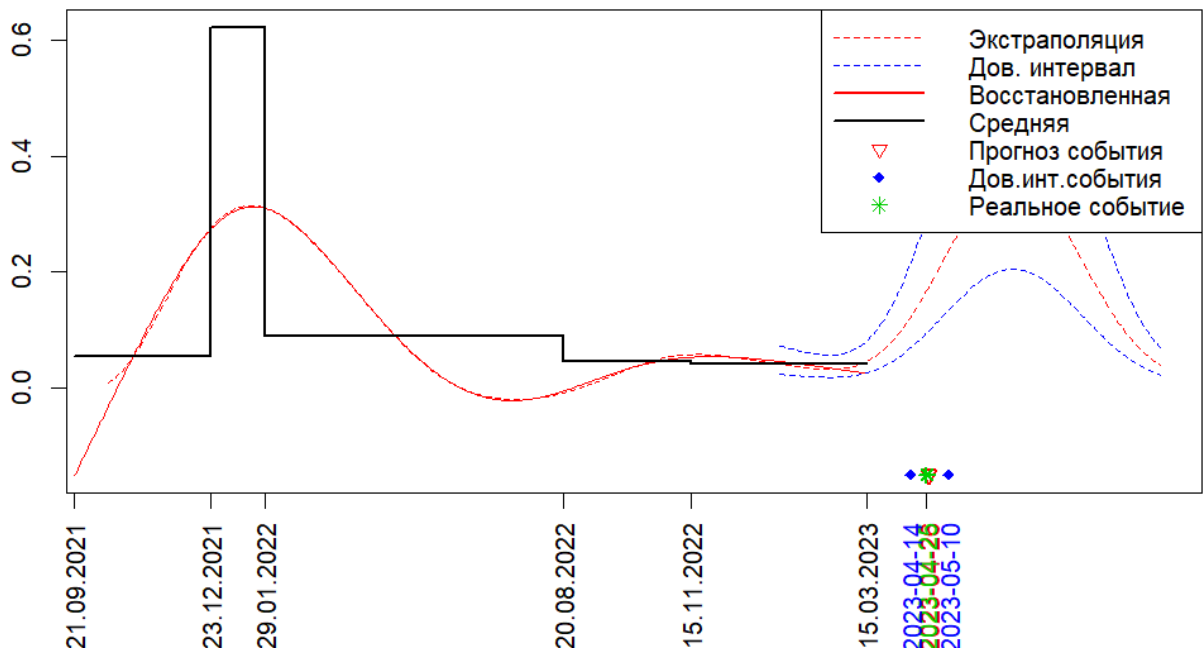


Ступенчатая функция показывает среднее значение функции на этом интервале $y_i/(t_{i+1} - t_i)$, площадь под каждой из ступенок (до оси ноль) – наблюдаемый интеграл u_i .

Источник: составлено автором.

Рисунок 5.3 – Прогноз и доверительный интервал, Калининградская область

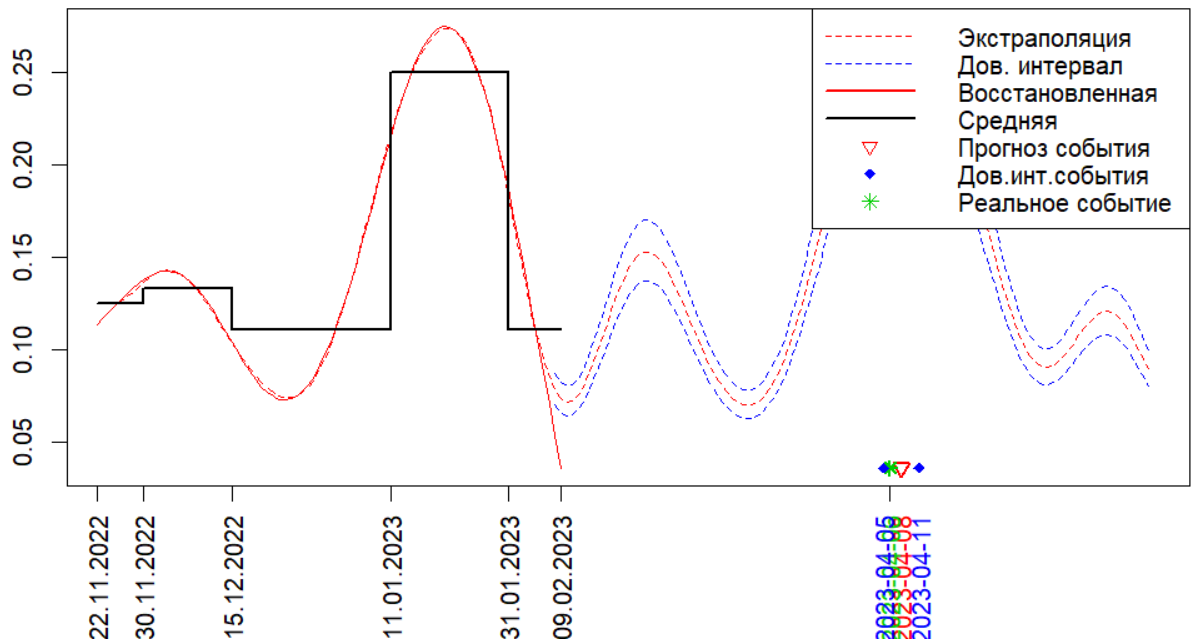
**Доппельгерц Актив Омега 3 капс. №30
Ростовская область**



Источник: составлено автором.

Рисунок 5.4 – Прогноз и доверительный интервал, Ростовская область

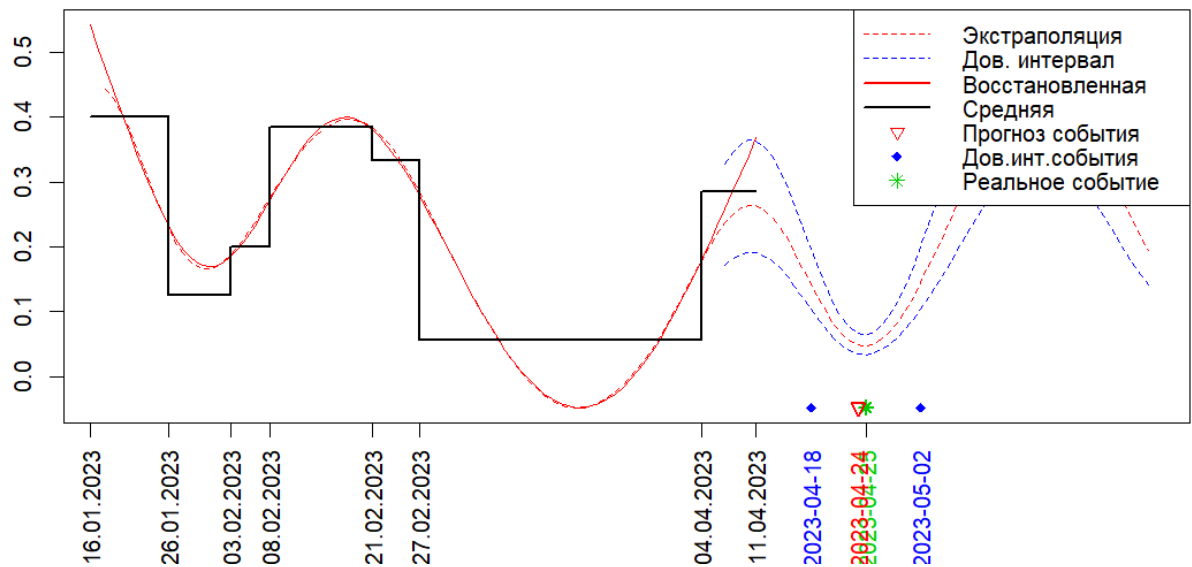
**Доппельгерц Киндер Омега-3 д/дет. с 7лет капс. №45
Московская область**



Источник: составлено автором.

Рисунок 5.5 – Прогноз и доверительный интервал, Московская область

**Доппельгерц Актив Омега-3 Концентрат капс. №60
Санкт-Петербург**



Источник: составлено автором.

Рисунок 5.6 – Прогноз и доверительный интервал, Санкт-Петербург

В работе [217] представлено множество других примеров с подробными пояснениями, касающимися предварительной обработки данных, восстановления параметров, подбора гиперпараметров, построения доверительных интервалов. Также разбираются причины, по которым в некоторых примерах были получены неадекватные модели (ошибка при

экстраполяции, ошибка при восстановлении функции, ошибка при выборе модели). Подробности в работе [217].

Заметим, что целью является сам прогноз событий. Получение экономической выгоды от способности предсказывать события выходит за рамки данной работы. Тем не менее краткое описание экономических выгод дано в конце главы.

5.2 Сравнение с методами классификации

В данном параграфе сравнивается предлагаемый подход, основанный на восстановлении параметров модели механизмов образования событий, с такими методами, как методы классификации, методы регрессии, метод на основе потоков случайных событий и два других метода. Сравняется точность прогнозирования последнего события, полученного моделью по выборке без этого последнего события. Рассмотрены как примеры на искусственно смоделированных данных, так и примеры на реальных данных. Целью раздела является стремление показать конкурентоспособность, а иногда даже превосходство предлагаемого подхода для прогнозирования событий над существующими методами на некоторых примерах (существующие методы несколько не умаляются, в то же время показывается, как их можно адаптировать к прогнозированию событий). Материалы данного раздела опубликованы в [219].

Напомним, что методы классификации обучаются на выборке наблюдений $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ и $Y_i \in \{0,1\}$, где $x_{i,j}$ – наблюдаемые значения признаков, относящиеся к тому же наблюдению i , для которого известна метка класса Y_i . В нашей задаче данными являются только события $(t_i; y_i)$, где t_i – момент времени возникновения события, y_i – числовые характеристики событий (например объёмы покупок). Как по этим данным составить задачу классификации, позволяющую прогнозировать будущие события $(t_{n+1}; y_{n+1})$? К сожалению, методами классификации нельзя предсказывать числовые значения, поэтому предсказать y_{n+1} не получится.

Но можно научить модель предсказывать момент возникновения события t_{n+1} . Только это будет не момент времени, а интервал времени, на котором может либо появиться, либо не появиться событие. В первой главе на рисунке 1.3 уже была кратко представлена схема модели классификации. Осталось разобраться, что будет выступать в качестве обучающей выборки наблюдений.

Модель для задачи классификации появления событий

Разобьём ось времени на интервалы одинаковой ширины. Но ширина может быть разной. Причём эта ширина будет влиять на объясняющую способность модели. В нашей модели будем пробовать разные значения ширины интервалов (это будут разные модели). Ширина интервала будет 14 дней, 7 дней и 3 дня.

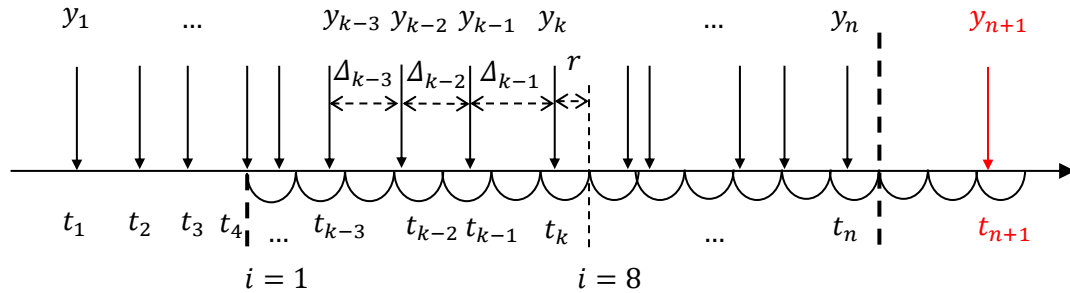
В нашей модели наблюдения будут состоять из следующей информации (здесь k – номер предшествующего события):

- $\Delta_{k-1}, \Delta_{k-2}, \Delta_{k-3}$ – ширина трёх предшествующих интервалов между событиями;
- $\bar{\Delta}$ – средняя ширина всех предшествующих интервалов между событиями;
- r – прошедшее время спустя предшествующее событие до начала текущего интервала;
- ν – средняя частота всех предшествующих событий;
- $u_k, u_{k-1}, u_{k-2}, u_{k-3}$ – значения предшествующих событий;
- \bar{u} – среднее значение всех предшествующих событий;
- $v_{k-1}, v_{k-2}, v_{k-3}$ – средняя скорость, с которой расходовался запас (т.е. $v_{k-1} = u_{k-1}/(t_k - t_{k-1})$);
- \bar{v} – среднее по всем предшествующим событиям средних скоростей.

Итого 15 признаков.

Так как для каждого наблюдения задействуется информация о трёх интервалах между предшествующими событиями, то началом самого первого

наблюдения выбирается момент появления четвёртого события t_4 . От этого момента будут откладываться интервалы шириной, например, 14 дней, которые и будут являться наблюдениями. Последним интервалом будет тот, в который попало последнее событие. Пример формирования 8-го наблюдения показан на рисунке 5.7.



Источник: составлено автором.

Рисунок 5.7 – Пример формирования 8-го наблюдения

Появление событий на интервале приводит к метке класса $Y_i = 1$, а отсутствие – к метке класса $Y_i = 0$. В результате получается обучающий набор данных. Для простейшего сквозного примера (моделирования системы управления запасами, таблица 2.1) обучающий набор данных представлен на рисунке 5.8.

Для предсказания будущего события требуется подготовить свой набор данных (тестовый/контролирующий). Для этого всего лишь продолжаем откладывать интервалы от последнего интервала обучающего набора данных. Так же как и ранее, формируем 15 признаков для каждого интервала. Это и будут наблюдения для тестовой выборки. Прекращаем откладывать новые интервалы, когда превысим некоторый горизонт, например в 200 дней. Пример тестовой выборки для простейшего примера показан на рисунке 5.9.

Что в обучающей выборке, что в контролирующей у наблюдений многие значения признаков повторяются. Это происходит потому, что если для очередного наблюдения (интервала) набор предшествующих событий не поменялся, то и соответствующие им признаки не будут меняться. В тестовом наборе данных можно заметить, что изменяется лишь признак r (прошедшее время спустя предшествующее событие).

	dt1	dt2	dt3	avg_dt	y	y1	y2	y3	avg_y	r	frequen...	v1	v2	v3	avg_v	Y
0	74	144	69	95.66667	3606.921...	3625.529...	3615.590...	3628.994...	3619.259...	0	0.01394	48.99365	25.10827	52.59412	52.59412	0
1	74	144	69	95.66667	3606.921...	3625.529...	3615.590...	3628.994...	3619.259...	14	0.01394	48.99365	25.10827	52.59412	52.59412	0
2	74	144	69	95.66667	3606.921...	3625.529...	3615.590...	3628.994...	3619.259...	28	0.01394	48.99365	25.10827	52.59412	52.59412	0
3	74	144	69	95.66667	3606.921...	3625.529...	3615.590...	3628.994...	3619.259...	42	0.01394	48.99365	25.10827	52.59412	52.59412	0
4	74	144	69	95.66667	3606.921...	3625.529...	3615.590...	3628.994...	3619.259...	56	0.01394	48.99365	25.10827	52.59412	52.59412	0
5	74	144	69	95.66667	3606.921...	3625.529...	3615.590...	3628.994...	3619.259...	70	0.01394	48.99365	25.10827	52.59412	52.59412	0
6	74	144	69	95.66667	3606.921...	3625.529...	3615.590...	3628.994...	3619.259...	84	0.01394	48.99365	25.10827	52.59412	52.59412	0
7	74	144	69	95.66667	3606.921...	3625.529...	3615.590...	3628.994...	3619.259...	98	0.01394	48.99365	25.10827	52.59412	52.59412	1
8	111	74	144	99.50000	3609.150...	3606.921...	3625.52...	3615.590...	3617.237...	1	0.01256	32.49479	48.993...	25.10827	38.85119	0
9	111	74	144	99.50000	3609.150...	3606.921...	3625.52...	3615.590...	3617.237...	15	0.01256	32.49479	48.993...	25.10827	38.85119	0
10	111	74	144	99.50000	3609.150...	3606.921...	3625.52...	3615.590...	3617.237...	29	0.01256	32.49479	48.993...	25.10827	38.85119	0
11	111	74	144	99.50000	3609.150...	3606.921...	3625.52...	3615.590...	3617.237...	43	0.01256	32.49479	48.993...	25.10827	38.85119	0
12	111	74	144	99.50000	3609.150...	3606.921...	3625.52...	3615.590...	3617.237...	57	0.01256	32.49479	48.993...	25.10827	38.85119	0
13	111	74	144	99.50000	3609.150...	3606.921...	3625.52...	3615.590...	3617.237...	71	0.01256	32.49479	48.993...	25.10827	38.85119	0
14	111	74	144	99.50000	3609.150...	3606.921...	3625.52...	3615.590...	3617.237...	85	0.01256	32.49479	48.993...	25.10827	38.85119	0
15	111	74	144	99.50000	3609.150...	3606.921...	3625.52...	3615.590...	3617.237...	99	0.01256	32.49479	48.993...	25.10827	38.85119	1
16	109	111	74	101.400...	3631.264...	3609.150...	3606.921...	3625.529...	3619.575...	4	0.01183	33.11148	32.49479	48.99365	42.23201	0
17	109	111	74	101.400...	3631.264...	3609.150...	3606.921...	3625.529...	3619.575...	18	0.01183	33.11148	32.49479	48.99365	42.23201	0
18	109	111	74	101.400...	3631.264...	3609.150...	3606.921...	3625.529...	3619.575...	32	0.01183	33.11148	32.49479	48.99365	42.23201	0
19	109	111	74	101.400...	3631.264...	3609.150...	3606.921...	3625.529...	3619.575...	46	0.01183	33.11148	32.49479	48.99365	42.23201	0
● ● ●																
66	101	78	141	101.272...	3612.136...	3649.902...	3601.899...	3618.266...	3618.503...	97	0.01077	36.13765	46.17820	25.66146	38.21648	0
67	101	78	141	101.272...	3612.136...	3649.902...	3601.899...	3618.266...	3618.503...	111	0.01077	36.13765	46.17820	25.66146	38.21648	1
68	119	101	78	102.750...	3629.911...	3612.136...	3649.90...	3601.899...	3619.380...	6	0.01054	30.35409	36.13765	46.17820	39.01265	0
69	119	101	78	102.750...	3629.911...	3612.136...	3649.90...	3601.899...	3619.380...	20	0.01054	30.35409	36.13765	46.17820	39.01265	0
70	119	101	78	102.750...	3629.911...	3612.136...	3649.90...	3601.899...	3619.380...	34	0.01054	30.35409	36.13765	46.17820	39.01265	0
71	119	101	78	102.750...	3629.911...	3612.136...	3649.90...	3601.899...	3619.380...	48	0.01054	30.35409	36.13765	46.17820	39.01265	0
72	119	101	78	102.750...	3629.911...	3612.136...	3649.90...	3601.899...	3619.380...	62	0.01054	30.35409	36.13765	46.17820	39.01265	1

Источник: составлено автором.

Рисунок 5.8 – Обучающий набор данных для простейшего примера

	dt1	dt2	dt3	avg_dt	y	y1	y2	y3	avg_y	r	frequency	v1	v2	v3	avg_v
0	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	3	0.01072	49.72482	30.35409	36.13765	38.75129
1	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	17	0.01072	49.72482	30.35409	36.13765	38.75129
2	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	31	0.01072	49.72482	30.35409	36.13765	38.75129
3	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	45	0.01072	49.72482	30.35409	36.13765	38.75129
4	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	59	0.01072	49.72482	30.35409	36.13765	38.75129
5	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	73	0.01072	49.72482	30.35409	36.13765	38.75129
6	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	87	0.01072	49.72482	30.35409	36.13765	38.75129
7	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	101	0.01072	49.72482	30.35409	36.13765	38.75129
8	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	115	0.01072	49.72482	30.35409	36.13765	38.75129
9	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	129	0.01072	49.72482	30.35409	36.13765	38.75129
10	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	143	0.01072	49.72482	30.35409	36.13765	38.75129
11	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	157	0.01072	49.72482	30.35409	36.13765	38.75129
12	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	171	0.01072	49.72482	30.35409	36.13765	38.75129
13	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	185	0.01072	49.72482	30.35409	36.13765	38.75129
14	73	119	101	100.46154	3620.06693	3629.91157	3612.13698	3649.90251	3619.42961	199	0.01072	49.72482	30.35409	36.13765	38.75129

Источник: составлено автором.

Рисунок 5.9 – Тестовый набор данных для простейшего примера

Для наблюдений тестового набора данных требуется предсказать метку класса Y_i .

В качестве методов классификации используются все методы библиотеки `scikit learn` [107] и метод классификации библиотеки `XGboost` [117]. Такими методами являются `SVC` (Support Vector Classification) [108], `LogisticRegression` [178], `KNeighborsClassifier` [109], `GaussianNB` (Gaussian Naive Bayes) [179], `DecisionTreeClassifier` [113], `RandomForestClassifier` [180], `GradientBoostingClassifier` [114], `MLPClassifier` (Multi-layer Perceptron Classification) [115], `XGBClassifier` [117].

Предсказанными метками класса для наблюдений тестовой выборки будет последовательность следующего вида

$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1).$$

В начале идёт какое-то количество нулей, а затем все остальные наблюдения принимают значение 1 (только для метода ближайших соседей последовательность имеет вид $(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1)$, когда единицы могут быть разбавлены нулями).

Прогнозом события тогда будем считать первый интервал, на котором появляется единица. Где именно на этом интервале появляется событие, методы классификации ответить не могут. Но можно изменять саму ширину интервалов времени при использовании методов классификации. Однако некоторые методы могут дать последовательность полностью из нулей, в этом случае предполагаем, что метод не ожидает появления события на указанном горизонте (возможно, событие могло бы появиться позже, но горизонт прогнозирования был ограничен, например 200 днями).

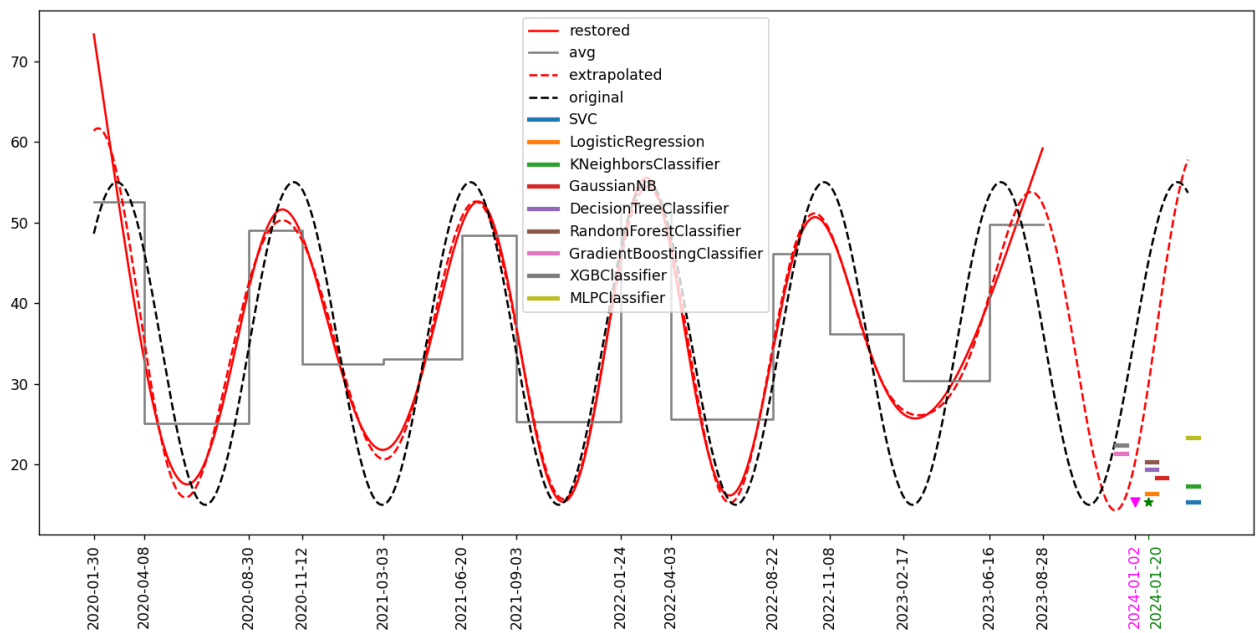
Далее представим результаты применения методов классификации для прогнозирования событий и сравним их с результатами применения нового подхода, основанного на моделях механизма образования событий, на разных примерах – как на искусственных, с разными механизмами образования событий, так и на примерах с реальными данными.

Простейшая модель механизма образования событий

Эта простейшая модель использовалась в качестве сквозного примера, начиная с первой главы. События образуются, как в моделях управления запасами. Ежедневно из запасов вычитается величина нестационарного спроса, который в этом простейшем примере задан гармонической функцией, рисунок 1.11 в первой главе. Когда запасы опускаются ниже некоторого критического уровня, происходит пополнение запасов до фиксированного максимума (мгновенно для простоты рассуждений). Динамика запасов изображена на рисунке 1.12 в первой главе, но она не наблюдаема. Заметим, что единственными наблюдаемыми для исследователя данными являются

сами события пополнения запаса (t_i, y_i) , таблица 2.1 во второй главе. Ничего другого исследователь не знает. Требуется спрогнозировать будущее событие.

Результаты применения предлагаемого метода и методов классификации изображены на рисунке 5.10. Фактическое событие образуется 20.01.2024 (обозначается зелёной звёздочкой ★). Прогноз по предлагаемому методу, основанному на восстановлении параметров механизма образования событий, даёт прогноз 02.01.2024 (обозначается пурпурным треугольником ▼), на 18 дней раньше (на 12% раньше от интервала 145 дней между событиями). Это отклонение можно легко объяснить тем, что функция спроса восстановилась и экстраполировалась завышенной, поэтому событие произошло раньше. Прогнозы разными методами классификации при ширине интервала, равной 14 дням, изображены горизонтальными линиями разного цвета.



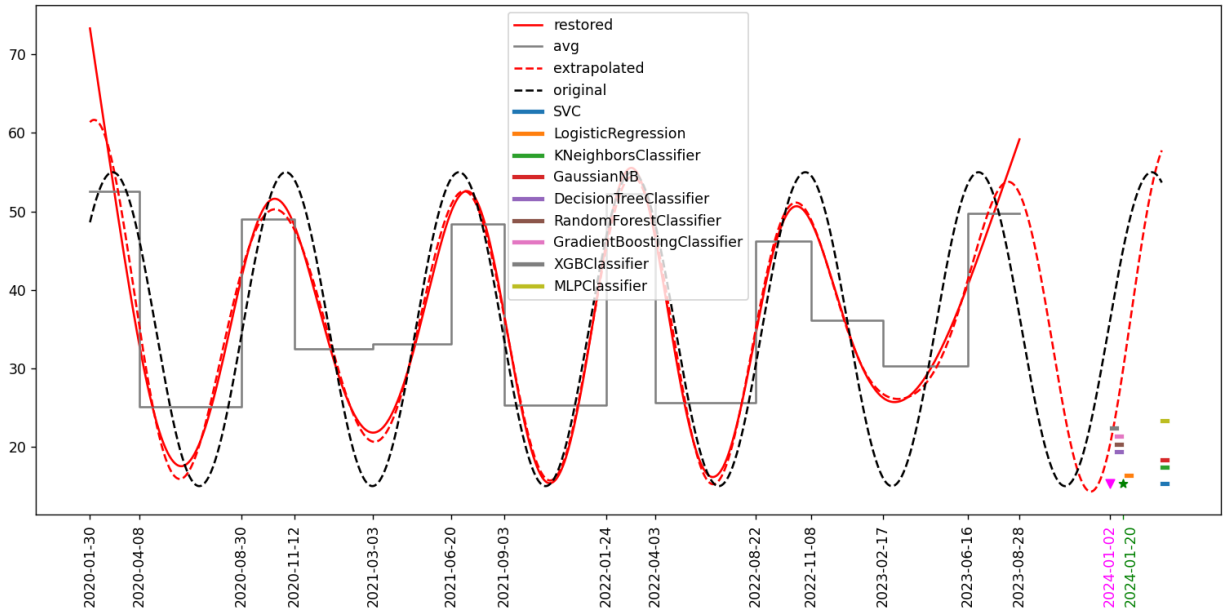
Источник: составлено автором.

Рисунок 5.10 – Простейшая модель, прогноз методами классификации при ширине интервала в 14 дней

Здесь лучше всех прогноз оказался у метода дерева решений и случайного леса. Как ни странно, но градиентный бустинг и XGBoost дали плохой результат. Причём уже затруднительно объяснить причину, по которой это произошло. Отрезки, которые изображены на самой правой границе (MLP,

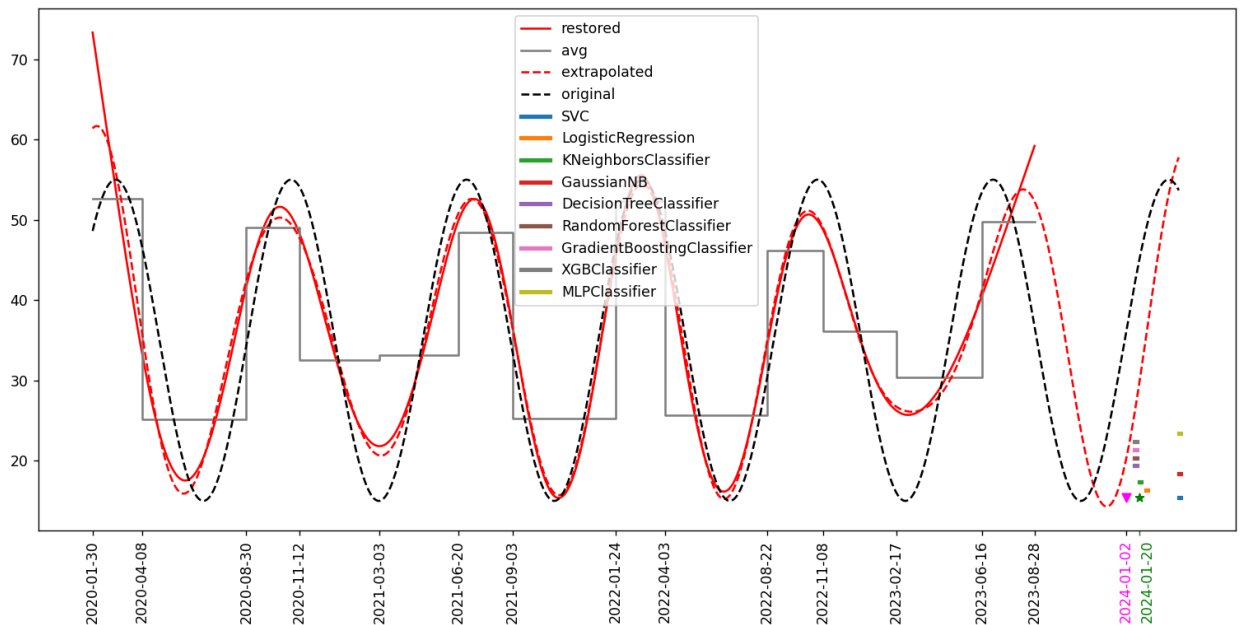
KNeighborsClassifier, SVC), на самом деле не дали прогноза, вся последовательность меток класса состояла для них из 0 (тут они всё же изображены, чтобы показать, что прогноз будет ещё позже).

Результаты для методов классификации при ширине интервалов 7 дней и 3 дня изображен на рисунках 5.11 и 5.12 соответственно.



Источник: составлено автором.

Рисунок 5.11 – Простейшая модель, прогноз методами классификации при ширине интервала в 7 дней



Источник: составлено автором.

Рисунок 5.12 – Простейшая модель, прогноз методами классификации при ширине интервала в 3 дня

При интервалах 7 дней и 3 дня методы классификации дают уже хороший прогноз, который близок к фактическому событию, и опережают предлагаемый метод по точности.

Однако стоит обратить внимание на следующее. На графиках заметно чередование высоких узких ступенек с низкими и широкими. А в модели классификации как раз использовалась информация о трёх предшествующих интервалах и средних значениях. В итоге методы классификации смогли определить закономерность. Если этого чередования не будет, то результат будет уже другим.

Простейшая модель, другие данные

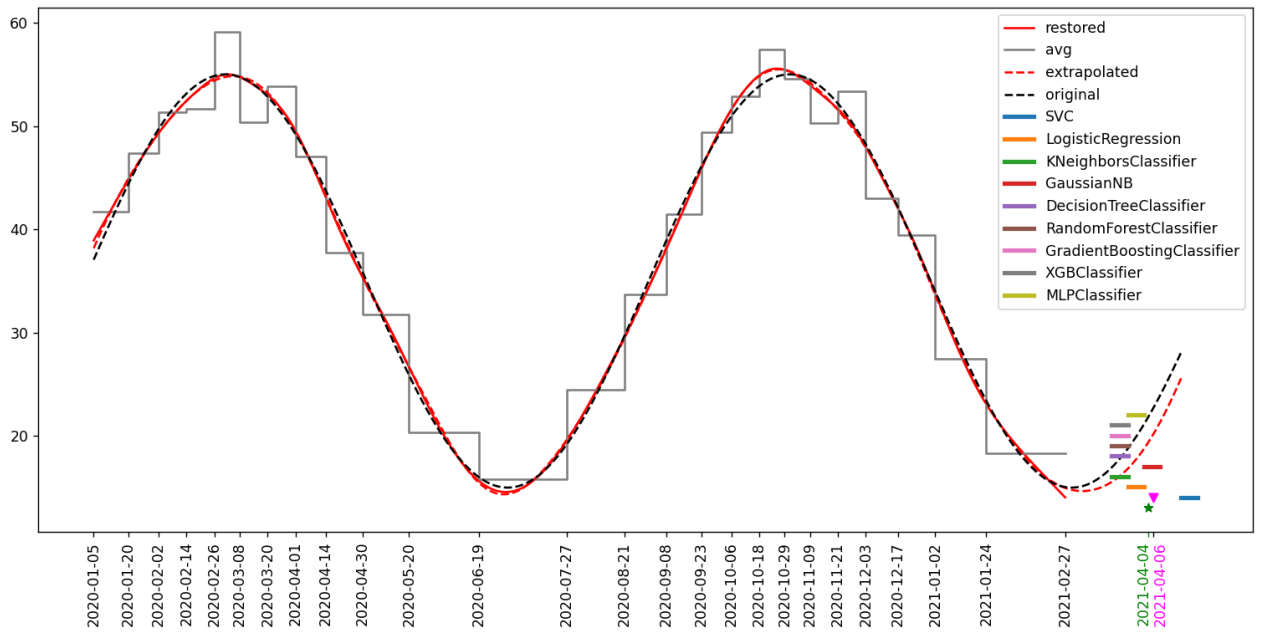
Рассмотрим также простейшую модель механизма образования событий, но с другими параметрами. В результате получаем данные событий, таблица 5.1. Пытаемся предсказать будущее событие по этой выборке.

Таблица 5.1 – Данные, полученные простейшей моделью с другими параметрами

Дата t_i	Объём y_i	Дата t_i	Объём y_i	Дата t_i	Объём y_i
05.01.2020	625,158	30.04.2020	634,506	29.10.2020	600,523
20.01.2020	615,908	20.05.2020	610,553	09.11.2020	603,194
02.02.2020	615,978	19.06.2020	601,407	21.11.2020	640,554
14.02.2020	620,388	27.07.2020	611,993	03.12.2020	601,940
26.02.2020	650,588	21.08.2020	607,081	17.12.2020	630,624
08.03.2020	604,167	08.09.2020	622,323	02.01.2021	604,972
20.03.2020	646,375	23.09.2020	642,343	24.01.2021	621,717
01.04.2020	612,177	06.10.2020	634,413	27.02.2021	611,960
14.04.2020	603,485	18.10.2020	631,813	-	-

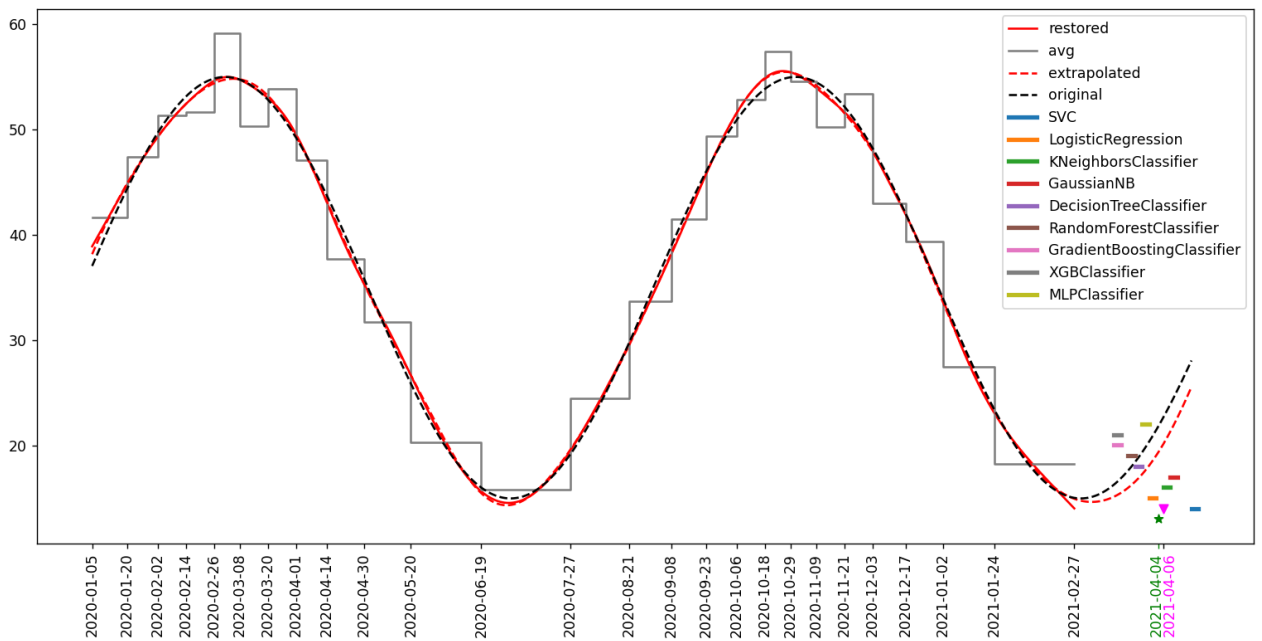
Источник: составлено автором.

Событий в выборке стало почти в 2 раза больше. Результаты должны были бы стать лучше. Результат использования методов классификации при ширине интервалов 7 дней и 3 дня изображён на рисунках 5.13 и 5.14 соответственно. Однако видно, что те методы классификации, которые хорошо показали себя ранее, дали совсем плохой прогноз. Особенно градиентный бустинг и XGBoost.



Источник: составлено автором.

Рисунок 5.13 – Простейшая модель с другими параметрами, прогноз методами классификации при ширине интервала в 7 дней



Источник: составлено автором.

Рисунок 5.14 – Простейшая модель с другими параметрами, прогноз методами классификации при ширине интервала в 3 дня

Смотреть надо не на то, что предлагаемый метод дал близкий прогноз, это было ожидаемо, так как событий стало больше и параметры были определены точнее, а на то, как сильно промахнулись методы классификации. Ошибка у градиентного бустинга и XGBoost составила практически половину интервала между событиями. У методов дерева решений и случайного леса

тоже существенная ошибка. Как так получилось? Скорее всего, это связано с тем, что в выборке наблюдений отсутствовала нужная последовательность интервалов между событиями и средних скоростей, соответствующая появлению следующего события. Поэтому многие методы классификации не смогли правильно угадать, появится событие или нет на заданном интервале.

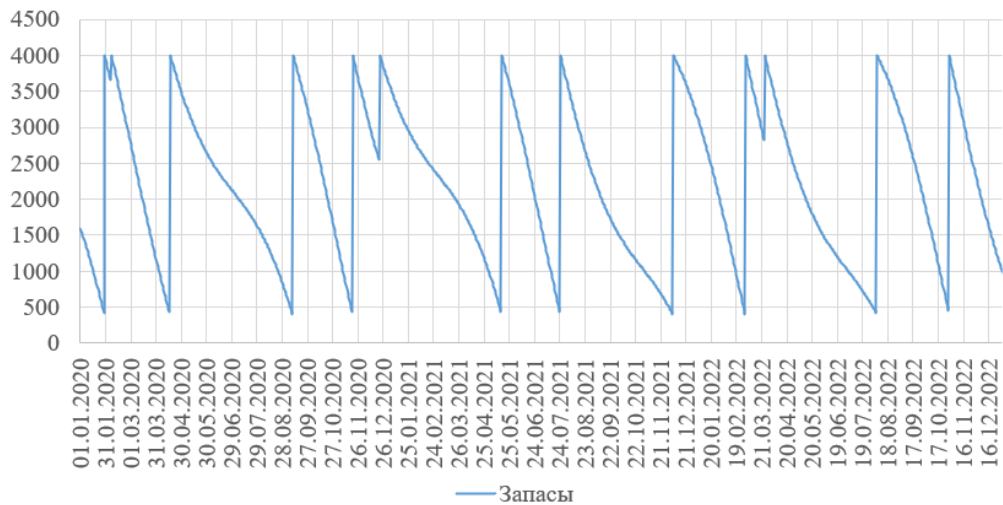
Модель с дополнительными событиями

Эта модель является усложнением простейшей модели. Помимо основных событий, вызванных исчерпанием запаса, имеется небольшое количество дополнительных событий, которые появились по какой-либо другой причине. Эти дополнительные события будут случайными, они не объясняются механизмом образования событий. Эти дополнительные события также приводят к пополнению запасов. Можно дать достаточно простую интерпретацию. Например, клиент пополняет запасы не только когда они закончились, но также либо когда он случайно оказывался рядом с магазином, либо когда появилась внезапная скидка на потребляемый товар. Заметим: механизм должен быть детерминированной последовательностью операторов. В нём не должно происходить разыгрывания случайных чисел, так как прогнозировать события с помощью броска кубика неправильно. Если записать модель следующим образом,

- 1) Пока ($t \leq t_{end}$)
- 2) $X = X - P_1(t)$
- 3) Если ($X \leq P_2$ или $\text{RandomTrue}(P_4)$), то
- 4) Создать событие ($t, P_3 - X$)
- 5) $X = P_3$
- 6) Продвинуть время t и обновить параметр $P_1(t)$
- 7) Конец цикла,

то необходимо, чтобы функция $\text{RandomTrue}(P_4)$ всегда возвращала ложное значение, когда этот механизм используется для прогноза событий (отключить этот элемент на этапе прогноза).

Динамика запасов в такой модели со случайными дополнительными событиями будет, как на рисунке 5.15, а выборка событий представлена в таблице 5.2.



Источник: составлено автором.

Рисунок 5.15 – Динамика запасов в модели с дополнительными событиями

Таблица 5.2 – Данные для модели с дополнительными событиями

Дата t_i	Объём y_i	Дата t_i	Объём y_i	Дата t_i	Объём y_i
30.01.2020	3628	16.05.2021	3600	30.10.2022	3605
07.02.2020	401	25.07.2021	3618	22.01.2023	3600
17.04.2020	3612	06.12.2021	3615	06.06.2023	3611
10.09.2020	3638	02.03.2022	3648	29.06.2023	1231
20.11.2020	3621	25.03.2022	1236	20.09.2023	3616
22.12.2020	1487	05.08.2022	3602	-	-

Источник: составлено автором.

Исследователь видит только выборку событий и пытается предсказать будущее событие.

Для восстановления параметров механизма нужно предварительно сделать некоторые изменения в данных. Без этих изменений получается, что наблюдаемый объём покупки, например первый, был израсходован за короткий интервал времени (всего за 8 дней), из-за чего восстановление функции по интегралам будет произведено неверно. Если бы заранее знать, что второе событие является дополнительным, а не основным, то можно было бы понять, что за небольшой период времени – 8 дней – был израсходован небольшой второй объём. А до следующего события, например до третьего, уже бы расходовался большой объём, так как было произведено пополнение запасов до максимума. В результате достаточно провести простейшую

модификацию входных данных – поменять местами объём покупки дополнительного события с объёмом предыдущего события.

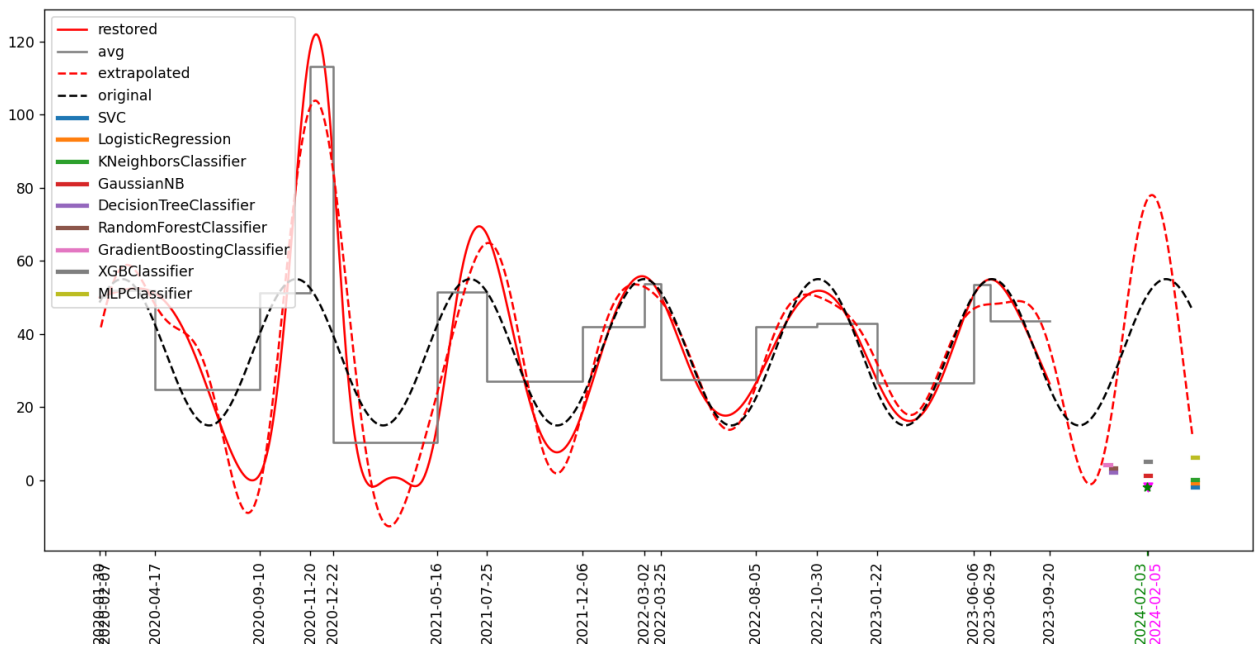
Примечание – Если бы в модели не было фиксированного максимума, когда клиент пополнял бы запасы на произвольную величину, то следовало бы объединить это дополнительное событие с предыдущим в одно событие, например $(t_1, y_1 + y_2)$.

Но заранее знать, какое из событий является дополнительным, а какое основным (вызвано механизмом), невозможно. Однако можно попробовать это предположить, перебрать все возможные комбинации, но их будет очень много (2^n). Можно поступить иначе. Например, относить все события, у которых объём покупок y_i меньше некоторой границы, к дополнительным. Воспользовавшись идеей, как в деревьях решений, выбирается граница разделения, наблюдения сортируются по величине объёма y_i и вводятся уровни разделения как среднее между двумя значениями. В результате надо будет проверить не более n уровней разделения (можно дополнительно ограничить количество уровней). В данном примере отсортированный ряд объёмов y_i выглядит следующим образом: 401, 1231, 1236, 1487, 3600, ... , 3648. Предполагая, что y_i дополнительного события не может превышать 90% максимального значения, имеем всего 5 границ разделения: 0, 816, 1233, 1361, 2554 (где 0 добавлен для того, чтобы все события относить к основным).

Для каждой выбранной границы разделения преобразуем набор данных (меняем местами объём дополнительного события с объёмом предыдущего события) и применяем предлагаемый метод, основанный на восстановлении параметров механизма образования событий. Затем определяем гиперпараметры в результате оптимизации прогноза нескольких последних событий и получаем оценку качества прогноза *Score* этих последних событий, формула (2.59) раздела 2.8. Эта оценка и будет являться индикатором того, насколько удачным оказался выбор границы разделения событий на дополнительные и основные. Останавливаем свой выбор на той границе, для которой оценка *Score* является наилучшей (наименьшей). Для этой границы

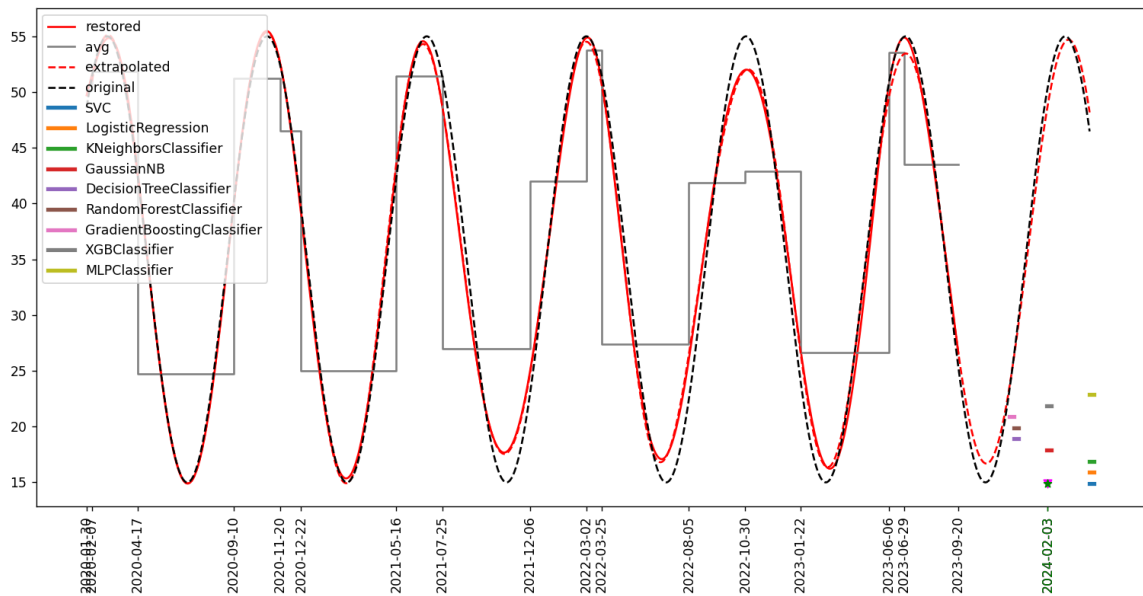
разделения преобразуем все исходные данные и получаем прогноз уже будущего события.

Заметим: при поиске гиперпараметров может произойти так называемое переобучение (*overfitting*). Например, при использовании всего двух последних событий поиск гиперпараметров останавливается на такой их комбинации, которая даёт хороший прогноз (*Score*) этих двух последних событий при неправильно выбранной границе разделения. Тем не менее в этом случае всё равно прогноз будущего события оказался очень близко к фактическому событию, рисунок 5.16. Отклонение составило всего 2 дня, и на графике это практически незаметно. Всё же желательно, чтобы переобучения не происходило. Для этого надо проводить поиск гиперпараметров по большему количеству событий. Например, уже при использовании трёх предыдущих событий наилучший уровень разбиения определился правильно. Прогноз будущего события полностью совпал с фактическим событием, рисунок 5.17.



Источник: составлено автором.

Рисунок 5.16 – Переобучение и неправильное разделение событий на дополнительные и основные, прогноз будущего события



Источник: составлено автором.

Рисунок 5.17 – Правильное разделение событий на дополнительные и основные, прогноз будущего события

Чем больше используется предыдущих событий при подборе гиперпараметров, тем меньше вероятность, что прогноз всех этих предыдущих событий может оказаться хорошим при неправильных входных данных. Однако увеличение количества предыдущих событий при поиске гиперпараметров, помимо увеличения объёма вычислений, приводит к тому, что обучающая выборка для предсказания этих событий сокращается (а это уменьшает способность предсказывать сами эти события).

Что касается методов классификации, то большинство из них неправильно прогнозируют будущее событие. Выше на рисунках 5.16 и 5.17 горизонтальными отрезками показаны интервалы, на которых прогнозируется появление события, ширина интервала 7 дней (при 14-ти и 3 днях картина похожая). Для метода дерева решений, градиентного бустинга и случайного леса погрешность составляет приблизительно половину интервала между событиями. Связано это, скорее всего, с тем, что из-за дополнительных событий не получится выявить никакой закономерности между признаками и временем до следующего события. В обучающей выборке не было нужной последовательности широких и узких интервалов. Лишь методы XGBClassifier и GaussianNB дали хороший прогноз (но при ширине интервала 3 дня метод

GaussianNB даёт очень плохой прогноз). Однако в предыдущем примере они показали себя плохо.

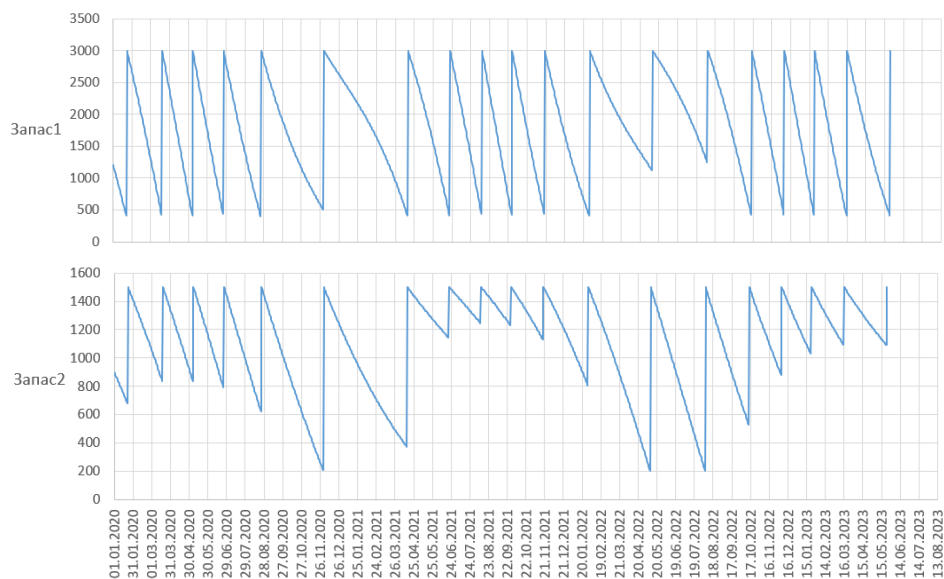
Модель с двумя продуктами

Изменяются запасы X_1 и X_2 уже двух продуктов. Каждый день из них вычитается спрос $P_1(t)$ и $P_2(t)$, разный для каждого продукта. У каждого продукта свой критический (P_3 и P_4) и свой максимальный уровень (P_5 и P_6) запасов. Событие возникает, когда запас хотя бы одного продукта опустился ниже критического уровня, происходит пополнение запасов обоих продуктов до своих максимумов. Модель механизма следующая.

- 1) Пока ($t \leq t_{end}$)
- 2) $X_1 = X_1 - P_1(t)$
- 3) $X_2 = X_2 - P_2(t)$
- 4) Если ($X_1 \leq P_3$ или $X_2 \leq P_4$), то
- 5) Создать событие ($t, P_5 - X_1, P_6 - X_2$)
- 6) $X_1 = P_5$
- 7) $X_2 = P_6$
- 8) Продвинуть время t и обновить параметр $P_1(t)$ и $P_2(t)$
- 9) Конец цикла.

Динамика запасов выглядит следующим образом, рисунок 5.18.

В результате имеется выборка наблюдений $(t_i, y_{1,i}, y_{2,i})$, таблица 5.3.



Источник: составлено автором.

Рисунок 5.18 – Динамика запасов двух продуктов

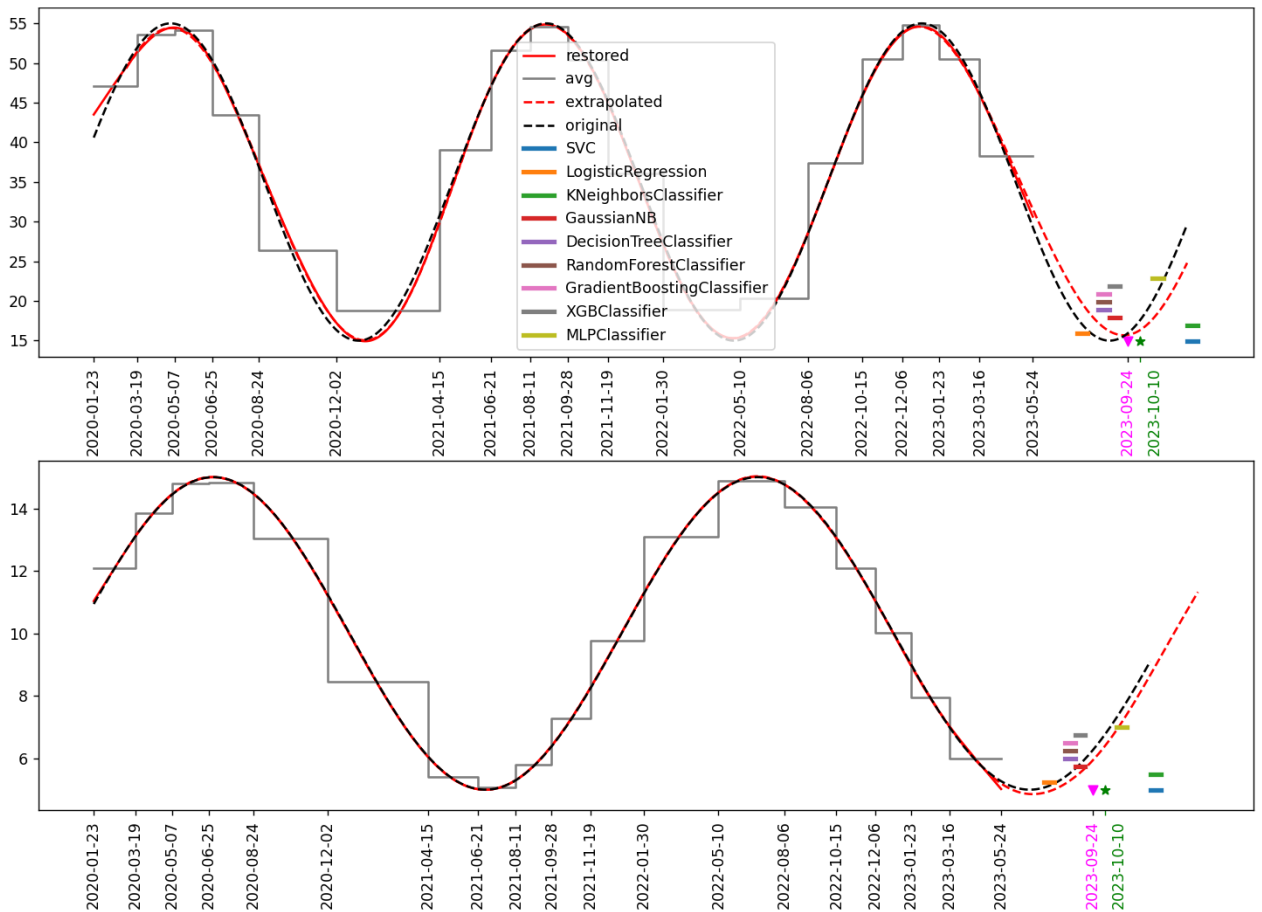
Таблица 5.3 – Данные модели с двумя продуктами

t_i	y_{1i}	y_{2i}	t_i	y_{1i}	y_{2i}	t_i	y_{1i}	y_{2i}
23.01.2020	2634	830	21.06.2021	2633	362	15.10.2022	2627	982
19.03.2020	2624	676	11.08.2021	2619	259	06.12.2022	2631	628
07.05.2020	2650	678	28.09.2021	2623	278	23.01.2023	2624	481
25.06.2020	2605	724	19.11.2021	2601	378	16.03.2023	2639	413
24.08.2020	2635	890	30.01.2022	2618	703	24.05.2023	2615	414
02.12.2020	2514	1303	10.05.2022	1893	1309	-	-	-
15.04.2021	2617	1133	06.08.2022	1786	1308	-	-	-

Источник: составлено автором.

Исследователь видит только эту выборку событий и пытается предсказать будущее событие.

Как и в предыдущем примере, необходимо сделать некоторые изменения в данных, подаваемых на вход метода. Надо определить, исчерпанием запаса какого из продуктов образовано событие. Поступаем похожим образом, как в предыдущем примере. Сортируем по возрастанию объёма первого продукта y_{1i} и вводим границы разделения как среднее значение между двумя соседними значениями. Тогда для выбранной границы разделения, события, у которых объём y_{1i} меньше этой границы, относим к исчерпанию второго продукта, а иначе первого. В итоге задача сводится к предыдущему примеру. Для каждого продукта имеется выборка основных и дополнительных событий. Меняем местами объём дополнительного события с объёмом предыдущего события (имеем события $(t_{i-1}, y_{1,i}, y_{2,i-1})$ и $(t_i, y_{1,i-1}, y_{2,i})$). Единственная трудность: объём самого первого события не на что менять, для упрощения меняем его на самое большое значение во всей выборке. Далее происходит восстановление функции по интегралам и прогноз события. Как и в предыдущем примере, выбирается такая граница разделения, которая позволяет лучше всего предсказать два (лучше три) предыдущих события (индивидуально по каждому продукту). И уже с этой границей происходит предсказание будущего события как момент исчерпания запаса одного из двух продуктов. Результат применения описанного метода показан на рисунке 5.19.



Источник: составлено автором.

Рисунок 5.19 – Восстановление скорости расхода запаса двух продуктов после преобразования входных данных, экстраполяция и прогноз будущего события

В модели методов классификации количество признаков расширяется до 24. Добавляются соответствующие значения для второго продукта: U_k , U_{k-1} , U_{k-2} , U_{k-3} , \bar{U} , v_{k-1} , v_{k-2} , v_{k-3} , \bar{v} . Тем не менее методы классификации плохо справляются с предсказанием будущего события. На рисунке 5.19 горизонтальными линиями указаны интервалы шириной 14 дней, на которых методы классификации предсказывают появление события (при интервалах 3 дня и 7 дней картина очень похожая). Из методов классификации лучше всего прогноз оказался у градиентного бустинга и GaussianNB.

Модель с изменяющимся максимумом

Пусть покупатель изменяет максимальный объём покупки $X_2 = P_3$ в зависимости от того, как быстро у него израсходовался предыдущий запас. Если X_1 закончился быстрее P_4 дней (50), то максимальный запас увеличивается в 2 раза. Если запас закончился позже, чем P_5 дней (90), то

максимальный запас уменьшается в 2 раза. Иначе максимальный запас не меняется (закончился в промежутке от 50 до 90 дней). Критический уровень запасов P_2 единиц (200). Данные модели представлены в таблице 5.4. Модель механизма может быть записана следующим образом.

- 1) $X_2 = P_3$
- 2) Пока ($t \leq t_{end}$)
- 3) $X_1 = X_1 - P_1(t)$
- 4) Если ($X_1 \leq P_2$), то
- 5) Если ($t - t_n < P_4$), то
- 6) $X_2 = X_2 \cdot 2$
- 7) Если ($t - t_n > P_5$), то
- 8) $X_2 = X_2/2$
- 9) Создать событие ($t, X_2 - X_1$)
- 10) $X_1 = P_3$
- 11) Продвинуть время t и обновить параметр $P_1(t)$
- 12) Конец цикла

Таблица 5.4 – Данные модели с изменяющимся максимумом

Дата t_i	Объём y_i	Дата t_i	Объём y_i	Дата t_i	Объём y_i
21.01.2020	2312	14.04.2021	2336	28.05.2022	4840
07.04.2020	2334	22.05.2021	4800	02.08.2022	4844
17.05.2020	4813	27.07.2021	4831	06.11.2022	2313
22.07.2020	4805	23.10.2021	4816	-	-
13.10.2020	4819	21.04.2022	2351	-	-

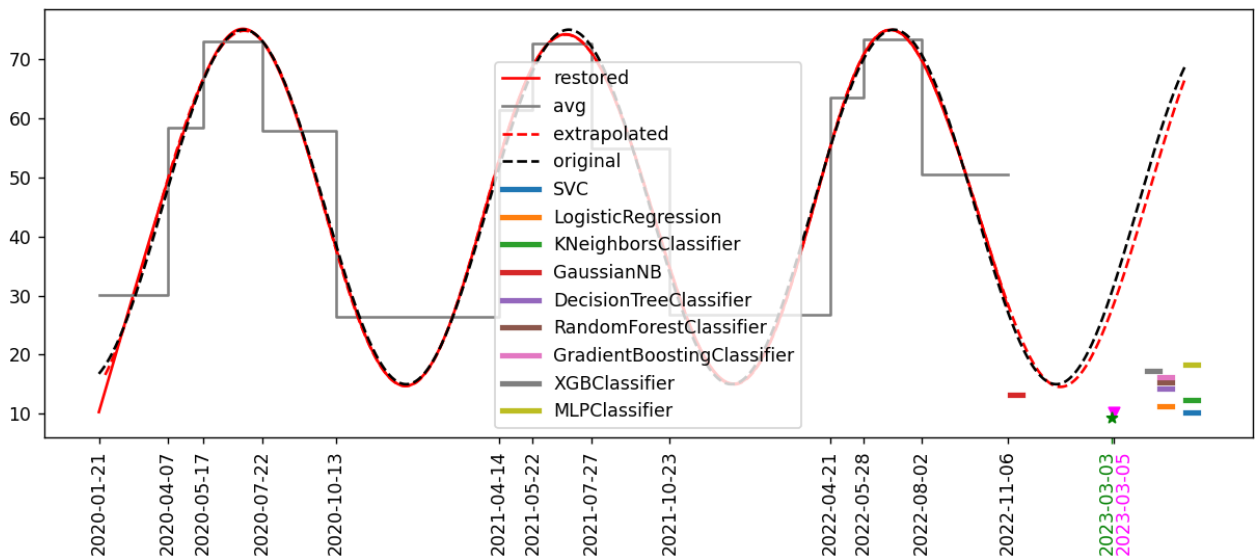
Источник: составлено автором.

Исследователь наблюдает лишь эти данные и имеет предположение о механизме образования событий, но не знает всех его параметров. Восстановление функции скорости расхода запаса по интегралам и её экстраполяция происходит, как раньше. Если требуется дать прогноз не только моменту появления будущего события t_{n+1} , но и изменяющейся величине y_{n+1} , то надо определить дополнительные параметры.

Из данных можем увидеть, что повышение объёма покупок происходило, когда между покупками прошло 40, 38 и 37 дней, максимальное среди них 40. Сокращение объёма происходило при интервалах 183, 180 и 96 дней, минимальное среди которых 96. Далее рассматриваем

скорректированные объёмы покупок $y_i - 0,5\hat{f}(t_i)$ (корректировка нужна, так как часть объёма откладывается для восполнения страховых запасов и не расходуется для компенсации использования страховых запасов). Выделяем две группы покупок со средним скорректированным объёмом 2310 и 4790. Тогда на основе уравнения $2(P_2 + 2310) = P_2 + 4790$ определяем критический уровень запасов $P_2 = 170,5$. Значения 40; 96; 170,5 приблизительно соответствуют закладываемым параметрам (50; 90; 200), для более точной идентификации требовался бы больший объём выборки (у нас имелось всего 13 наблюдений).

Прогноз будущего события показан на рисунке 5.20. Прогноз методами классификации при интервале в 14 дней показан горизонтальными отрезками (при интервалах 7 дней и 3 дня картина ещё хуже). Методы классификации дают очень плохой прогноз.



Источник: составлено автором.

Рисунок 5.20 – Прогноз будущего события в модели с изменяющимся максимумом

В этом примере также интересен прогноз объёмов покупок y_i , получим прогноз на три будущих события. Так как интервал между предыдущими событиями меньше или равен полученному значению 96, то при прогнозе первого события имеем корректное уменьшение запаса в 2 раза. Для прогноза второго события интервал оказывается, наоборот, очень маленьким, и прогнозируется увеличение запаса в 2 раза. Для третьего события интервал

между событиями прогнозируется 43 дня, когда параметр раннего времени наступления события был определён как 40 дней. Поэтому максимальный запас не меняется, хотя фактически в модели он поменялся, так как ранее время наступления события закладывалось 50 дней. Результаты показаны в таблице 5.5. Прогноз четвёртого события оказывается уже ошибочным. В этом примере сразу осуществляется прогноз на несколько событий вперёд на одних и тех же входных данных (не добавляли фактические данные новых событий). Заметим: методы классификации не в состоянии спрогнозировать значения y_i .

Таблица 5.5 – Прогноз трёх будущих событий

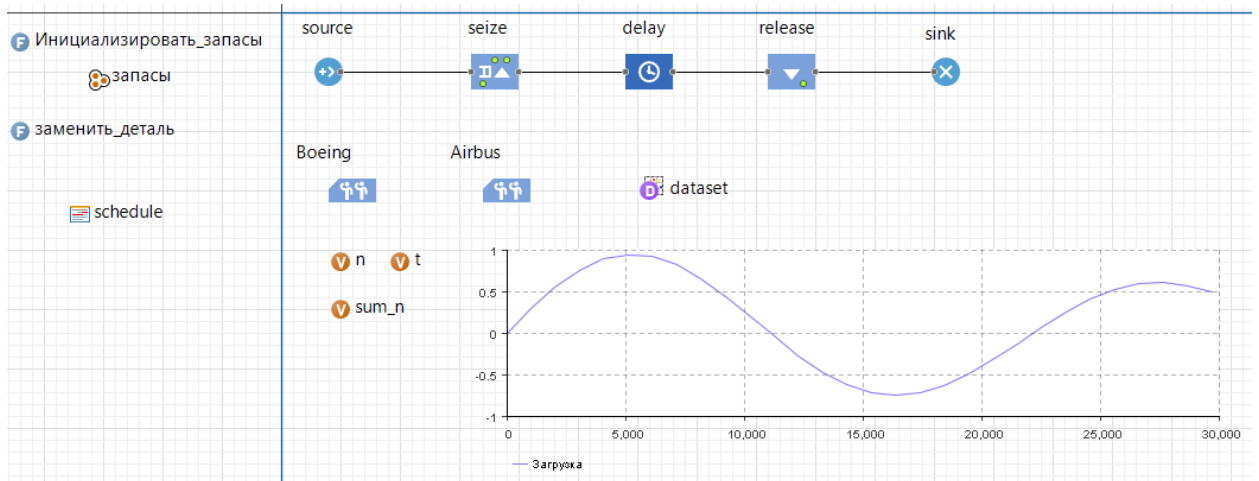
Факт t_i	Прогноз t_i	Факт y_i	Прогноз y_i
03.03.2023	05.03.2023	1075	1084
31.03.2023	04.04.2023	2301	2356
12.05.2023	17.05.2023	4811	2371

Источник: составлено автором.

Модель поставки запчастей для авиатранспорта

Описание модели. Данная модель создана в Anylogic, рисунок 5.21.

Интенсивность полётных заданий в блоке source задана расписанием в виде ступенчатой функции, таблица 5.6. Единица модельного времени – час.



Источник: составлено автором.

Рисунок 5.21 – Модель потребности в запчастях парком самолётов

Таблица 5.6 – Интенсивность полётных заданий (заданная расписанием)

Начало интервала	Конец интервала	Интенсивность в час
0	10 000	1
10 000	20 000	2
20 000	30 000	1

Источник: составлено автором.

Полётные задания задействуют один из свободных самолётов Boeing или Airbus. У каждого самолёта есть свой набор деталей. Каждая деталь имеет ограничение по налёту, после превышения которого следует заменить эту деталь. Также с определённой вероятностью происходят случайные поломки деталей. Пусть набор деталей Boeing и их характеристики заданы, как в таблице 5.7. Для Airbus также имеется свой набор деталей со своими характеристиками.

Таблица 5.7 – Предполагаемые характеристики деталей Boeing

Название	Тип	Количество	Ограничение по налёту, час	Вероятность поломки
Деталь 1	101	2	100	0.001
Деталь 2	102	2	150	0.001
Деталь 3	103	1	100	0.001

Источник: составлено автором.

Пусть каждый самолёт проводит в воздухе от 1 до 10 часов по треугольному закону с наиболее вероятным значением 5 часов. Проведённое в воздухе время увеличивает суммарное время налёта каждой детали самолёта. После выполнения полётного задания происходит проверка деталей самолёта. Если превышено максимальное время налёта, деталь заменяется, для упрощения мгновенно. Также при проверке с заданной вероятностью деталь может оказаться неисправной, и она также заменяется.

При замене каждой детали уменьшается запас соответствующего типа запчастей. При этом фиксируется количество деталей, замена которых происходит из-за случайных поломок. Когда запас запчастей конкретной детали заканчивается (опускается ниже некоторого критического уровня), происходит пополнение (заказ на пополнение) до фиксированного максимума.

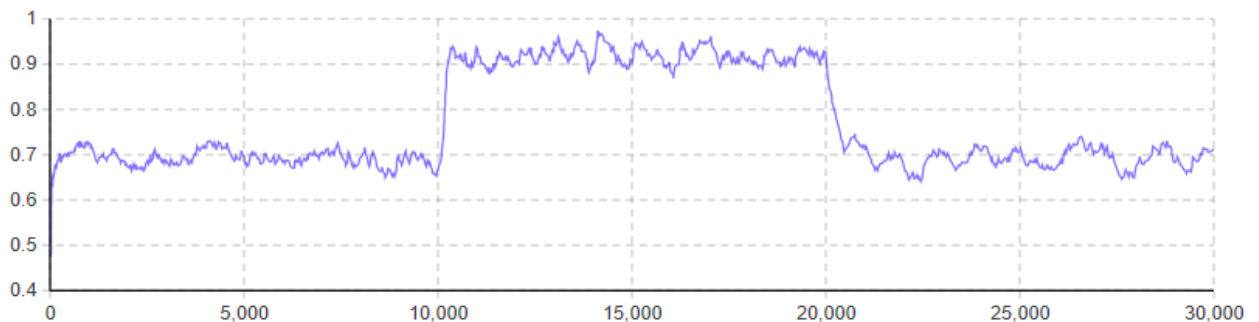
В результате моделирования получается выборка событий пополнения запаса запчастей каждой из деталей (предполагается, что запас каждой детали пополняется независимо), пример – таблица 5.8. Выводится суммарное количество случайных поломок каждой детали. Строится временной график коэффициента загрузки самолётов, например самолёта Boeing, рисунок

5.22 (загруженность блока ResourcePool не годится, так как эта загруженность подсчитывается и усредняется за всё время моделирования).

Таблица 5.8 – Пример собранных данных

Событие	Самолёт	Название	Тип	Дата	Количество
Пополнение запаса	Boeing	Деталь 1	101	07.01.2020	50
Пополнение запаса	Boeing	Деталь 3	103	07.01.2020	50
Пополнение запаса	Boeing	Деталь 2	102	10.01.2020	50
Пополнение запаса	Airbus	Деталь 1	201	26.01.2020	100
Пополнение запаса	Boeing	Деталь 1	101	30.01.2020	50
Пополнение запаса	Airbus	Деталь 2	202	03.02.2020	50
Пополнение запаса	Boeing	Деталь 2	102	15.02.2020	50
Пополнение запаса	Airbus	Деталь 3	203	19.02.2020	150
Пополнение запаса	Boeing	Деталь 1	101	25.02.2020	50
...
Пополнение запаса	Boeing	Деталь 1	101	22.03.2023	50
Пополнение запаса	Airbus	Деталь 2	202	04.04.2023	50
Пополнение запаса	Boeing	Деталь 2	102	04.04.2023	50
Пополнение запаса	Boeing	Деталь 3	103	14.04.2023	50
Пополнение запаса	Boeing	Деталь 1	101	17.04.2023	50
Пополнение запаса	Boeing	Деталь 1	101	12.05.2023	50
Пополнение запаса	Boeing	Деталь 2	102	13.05.2023	50
Пополнение запаса	Airbus	Деталь 1	201	03.06.2023	100

Источник: составлено автором.



Источник: составлено автором.

Рисунок 5.22 – Коэффициент загруженности самолётов Boeing

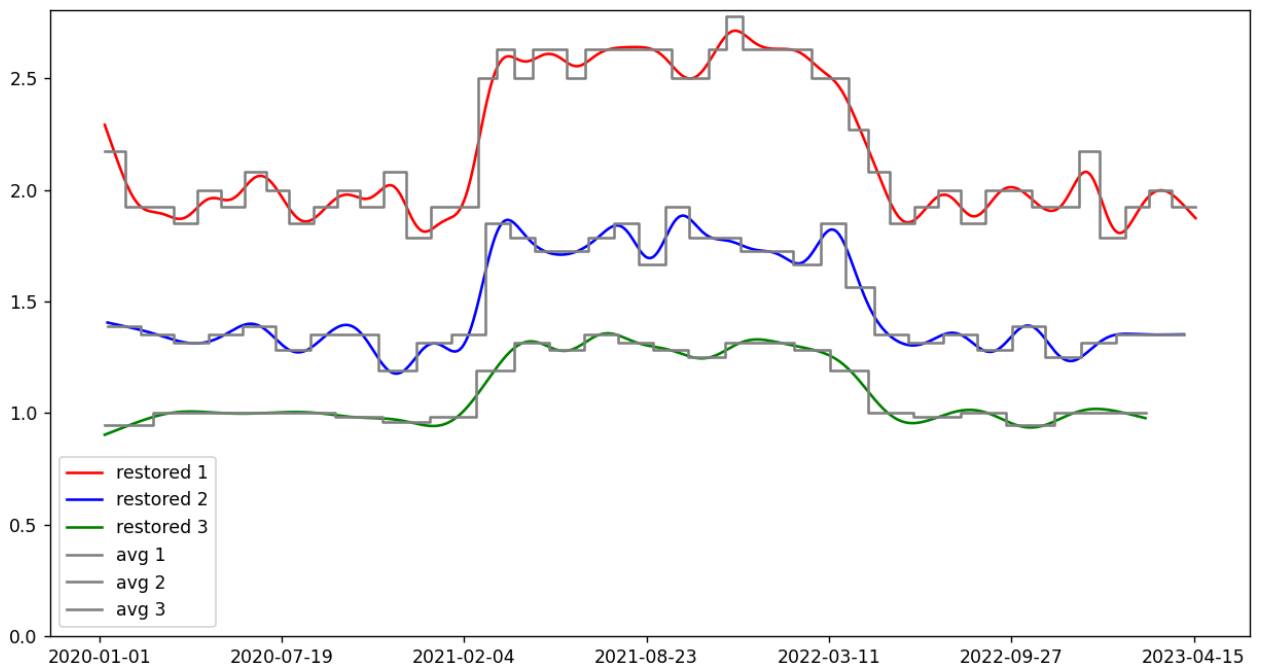
Применение метода. На вход метода подаются лишь данные о пополнении запасов запчастей самолёта, например, для самолёта Boeing данные имеют вид, как в таблице 5.9, где указаны время и объём пополнения запаса каждой из трёх деталей (для упрощения объём одинаковый).

Таблица 5.9 – Данные пополнения запаса запчастей, полученные в результате моделирования

t_1	y_1	t_2	y_2	t_3	y_3
07.01.2020	50	10.01.2020	50	07.01.2020	50
30.01.2020	50	15.02.2020	50	29.02.2020	50
25.02.2020	50	23.03.2020	50	19.04.2020	50
22.03.2020	50	30.04.2020	50	08.06.2020	50
18.04.2020	50	06.06.2020	50	28.07.2020	50
13.05.2020	50	12.07.2020	50	16.09.2020	50
...
22.03.2023	50	26.02.2023	50	02.01.2023	50
17.04.2023	50	04.04.2023	50	21.02.2023	50
12.05.2023	50	13.05.2023	50	14.04.2023	50

Источник: составлено автором.

Из этих данных можно восстановить интенсивность расхода запчастей в сутки, рисунок 5.23.

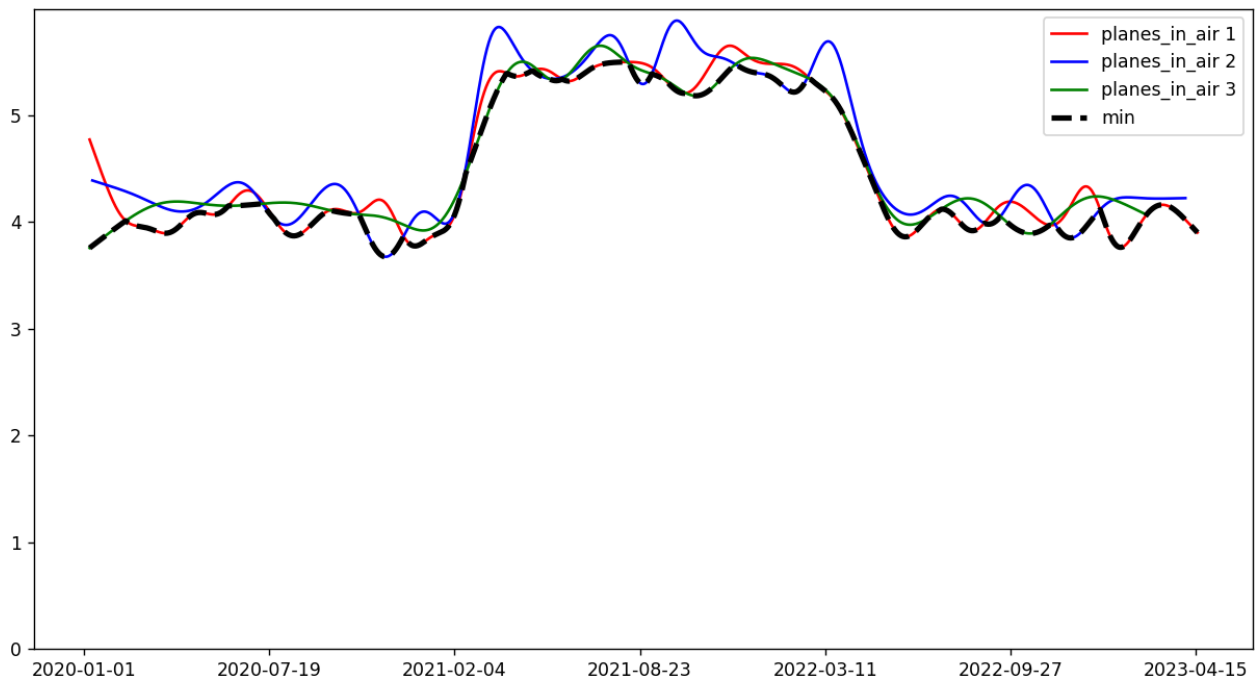


Источник: составлено автором.

Рисунок 5.23 – Интенсивность расхода запчастей в сутки

Но если известны такие характеристики деталей, как максимальное время налёта и сколько таких деталей установлено в самолёт, то у нас получается определить среднее количество самолётов в воздухе. Например, если интенсивность замены составляет 2,5 детали в час и известно, что максимальное время налёта для этой детали составляет, например, 100 часов, при этом у самолёта установлено две таких детали (изнашиваются одновременно обе), то среднее количество самолётов в воздухе определяется

как $2,5 \times 100/24/2 = 5,208$. Такие вычисления можно проделать для каждой детали самолёта, рисунок 5.24. Заметим: если бы, наоборот, было известно среднее количество самолётов в воздухе, то можно было бы определить максимальное время налёта для каждой детали. Так как эти зависимости показывают количество одних и тех же самолётов, то логично взять минимальное значение (нижнюю огибающую).

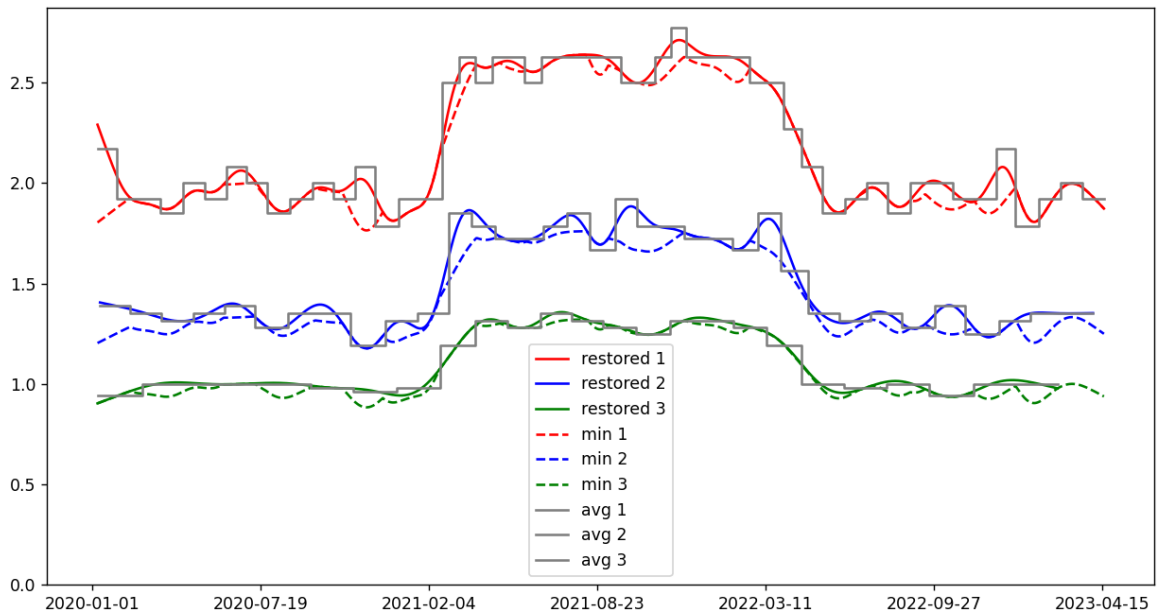


Источник: составлено автором.

Рисунок 5.24 – Среднее количество самолётов в воздухе, определённое по пополнению запасов каждой из деталей этого самолёта

По этой нижней огибающей количества самолётов в воздухе можно посчитать интенсивность расхода запчастей исключительно по налёту, рисунок 5.25. Тогда разница между двумя интенсивностями приблизительно будет показывать интенсивность случайных поломок деталей. Площадь между сплошными и пунктирными линиями будет показывать среднее количество случайных поломок. В результате имеем среднее количество случайных поломок 45,6; 67,5 и 27,4, детали 1, 2 и 3 соответственно (фактически при моделировании вышло 46; 62 и 28).

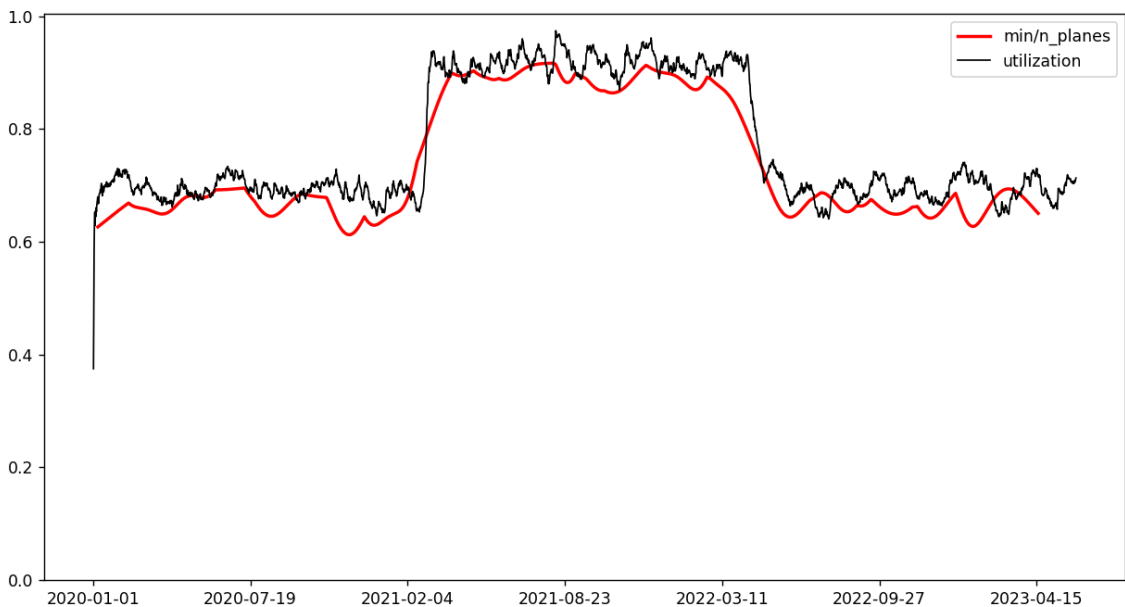
Примечание – При большом коэффициенте сглаживания, когда весь шум устраняется, разница между двумя интенсивностями будет намного меньше, количество случайных поломок также будет значительно меньше.



Источник: составлено автором.

Рисунок 5.25 – Интенсивность расхода запчастей, пунктирная линия – по налёту, сплошная линия – общая интенсивность, шт. в сутки. Разница между ними даёт интенсивность случайных поломок деталей

Если дополнительно знать число самолётов, то можно получить коэффициент загрузки этих самолётов, поделив значения нижней границы числа самолётов в воздухе на их фактическое число, рисунок 5.26.



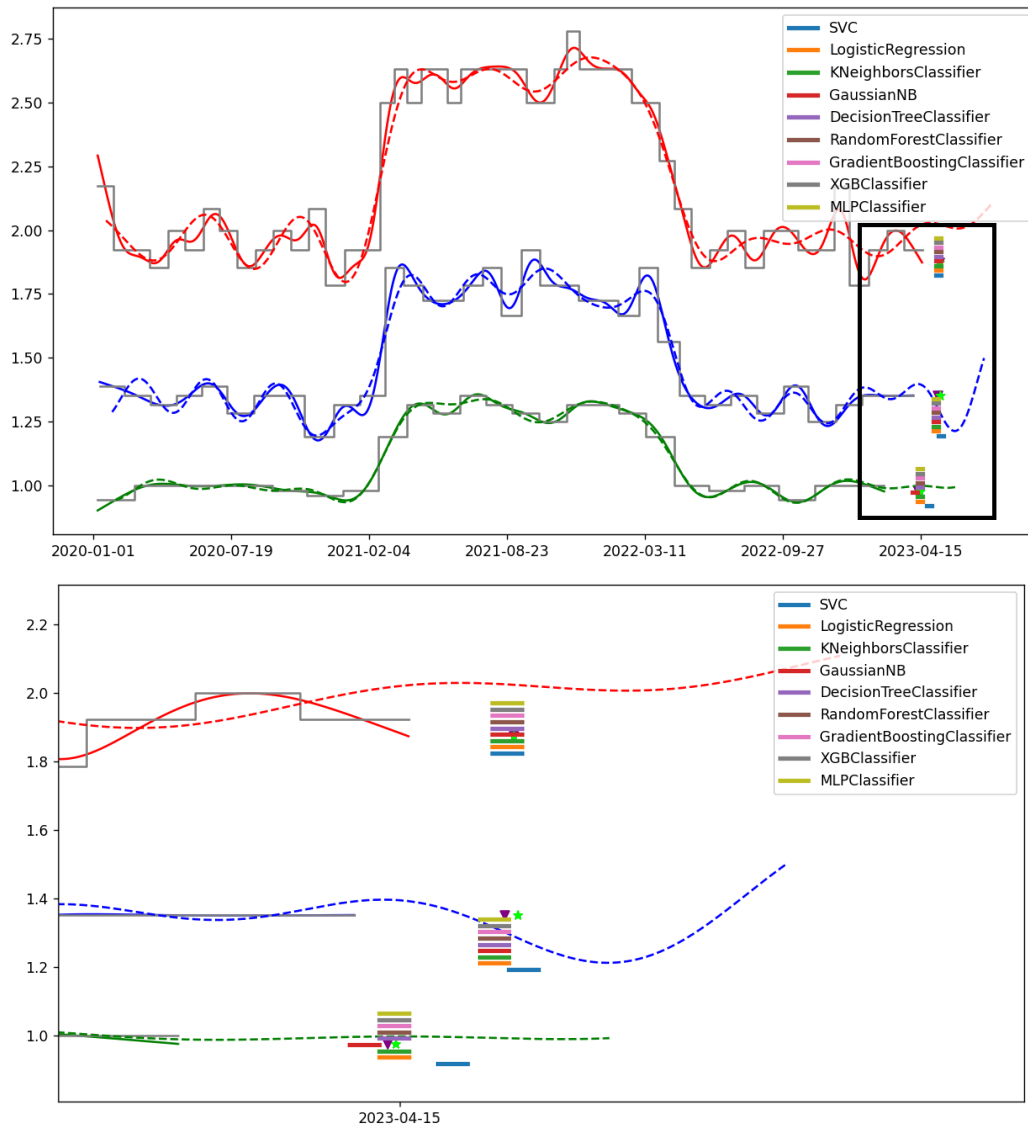
Источник: составлено автором.

Рисунок 5.26 – Коэффициент загрузки самолётов

Всё это получилось определить лишь из данных пополнения запчастей, таблица 5.9. Для Airbus можно аналогично получить похожие результаты.

Что касается прогноза очередного пополнения запаса запчастей, то почти все методы дают хороший прогноз, рисунок 5.27 (ширина интервала в

методах классификации равна 7 дням, при ширине 14 дней и 3 дня картина чуть хуже). Это можно связать с тем, что имеется достаточно много наблюдений, в которых присутствует похожее чередование наблюдений («ступенек»).



Источник: составлено автором.

Рисунок 5.27 – Прогноз следующего пополнения запаса запчастей

Этот пример показывает, сколько можно извлечь информации из дискретных событий, если предполагать модель механизма их образования. Методами машинного обучения не получится определить интенсивность расхода запчастей или интенсивность случайных поломок.

Поставка биологически активных добавок в аптечные пункты

Имеются реальные данные о поставках российского представительства компании ООО «Квайссер Фарма» биологически активных добавок. Ограничимся самым ходовым товаром, Doppelherz Aktiv Омега-3, который можно видеть на полках аптек по всей России. Такой препарат выпускается в разных упаковках, содержащих от 30 до 120 пилюль, рисунок 5.28. Рассмотрим данные о поставках препарата с наименованием «Doppelherz Aktiv Омега-3 капс. № 30» в одну из аптек Северной Осетии – Алании. Данные представлены в таблице 5.10.



Источник: составлено автором.

Рисунок 5.28 – Прогноз следующего пополнения запаса

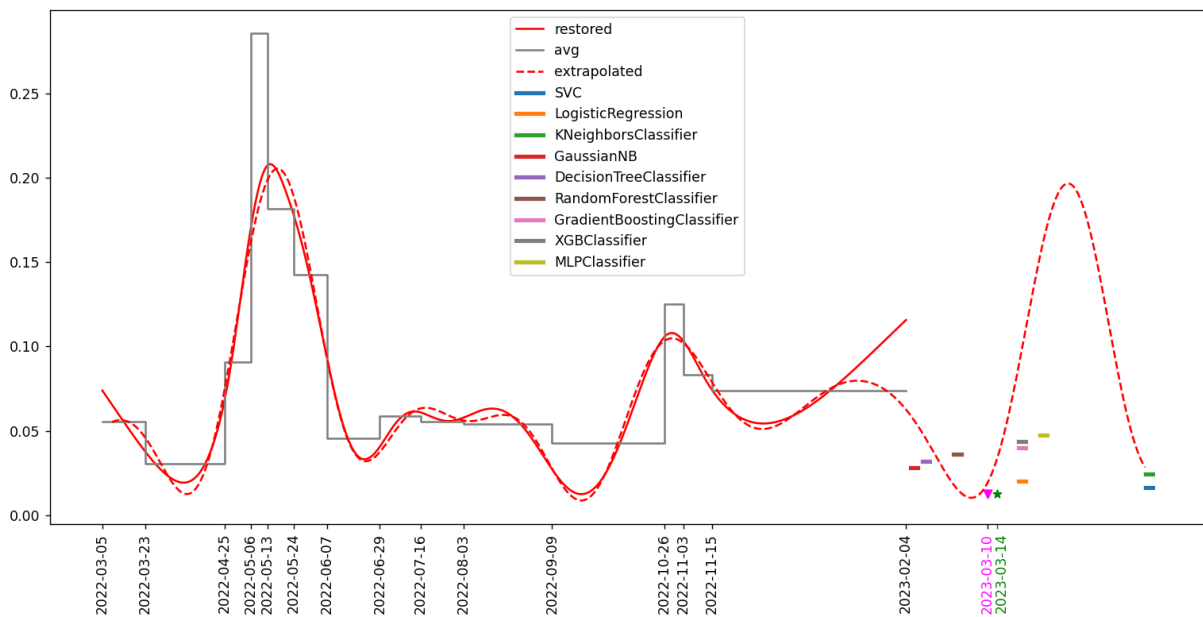
Таблица 5.10 – Данные о поставках товара «Doppelherz Aktiv Омега-3 в капсулах № 30» в аптечный пункт в Северной Осетии – Алании

Наименование	Дата	Количество, шт.
Дoppelherz Aktiv Омега-3 капс. № 30	05.03.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	23.03.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	25.04.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	06.05.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	06.05.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	13.05.2022	2
Дoppelherz Aktiv Омега-3 капс. № 30	24.05.2022	2
Дoppelherz Aktiv Омега-3 капс. № 30	07.06.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	29.06.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	16.07.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	03.08.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	03.08.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	09.09.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	13.09.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	26.10.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	03.11.2022	1
Дoppelherz Aktiv Омега-3 капс. № 30	15.11.2022	1
Дoppelherz Aktiv Омега-3 капс. №30 (Акция 4+1)	10.01.2023	5
Дoppelherz Aktiv Омега-3 капс. № 30	04.02.2023	1
Дoppelherz Aktiv Омега-3 капс. № 30	14.03.2023	1

Источник: составлено автором.

Данные содержат несколько записей, относящихся к одному и тому же дню. Например, есть две записи с датой 06.05.2022 и две записи с датой 03.08.2022. Объединяем события с одной датой в одно событие, так количество поставленной продукции станет в эти дни равным 2 шт. Также предполагаем, что приобретение товара по акции не связано с событием исчерпания запаса (акция была привлекательной, аптечный пункт совершил дополнительную поставку, когда запас ещё имелся), объединим это событие с предшествующим, тогда с даты 15.11.2022 расходуется 6 единиц продукции.

Прогноз будущего события показан на рисунке 5.29. Интересно, что разные методы классификации дают совершенно разный прогноз (ширина интервалов выбрана 3 дня, при 7-ми и 14 днях картина отличается незначительно). Методы классификации не смогли отыскать правильную закономерность в имеющихся данных. Скорее всего, это произошло из-за того, что в данных отсутствовало нужное чередование наблюдений.



Источник: составлено автором.

Рисунок 5.29 – Прогноз следующего пополнения запаса

В работе [215] представлено 12 дополнительных примеров на реальных данных прогнозирования поставок биологически активных добавок в аптечные пункты разных регионов России, а также разобрано три неудачных примера применения метода и объяснены причины произошедшему.

Пример из сферы финансов, займы/облигации – Московская область

Имеются данные о займах (выпусках облигаций) муниципальным образованием Московская область, таблица 5.11. Данные о займах собраны с помощью информационной системы Thomson Reuters.

Таблица 5.11 – Данные о выпуске облигаций муниципальным образованием Московская область

Дата	Количество, млрд
19.11.2002	1,9
10.12.2002	1
19.08.2003	4
27.04.2004	9,6
05.04.2005	12
25.04.2006	12
25.04.2007	16
24.12.2007	7
17.01.2008	19
11.12.2008	33
разрыв ~8 лет	
29.11.2016	25
28.12.2017	12,5

Источник: система Thomson Reuters Eikon (MOSCOW CITY; Debt & Credit; Retrieved 14-Feb-2019).

В данных после 2008 года на протяжении около восьми лет отсутствуют записи. Либо муниципальное образование не могло выпускать облигации по причине кризиса 2008 года, либо данные не собирались в этот период. После этого периода имеется совсем незначительное количество наблюдений. Поэтому будем рассматривать данные с 2002-го по 2008 год.

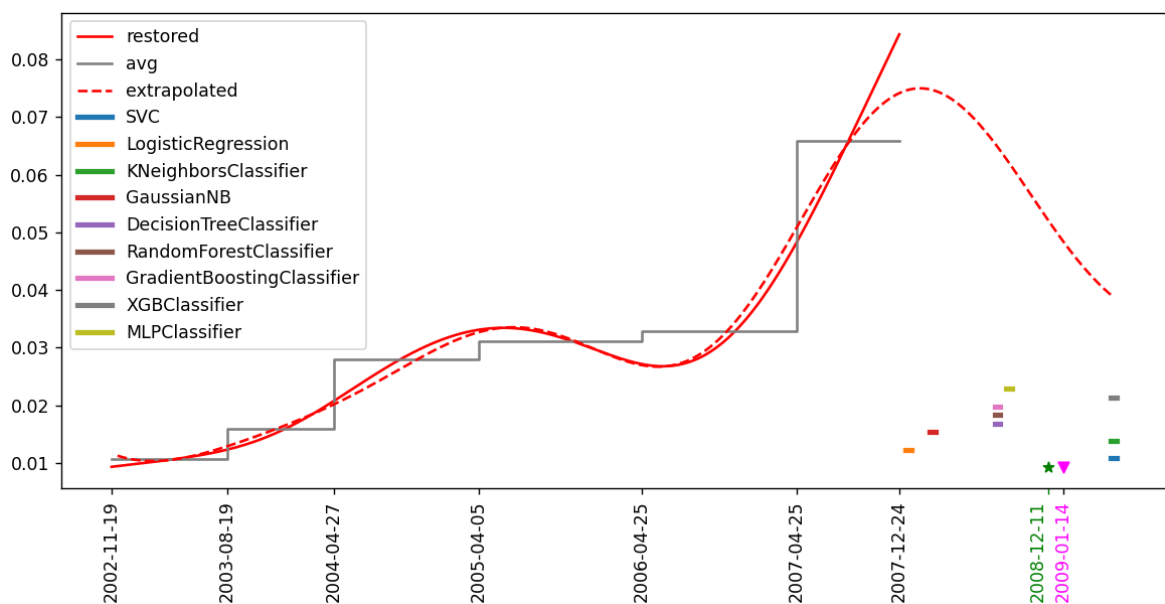
Также данные содержат несколько записей, которые идут через достаточно малый промежуток времени. Например, есть две записи с датами 19.11.2002 и 10.12.2002, между которыми прошло меньше месяца. Такие события объединим в одно. Аналогично поступим с записями 24.12.2007 и 17.01.2008 по той же причине.

Используя предположение, что муниципалитет выпускает облигации (берёт займ) тогда, когда ему не хватает собственных средств (закончились средства от предыдущего займа), можем восстановить скорость (интенсивность) расхода средств (также можно интерпретировать как результирующую функцию между доходом и расходом муниципалитета).

Сейчас нас интересует способность предсказывать события, связанные именно с займами. Конечно же, предсказывать сам кризис также очень хотелось бы научиться, но для этого требуется строить совсем другие модели по совсем другим данным. Различные модели финансовых кризисов на фондовых рынках рассматриваются в работе [181], однако это тема совсем другого исследования.

Сделаем прогноз последнего события 2008 года (до разрыва в данных), рисунок 5.30. Прогноз последнего события даёт дату 14.01.2009, когда реальное последнее событие было 11.12.2008. Прогноз оказался на 33 дня позже. Заметим, что интервал времени между последними событиями составляет 353 дня, т.е. ошибка прогноза составила 9,3% от интервала времени между событиями. Методы классификации при ширине интервала в 14 дней дают очень плохой прогноз. При других интервалах (7, 3, 21, 28) один или два метода начинают давать хороший прогноз, но при смене на другой интервал тот метод, что давал хороший прогноз, начинает давать очень плохой прогноз, и наоборот, никакого постоянства не обнаруживается.

Примечание – В предыдущей работе [215] ещё не был реализован подбор гиперпараметров на сетке, гиперпараметры выбирались вручную, тогда прогноз оказался даже точнее на три дня. В работе [215] рассматривается ещё несколько примеров выпуска облигаций муниципальными образованиями и коммерческими организациями



Источник: составлено автором.

Рисунок 5.30 – Прогноз следующего выпуска облигаций

Прогнозирование будущих покупок многими клиентами в онлайн-магазине

На Kaggle есть интересный пример [182] на реальных данных. Задача исследователя (Нагеш Сингх Чаухан) заключалась в том, чтобы определить, будет ли клиент совершать покупку в следующий квартал (90 дней). В анализируемом наборе данных почти миллион транзакций прошлых покупок, содержащих идентификатор клиента, дату, идентификатор товара, количество товара, его цену и другие параметры, рисунок 5.31.

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom
...
944463	575312	22083	PAPER CHAIN KIT RETROSPOT	6	2011-11-09 12:49:00	2.95	13588.0	United Kingdom
944464	575312	23355	HOT WATER BOTTLE KEEP CALM	4	2011-11-09 12:49:00	4.95	13588.0	United Kingdom
944465	575312	22110	BIRD HOUSE HOT WATER BOTTLE	6	2011-11-09 12:49:00	2.55	13588.0	United Kingdom
944466	575312	22037	ROBOT BIRTHDAY CARD	12	2011-11-09 12:49:00	0.42	13588.0	United Kingdom
944467	575312	21790	VINTAGE SNAP CARDS	12	2011-11-09 12:49:00	0.85	13588.0	United Kingdom

Источник: составлено автором по материалам [182].

Рисунок 5.31 – Набор данных транзакций покупок в онлайн-магазине

Исследователь (Нагеш) решает задачу методами классификации и строит классификационную модель, чтобы предсказать, совершит ли клиент онлайн-магазина покупку в следующем периоде. Обучающая матрица состоит из искусственно сконструированных признаков: CustomerID, Recency, RecencyCluster, Frequency, FrequencyCluster, Revenue, RevenueCluster, OverallScore, Segment. После добавления фиктивных переменных для качественного признака Segment обучающая матрица X будет выглядеть, как на рисунке 5.32. Нагешу удалось получить крайне высокую точность (accuracy) 0,92 и площадь под ROC-кривой 0,88. Что может быть лучше, это же выдающийся результат, если не целый ряд допущенных ошибок.

	CustomerID	Recency	RecencyCluster	Frequency	FrequencyCluster	Revenue	RevenueCluster	OverallScore	Segment_High-Value	Segment_Low-Value	Segment_Mid-Value
0	13085.0	57	3	92	3	1459.46	0	6	False	False	True
1	18087.0	44	3	95	3	14411.62	0	6	False	False	True
2	17519.0	33	3	190	3	4411.48	0	6	False	False	True
3	12362.0	12	3	242	3	4804.39	0	6	False	False	True
4	15712.0	9	3	156	3	3261.01	0	6	False	False	True
...

Источник: составлено автором по материалам [182].

Рисунок 5.32 – Вид обучающей матрицы X

1) Самая большая ошибка – это то, как формируется эндогенная переменная Y (NextPurchaseDayRange). Для этого по каждому клиенту определяется самая последняя покупка до начала следующего квартала и самая первая покупка в следующем квартале. Потом рассчитывается разница в днях между этими двумя покупками (если покупки в следующем квартале не было, то ставится 9999). И если эта разница превышает 90 дней, то объясняемая переменная принимает значение 0 (независимо от того, будет в следующем квартале событие или нет), иначе 1. Где здесь подвох? А он в том, что задействуется часть обучающей выборки при формировании объясняемой переменной. В модели как раз используется признак Recency, который показывает, сколько прошло дней от последней покупки обучающего периода до начала следующего квартала. Получается, что есть признак Recency, отсчитывающий количество дней до начала тестового периода, и одновременно с этим формируется Y , отсчитывая количество дней до первого события тестового периода. Естественно, если Recency будет превышать 90 дней, то Y примет значение 0. По-другому можно сказать, что та переменная Y , которая объясняется моделью, на самом деле конструируется по признаку Recency. Это очень серьёзная ошибка, метрики качества модели получаются очень высокими именно из-за этого. Если же исправить эту досадную ошибку и формировать Y правильно, проверяя, попало событие в следующий квартал или нет, то ассигасу падает до 0,76, а площадь под ROC-кривой до 0,73.

Примечание – Другим способом исправления ошибки является исключение признака `Recency`, а также признака `RecencyCluster` и корректировка признаков `OverallScore` и `Segment`, которые формируются из суммы кластеров `Recency`, `Frequency`, `Revenue` (исключив влияние `Recency` на формирование `OverallScore` и `Segment`). В результате `accracy` падает до 0,74, а площадь под ROC-кривой до 0,705. Заметим, что в этом случае получается задача предсказания события в следующем квартале от последнего события каждого клиента, а не в будущем квартале от заданной даты, что нелогично с практической точки зрения.

2) Другой ошибкой является описанная в первой главе неправильная валидация, когда набор данных разбивается на обучающую и контролирующую выборку, относящуюся к одному и тому же периоду времени. Выборка делится на обучающую и контролирующую в пропорциях 80 и 20%, причём по клиентам (по вертикали), как на рисунке 1.11 в первой главе. То есть метрика качества показывает, насколько модель, обученная на данных 80% клиентов, способна объяснить совершение покупки в том же квартале для оставшихся 20% клиентов. Не оценивается способность предсказывать события в следующем квартале. Чтобы исправить эту ошибку, были внесены значительные изменения в открытый программный код на Kaggle. Был реализован правильный способ валидации. Надо правильно организовать разбиение выборки на обучающую и контролирующую по времени, без последнего квартала и с последним кварталом, как это показано на рисунке 1.8 в первой главе. Обучающая матрица X сформирована по событиям до предпоследнего квартала, а обучающий столбец Y сформирован по событиям этого предпоследнего квартала. Контролирующая матрица X сформирована по событиям до последнего квартала, а контролирующий столбец Y сформирован по событиям самого последнего квартала. Причём задействуются все 100% наблюдений (клиентов, а не 80 на 20%). Наконец, обучив модель по обучающим X и Y , строится прогноз по контролирующей матрице X , который сверяется с контролирующим столбцом Y . В результате самая главная метрика площади под ROC-кривой падает до 0,71 (`accracy` остаётся 0,76). Эти результаты расчёта метрики уже корректны. Такой способ валидации единственный правильный, так как перед нами задача предсказания

события в будущем, а не задача предсказания события от другого клиента в этом же периоде (в настоящем).

3) Однако одним из вопросов остаётся, насколько справедливо использовать интервал шириной 90 дней. В обучающем наборе есть клиенты, которые совершают покупки чуть ли не каждые 3 дня или 7 дней. Насколько обосновано брать такой широкий интервал, чтобы прогнозировать будущее событие? В практике динамичной среды реального сектора прогнозы с периодом около месяца более актуальны. Меняя ширину прогнозируемого интервала с 90 дней на 28, получаем падение метрики площади под ROC-кривой до 0,614 при неправильной валидации и до 0,605 при правильной валидации. Выбор ширины интервала в 90 дней формально не является ошибкой, но это бесполезно в современном динамичном мире. Возможно даже, ширина в 28 дней является слишком большой, раз встречаются клиенты, совершающие покупки через 3–7 дней.

4) Ещё одна проблема заключается в том, что идентификатор клиента CustomerID включается как признак в обучающую матрицу X . Если его выбросить, то качество модели слегка падает, площадь под ROC-кривой становится 0,599. Это говорит о том, что идентификатор клиента мог частично объяснять наблюдаемую переменную. Идентификатор CustomerID имеет числовое значение, где-то в модели числовое значение идентификатора одних клиентов объясняет появление событий как тех же, так и других клиентов (обучение по всем клиентам). Это кажется нелогичным (особенно при неправильной валидации). Вопрос: оставлять или выбрасывать идентификатор клиента из признаков модели, является дискуссионным.

После исправления описанных недочётов уже можно доверять получаемой метрике качества модели как оценке способности модели предсказывать именно будущие события. Однако вопрос к корректности всё равно остаётся, так как модель обучается по всем клиентам. Корректно ли предсказывать будущий период для одного клиента, обучая модель на прошлых периодах других клиентов? Это дискуссионный вопрос, но именно

так часто делают в методах машинного обучения при работе с большими данными от разных клиентов.

Далее применяем метод, основанный на построении моделей механизмов образования событий (индивидуальных для каждого клиента). Можно предположить, что образование событий продиктовано нарастанием потребности что-то купить (шопинга), а потраченные средства гасят эту потребность. Эту модель можно назвать «деньги жгут карман». Чем больше денег потратил клиент, неважно на какие товары, тем позже будет следующая покупка. Тогда по количеству потраченных средств с помощью методов монотонной сплайновой коллокации, описанной во второй главе, восстанавливается нестационарная скорость восполнения потраченных средств на шопинг (или скорость нарастания желания совершить покупку, выраженная в долларах в день). Эта скорость экстраполируется на будущее с помощью разложения на сумму ограниченного количества гармоник алгоритмом Куинна и Фернандеса. Гиперпараметры, такие как коэффициент сглаживания α , количество отбрасываемых точек слева и справа n_1 и n_2 , подбираются с помощью поиска на сетке для глобальной оптимизации и алгоритма Нелдера-Мида для локальной оптимизации в окрестности узлов сетки, как описано в разделе 2.8. Критерием выступает ошибка прогноза двух последних событий обучающей выборки событий, где ошибка прогноза определяется по формуле (2.59). Схема обучения индивидуальных моделей изображена на рисунке 2.20 второй главы.

Мы применяем описанный метод для клиентов, у которых было хотя бы пять покупок, таких оказалось 1400 из 5415. Для оставшихся 4015 клиентов прогноз не делался (у большинства из них всего по одной-две покупки), для них смело можно предположить отсутствие события в будущем периоде. Получаем прогнозы будущих событий для отобранных 1400 клиентов. Однако, насколько можно доверять этим прогнозам? В процессе поиска гиперпараметров были найдены такие гиперпараметры, которые дают оптимальные значения погрешности $Score1$ и $Score2$ двух последних

событий обучающей выборки. Для тех клиентов, у которых эти погрешности оказываются меньше 6%, будем доверять построенным моделям и полученным прогнозам будущих покупок. Если по прогнозу последнего события опять построить переменную со значением 0 или 1 (вернуться в канву методов классификации), то площадь под ROC-кривой становится 0,615. У Нагеша использовался метод XGBClassifier, который является одним из лучших в машинном обучении, площадь под ROC-кривой для того же набора данных составляла 0,599 (или 0,605, если не выбрасывать признак CustomerID).

Однако у Нагеша использовался подход, когда обучающая матрица формировалась по всем клиентам. Можно использовать методы классификации по каждому клиенту в отдельности, когда обучающая матрица формируется так, как было описано в начале этого раздела (разбивая ось времени на интервалы шириной в 14, 7 дней и 3 дня, но прогноз делается на 28 дней). Выбирая методы классификации – дерево решений, градиентный бустинг, XGBClassifier, которые показали себя лучше всех в предыдущих примерах, проводим соответствующие исследования с разной шириной интервалов. Для метода дерева решений площадь под ROC-кривой при ширине интервала в 14, 7 дней и 3 дня составила 0,564; 0,557; 0,569 соответственно. Для метода градиентного бустинга площадь под ROC-кривой при ширине интервала в 14, 7 дней и 3 дня составила 0,573; 0,550; 0,556 соответственно. Для метода XGBClassifier библиотеки XGBoost площадь под ROC-кривой при ширине интервала в 14, 7 дней и 3 дня составила 0,631; 0,635; 0,607 соответственно.

В итоге имеем такие результаты. Предложенный метод (основанный на построении моделей механизмов образования событий для каждого клиента): обходит метод классификации XGBClassifier, если строить обучающую матрицу по всем клиентам (как у Нагеша); обходит метод дерева решений и градиентный бустинг, если строить обучающую матрицу индивидуально по каждому клиенту при интервалах 14, 7 дней и 3 дня; обходит метод

XGBClassifier, если строить обучающую матрицу индивидуально по каждому клиенту при интервалах в 3 дня, но проигрывает ему при интервалах в 14 и 7 дней. Таким образом, показано, что предлагаемый метод на основе моделей механизмов образования событий может являться конкурентным для передовых методов машинного обучения.

На самом деле метрика площади под ROC-кривой всё равно грубая для оценки способности предсказывать будущие события. Хотя событие и не выпало на указанный временной интервал, но оно могло выпасть с задержкой всего на 1 день. А если событие выпало на указанной интервал (правильно классифицировалось), то неизвестно, как далеко от фактической даты оно спрогнозировалось. Эта метрика классификации не учитывает точность прогноза. Да и к выбору ширины интервала остаются вопросы, желательно делать его как можно меньше, но тогда методы классификации начинают плохо работать.

В разобранных примерах, как на искусственно смоделированных данных, так и на реальных данных, было показано, что разработанный метод, основанный на моделях механизмов образования событий, зачастую превосходит современные методы классификации. Также во многих примерах было заметно, что получение хорошего прогноза методами классификации очень похоже на случайность, когда в обучающей выборке должна встретиться нужная последовательность наблюдений, которая позволит объяснить, сколько дней должно пройти до следующего события. Использование моделей механизмов образования событий, наоборот, позволяет объяснить причины удачных и неудачных предсказаний, например, погрешностью восстановления параметров. Самое главное – получается уловить суть появления события, а не только лишь статистическую зависимость от наблюдаемых и сконструированных признаков.

5.3 Сравнение с методами регрессии

Напомним: методы регрессии обучаются на выборке наблюдений $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ и $Y_i \in R$ (а не 0 или 1). Для событий, заданных

выборкой $(t_i; y_i)$, строим несколько зависимостей. Одной зависимостью будет длительность интервала времени до следующего события $Y_i = t_{i+1} - t_i$ от признаков X_i . В первой главе на рисунке 1.14 уже была представлена схема прогнозирования времени следующего события с помощью регрессионных методов. Второй зависимостью будет зависимость самих значений событий $Y_i = y_{i+1}$ от признаков X_i . Для большинства методов строятся две разные модели (не все предназначены выдавать несколько значений). А для метода, основанного на регрессии с помощью нейросетей, способного давать сразу два выходных значения, обойдёмся одной моделью.

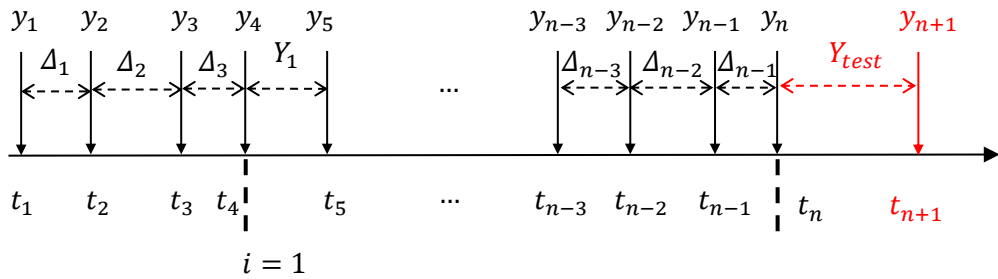
Модель для регрессионных методов

В отличие от модели классификации, где в качестве наблюдений использовались интервалы разной ширины, здесь в качестве наблюдений используются сами события. Модель получается даже проще. Будем использовать те же признаки, за исключением того, что не будет признака r (время до начала текущего интервала). Итого будет 14 признаков (здесь k – номер текущего события):

- $\Delta_{k-1}, \Delta_{k-2}, \Delta_{k-3}$ – ширина трёх предшествующих интервалов между событиями;
- $\bar{\Delta}$ – средняя ширина всех предшествующих интервалов между событиями;
- ν – средняя частота всех предшествующих событий;
- $y_k, y_{k-1}, y_{k-2}, y_{k-3}$ – значения текущего и предшествующих событий;
- \bar{y} – среднее значение всех предшествующих событий;
- $v_{k-1}, v_{k-2}, v_{k-3}$ – средняя скорость, с которой расходовался запас (т.е. $v_{k-1} = y_{k-1} / (t_k - t_{k-1})$);
- \bar{v} – среднее по всем предшествующим событиям средних скоростей.

Так как для каждого наблюдения задействуется информация от трёх предшествующих событий, то самым первым наблюдением будет информация, доступная к четвёртому событию. Пример формирования

первого наблюдения обучающей выборки и первого и единственного наблюдения контролирующей выборки показан на рисунке 5.33.



Источник: составлено автором.

Рисунок 5.33 – Пример формирования наблюдений

В результате получается обучающий набор данных, состоящий всего из $n - 4$ наблюдений, что может быть проблемой при идентификации параметров. Для простейшего сквозного примера (моделирования системы управления запасами, таблица 2.1) обучающий набор данных представлен на рисунке 5.34 (всего 10 наблюдений, но во многих методах так или иначе используется некоторая регуляризация).

	dt1	dt2	dt3	avg_dt	y	y1	y2	y3	avg_y	frequency	v1	v2	v3	avg_v	Yt	Yy
0	74	144	69	95.66667	3606.92133	3625.52996	3615.59078	3628.99402	3619.25902	0.01394	48.99365	25.10827	52.59412	42.23201	111	3609.15096
1	111	74	144	99.50000	3609.15096	3606.92133	3625.52996	3615.59078	3617.23741	0.01256	32.49479	48.99365	25.10827	39.79771	109	3631.26413
2	109	111	74	101.40000	3631.26413	3609.15096	3606.92133	3625.52996	3619.57520	0.01183	33.11148	32.49479	48.99365	38.46046	75	3614.04748
3	75	109	111	97.00000	3614.04748	3631.26413	3609.15096	3606.92133	3618.78552	0.01203	48.41886	33.11148	32.49479	40.11986	143	3608.33269
4	143	75	109	103.57143	3608.33269	3614.04748	3631.26413	3609.15096	3617.47892	0.01103	25.27306	48.41886	33.11148	37.98989	69	3618.26600
5	69	143	75	99.25000	3618.26600	3608.33269	3614.04748	3631.26413	3617.56637	0.01134	52.29468	25.27306	48.41886	39.78586	141	3601.89925
6	141	69	143	103.88889	3601.89925	3618.26600	3608.33269	3614.04748	3615.99966	0.01070	25.66146	52.29468	25.27306	38.21648	78	3649.90251
7	78	141	69	101.30000	3649.90251	3601.89925	3618.26600	3608.33269	3619.08174	0.01086	46.17820	25.66146	52.29468	39.01265	101	3612.13698
8	101	78	141	101.27273	3612.13698	3649.90251	3601.89925	3618.26600	3618.50301	0.01077	36.13765	46.17820	25.66146	38.75129	119	3629.91157
9	119	101	78	102.75000	3629.91157	3612.13698	3649.90251	3601.89925	3619.38059	0.01054	30.35409	36.13765	46.17820	38.05152	73	3620.06693

Источник: составлено автором.

Рисунок 5.34 – Обучающий набор данных для простейшего примера

Для предсказания будущего события контролирующей (тестовый) набор данных состоит всего из одного наблюдения, рисунок 5.35.

	dt1	dt2	dt3	avg_dt	y	y1	y2	y3	avg_y	frequency	v1	v2	v3	avg_v
0	73	119	101	100.46154	3620.06...	3629.9...	3612.1...	3649.9...	3619.42...	0.01072	49.724...	30.35409	36.13765	38.94947

Источник: составлено автором.

Рисунок 5.35 – Тестовый набор данных для простейшего примера

В качестве регрессионных методов используются почти все методы библиотеки scikit learn [107] и метод регрессии библиотеки XGboost [117]. Методы следующие: LinearRegression [183], SVR (Support Vector Regression) [108], KernelRidge (Kernel ridge regression [109]), KNeighborsRegressor [110], GaussianProcessRegressor [111], PLSRegression (Partial Least Squares [112]),

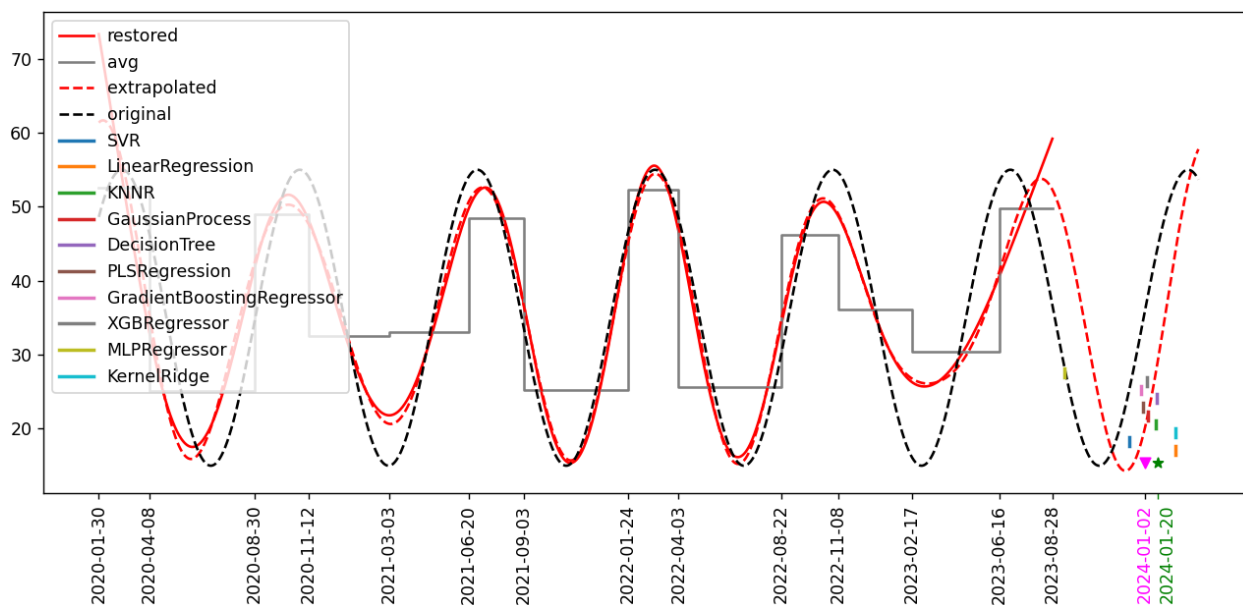
DecisionTreeRegressor [113], GradientBoostingRegressor [114], MLPRegressor (Multi-layer Perceptron Regressor) [115], XGBRegressor [117].

Далее представим результаты применения методов регрессии для прогнозирования событий и сравним их с результатами применения нового подхода, основанного на моделях механизма образования событий. Результатом применения методов является интервал времени до следующего события $\hat{t}_{n+1} - t_n$; откладывая этот интервал от последнего события t_n , получаем прогноз момента будущего события \hat{t}_{n+1} . Другим результатом является величина события \hat{y}_{n+1} .

Так как содержание задач, решаемых в примерах, уже было рассмотрено выше, сконцентрируемся на результатах.

Простейшая модель

Результаты предсказания будущего события представлены на рисунке 5.36 и в таблице 5.12. На рисунке фактическое событие обозначено зелёной звёздочкой ★; прогноз по методу, основанному на восстановлении параметров механизма образования событий, обозначен пурпурным треугольником ▼; прогнозы другими методами обозначаются вертикальными чёрточками разного цвета. Лучшее всего результат оказался у метода ближайших соседей и дерева решений, далее идут гауссовы процессы и XGBoost регрессор, они немного обходят предлагаемый метод в этом примере. Метод PLSRegression и градиентный бустинг дают результат чуть хуже. Остальные методы дают плохой прогноз.



Источник: составлено автором.

Рисунок 5.36 – Простейшая модель, результат предсказания будущего события

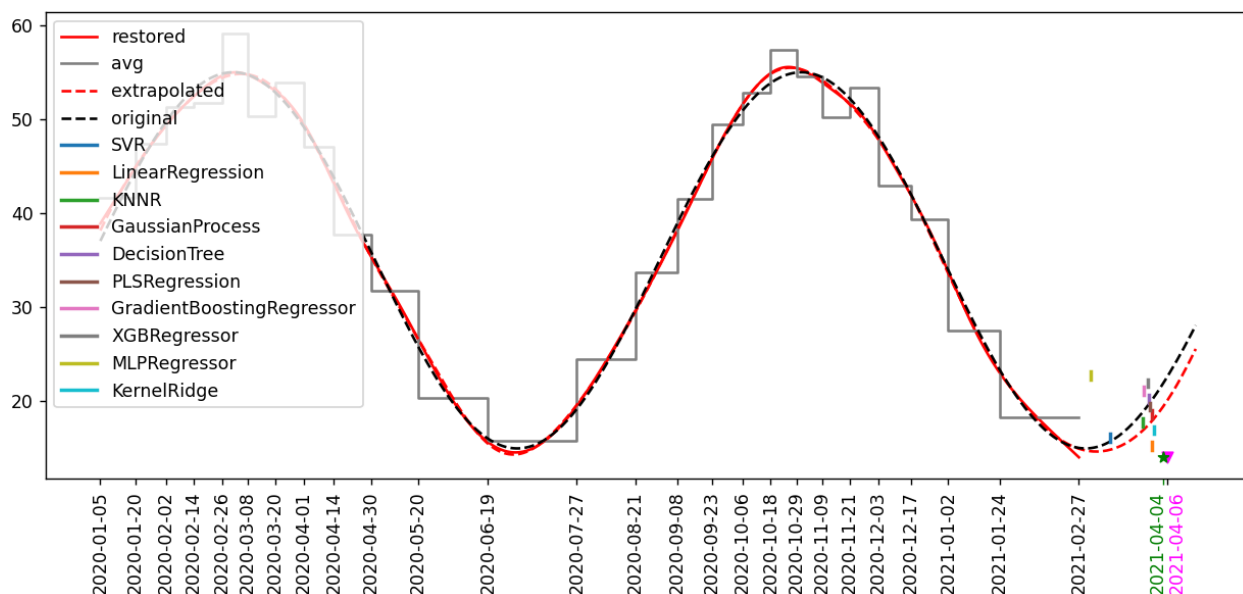
Таблица 5.12 – Прогноз будущего события в простейшей модели

Метод	Прогноз \hat{t}_{n+1}	Факт t_{n+1}	Прогноз \hat{y}_{n+1}	Факт y_{n+1}
Механизм образования событий	2024-01-02	2024-01-20	3613	3633
SVR	2023-12-11		3616	
LinearRegression	2024-02-13		3597	
KNeighborsRegressor	2024-01-17		3605	
GaussianProcessRegressor	2024-01-05		3611	
DecisionTreeRegressor	2024-01-16		3609	
PLSRegression	2023-12-30		3613	
GradientBoostingRegressor	2023-12-27		3610	
XGBRegressor	2024-01-04		3604	
MLPRegressor	2023-09-12		3985	
KernelRidge	2024-02-12		3608	

Источник: составлено автором.

Простейшая модель, другие данные

Результаты предсказания будущего события представлены на рисунке 5.37 и в таблице 5.13. Из регрессионных методов лучше всех результат оказался у метода KernelRidge, линейной регрессии и у гауссовых процессов, которые немного проигрывают предлагаемому методу. Далее следуют методы дерева решений, PLSRegression, XGBRegressor и градиентный бустинг. Остальные методы показывают плохой результат.



Источник: составлено автором.

Рисунок 5.37 – Простейшая модель с другими данными, результат предсказания будущего события

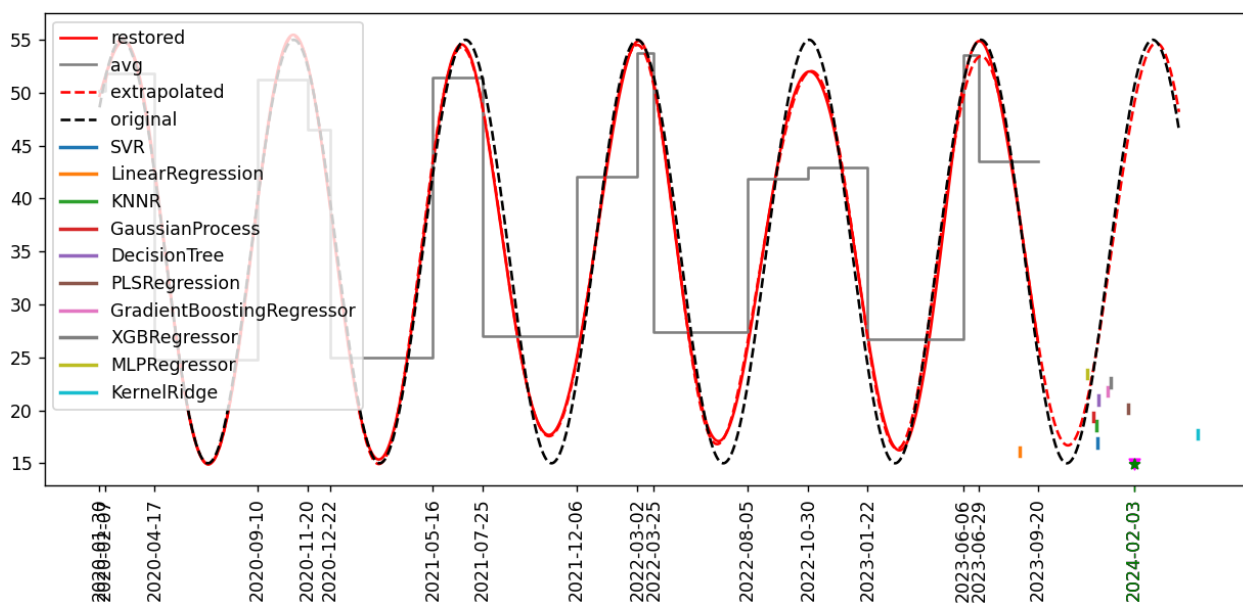
Таблица 5.13 – Прогноз будущего события в простейшей модели с другими данными

Метод	Прогноз \hat{t}_{n+1}	Факт t_{n+1}	Прогноз \hat{y}_{n+1}	Факт y_{n+1}
Механизм образования событий	2021-04-06	2021-04-04	608	621
SVR	2021-03-12		612	
LinearRegression	2021-03-30		600	
KNeighborsRegressor	2021-03-25		612	
GaussianProcessRegressor	2021-03-30		589	
DecisionTreeRegressor	2021-03-29		601	
PLSRegression	2021-03-29		597	
GradientBoostingRegressor	2021-03-26		616	
XGBRegressor	2021-03-28		613	
MLPRegressor	2021-03-03		682	
KernelRidge	2021-03-31	586		

Источник: составлено автором.

Модель с дополнительными событиями

Результаты предсказания будущего события представлены на рисунке 5.38 и в таблице 5.14. Среди методов регрессии самый лучший результат оказался у метода PLSRegression. Далее идут методы XGBRegressor и градиентный бустинг, однако они ошибаются уже более чем на месяц. Остальные методы дают совсем плохой прогноз. Обычная линейная регрессия вообще выдала отрицательный интервал между событиями и спрогнозировала событие в прошлом.



Источник: составлено автором.

Рисунок 5.38 – Модель с дополнительными событиями, регрессионные методы

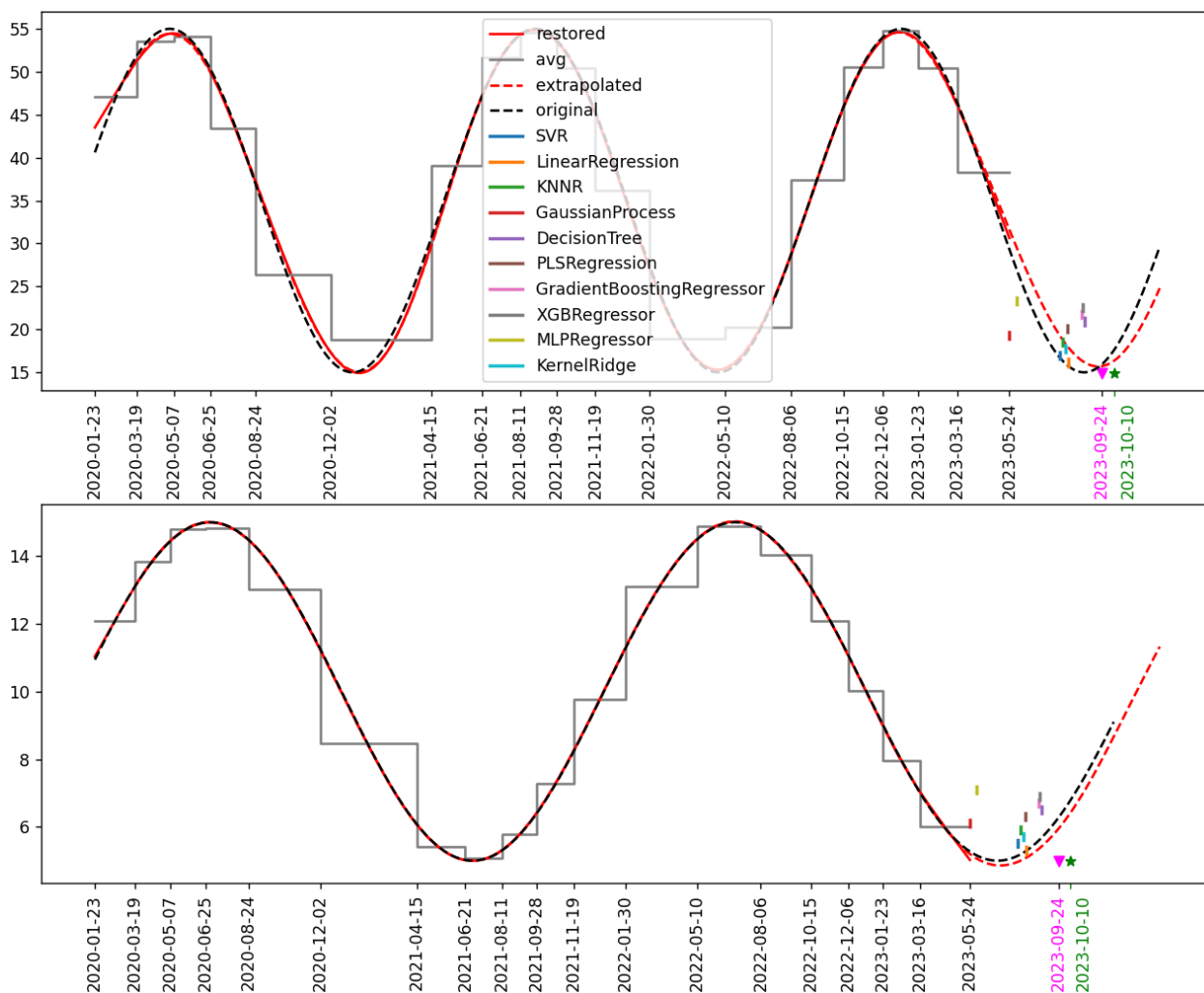
Таблица 5.14 – Прогноз будущего события в модели с дополнительными событиями

Метод	Прогноз \hat{t}_{n+1}	Факт t_{n+1}	Прогноз \hat{y}_{n+1}	Факт y_{n+1}
Механизм образования событий	2024-02-03	2024-02-03	3641	3601
SVR	2023-12-13		3605	
LinearRegression	2023-08-25		3131	
KNeighborsRegressor	2023-12-11		3608	
GaussianProcessRegressor	2023-12-07		3578	
DecisionTreeRegressor	2023-12-15		3605	
PLSRegression	2024-01-26		4245	
GradientBoostingRegressor	2023-12-27		3549	
XGBRegressor	2024-01-01		3357	
MLPRegressor	2023-11-29		3603	
KernelRidge	2024-05-03		6860	

Источник: составлено автором.

Модель с двумя продуктами

Результаты предсказания будущего события представлены на рисунке 5.39 и в таблице 5.15. Из используемых методов регрессии лучший результат показал метод деревьев решений. Следующими идут XGBRegressor и градиентный бустинг. Но они всё равно значительно ошибаются. Ошибка остальных методов совсем огромная. Лучший прогноз получился у предлагаемого метода на основе восстановления параметров механизма образования событий.



Источник: составлено автором.
Рисунок 5.39 – Модель с двумя продуктами

Таблица 5.15 – Прогноз будущего события в модели с двумя продуктами

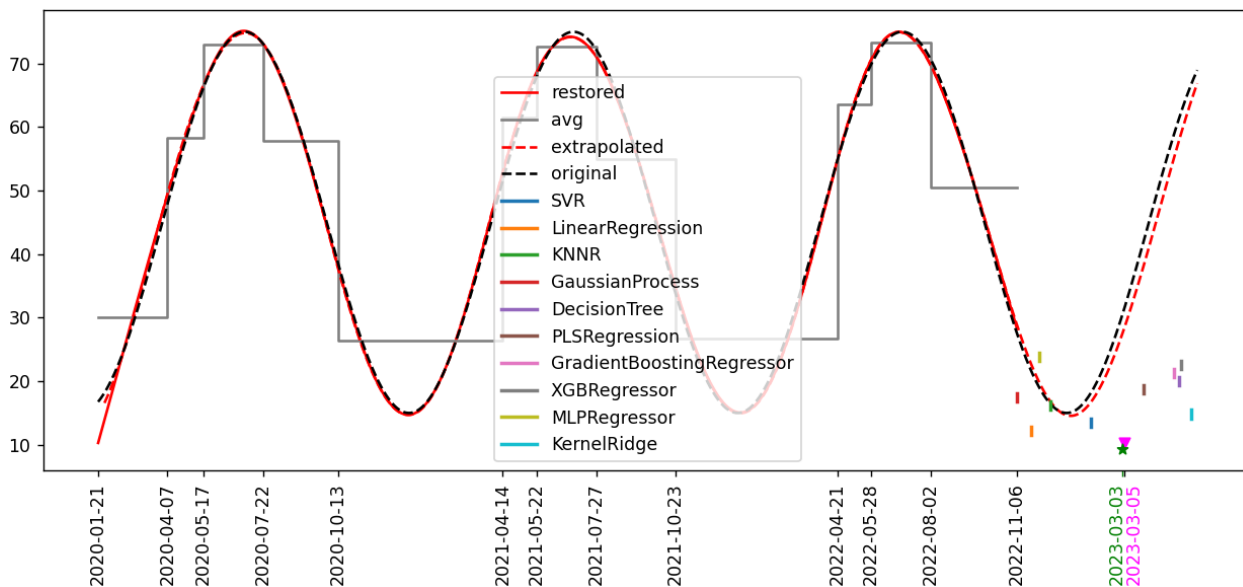
Метод	Прогноз \hat{t}_{n+1}	Факт t_{n+1}	Прогноз $\hat{y}_{1,n+1}$	Факт $y_{1,n+1}$	Прогноз $\hat{y}_{2,n+1}$	Факт $y_{2,n+1}$
Механизм образования событий	2023-09-24	2023-10-10	2604	2614	759	759
SVR	2023-07-30		2619		628	
LinearRegression	2023-08-10		2223		697	
KNeighborsRegressor	2023-08-02		2489		723	
GaussianProcessRegressor	2023-05-24		0		705	
DecisionTreeRegressor	2023-09-01		2617		1309	
PLSRegression	2023-08-08		2373		746	
GradientBoostingRegressor	2023-08-27		2612		769	
XGBRegressor	2023-08-28		2600		765	
MLPRegressor	2023-06-02		2511		765	
KernelRidge	2023-08-06		2163		852	

Источник: составлено автором.

Модель с изменяющимся максимумом

Результаты предсказания будущего события представлены на рисунке 5.40 и в таблице 5.16. Все используемые методы регрессии не смогли

выявить правильной закономерности в данных, полученных в модели с изменяющимся максимумом. Все методы регрессии дали очень плохой прогноз. Метод на основе восстановления параметров механизма образования событий хорошо справился с этой непростой задачей.



Источник: составлено автором.

Рисунок 5.40 – Модель с изменяющимся максимумом

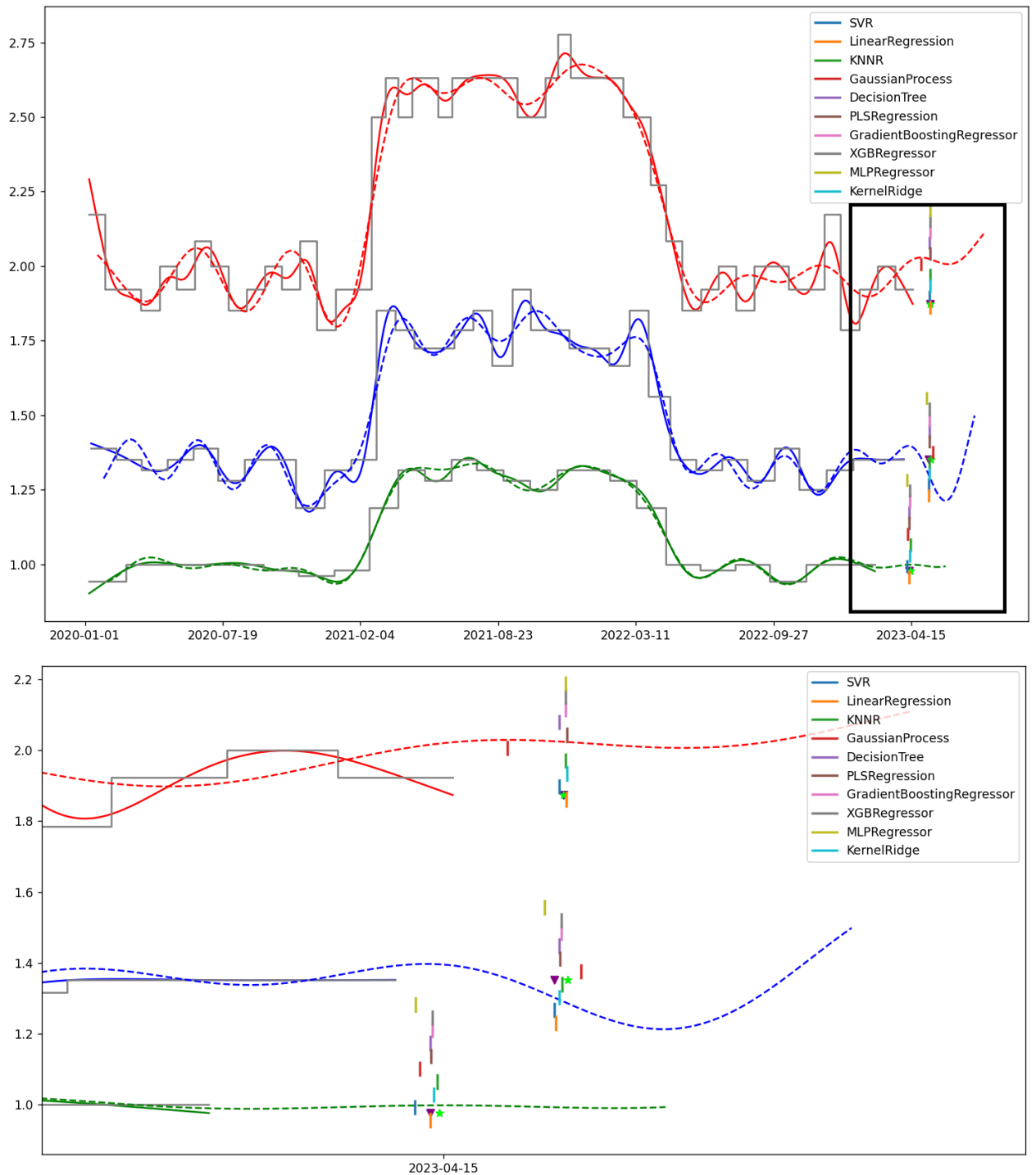
Таблица 5.16 – Прогноз будущего события в модели с изменяющимся максимумом.

Метод	Прогноз \hat{t}_{n+1}	Факт t_{n+1}	Прогноз \hat{y}_{n+1}	Факт y_{n+1}
Механизм образования событий	2023-03-05	2023-03-03	1084	1075
SVR	2023-01-27		4815	
LinearRegression	2022-11-21		8448	
KNeighborsRegressor	2022-12-13		4820	
GaussianProcessRegressor	2022-11-06		0	
DecisionTreeRegressor	2023-05-05		2313	
PLSRegression	2023-03-26		2751	
GradientBoostingRegressor	2023-04-29		3560	
XGBRegressor	2023-05-06		2353	
MLPRegressor	2022-11-30		3770	
KernelRidge	2023-05-17		-8918	

Источник: составлено автором.

Модель поставки запчастей для авиатранспорта

Результаты предсказания будущего события представлены на рисунке 5.41. Почти все методы регрессии дают хороший прогноз. Однако методами регрессии нельзя получить те выводы, которые получились с помощью восстановления параметров механизма образования событий.



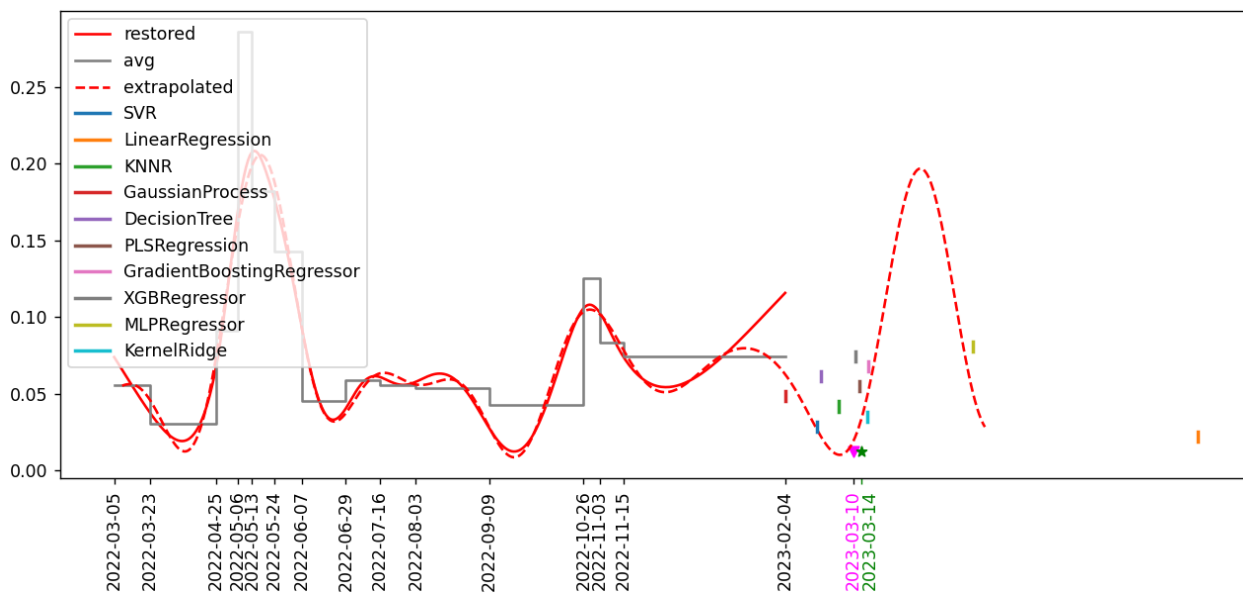
Источник: составлено автором.

Рисунок 5.41 – Прогноз следующего пополнения запаса запчастей

Поставка биологически активных добавок в аптечные пункты

Результаты предсказания будущего события представлены на рисунке 5.42 и в таблице 5.17. Среди методов регрессии самым лучшим предсказанием момента будущего события t_{n+1} обладает метод PLSRegression, однако величину y_{n+1} он прогнозирует отрицательной.

Следующими идут методы XGBRegressor и градиентный бустинг. Остальные методы дают плохой прогноз.



Источник: составлено автором.

Рисунок 5.42 – Прогноз следующего пополнения запаса Омега-3

Таблица 5.17 – Прогноз будущего пополнения запаса Омега-3

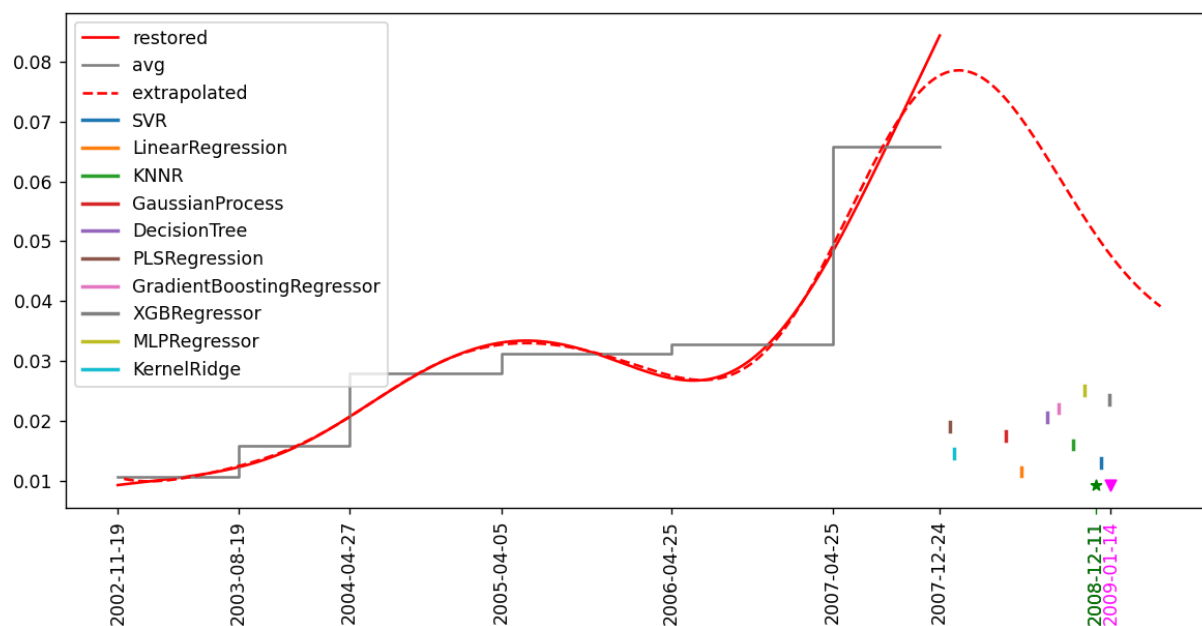
Метод	Прогноз \hat{t}_{n+1}	Факт t_{n+1}	Прогноз \hat{y}_{n+1}	Факт y_{n+1}
Механизм образования событий	2023-03-10	2023-03-14	1,63	1
SVR	2023-02-19		6,75	
LinearRegression	2023-08-29		1,12	
KNeighborsRegressor	2023-03-03		1	
GaussianProcessRegressor	2023-02-04		0,02	
DecisionTreeRegressor	2023-02-22		1	
PLSRegression	2023-03-13		-4,11	
GradientBoostingRegressor	2023-03-17		1,07	
XGBRegressor	2023-03-11		0,96	
MLPRegressor	2023-05-09		-3,99	
KernelRidge	2023-03-16		-2,06	

Источник: составлено автором.

Пример из сферы финансов, займы/облигации – Московская область

Величину события y_{n+1} предсказываем следующим образом. Замечаем, что в данных (после объединения событий), таблица 5.10, практически ежегодно идёт значительное увеличение объёма выпуска облигаций. Среднее увеличение составляет около 43%. Тогда прогноз y_{n+1} даёт 37. Результаты предсказания будущего события представлены на рисунке 5.43 и в таблице 5.18.

Из используемых методов регрессии лучший прогноз момента наступления события показал метод SVR (регрессия с помощью метода опорных векторов). Следующими идут методы XGBRegressor и MLPRegressor. Однако значение y_{n+1} они предсказывают заниженными. Линейная регрессия даёт лучший прогноз y_{n+1} , но очень плохой прогноз t_{n+1} . Остальные методы значительно ошибаются в обеих величинах.



Источник: составлено автором.

Рисунок 5.43 – Прогноз следующего выпуска облигаций

Таблица 5.18 – Прогноз будущего выпуска облигаций

Метод	Прогноз \hat{t}_{n+1}	Факт t_{n+1}	Прогноз \hat{y}_{n+1}	Факт y_{n+1}
Механизм образования событий	2009-01-14	2008-12-11	37	33
SVR	2008-12-22		16	
LinearRegression	2008-06-26		29	
KNeighborsRegressor	2008-10-20		20	
GaussianProcessRegressor	2008-05-21		5	
DecisionTreeRegressor	2008-08-23		26	
PLSRegression	2008-01-16		44	
GradientBoostingRegressor	2008-09-17		24	
XGBRegressor	2009-01-11		12	
MLPRegressor	2008-11-15		23	
KernelRidge	2008-01-25		34	

Источник: составлено автором.

Прогнозирование будущих покупок многими клиентами в онлайн-магазине

По событиям каждого клиента отдельно обучается регрессионная модель. Обучающая матрица формируется так же, как описано выше.

С помощью обученной модели получается прогноз интервала времени до следующего события. Откладывая его от последнего события, получаем прогноз момента времени будущего события. Обрато возвращаемся в рамки задачи классификации. Если прогнозируемое событие попадает в следующий интервал, равный 28 дням, то классифицируем прогноз как 1, иначе 0. Так поступаем по всем клиентам. Далее сверяем результаты прогноза с реальными и получаем метрику качества классификации (сверяем по тем же клиентам, для которых была принята модель механизма образования события, т.е. для которых погрешность определения двух последних событий *Score1* и *Score2* оказалась меньше 6%).

В итоге площадь под ROC-кривой составила: 0,602 у метода дерева решений; 0,581 – у GradientBoostingRegressor; 0,569 – у XGBRegressor. У предлагаемого метода, основанного на восстановлении параметров механизма образования событий, площадь под ROC-кривой была 0,615. В итоге предложенный метод обходит передовые методы регрессии (когда прогноз строится только по выборке событий). Правда, у всех методов низкая точность.

В разобранных примерах, как на искусственно смоделированных данных, так и на реальных данных, было продемонстрировано, что разработанный метод, основанный на восстановлении моделей механизмов образования событий, зачастую превосходит современные регрессионные методы. Отдельно хочется отметить, что когда на одних данных одни методы регрессии давали хороший, то на других данных эти же методы давали уже плохой прогноз. Зависит это от того, есть или нет в обучающей выборке нужная последовательность наблюдений, которая объясняет, сколько дней должно пройти до следующего события. Но наличие или отсутствие такой последовательности больше похоже на случайность. В новом подходе, когда происходит восстановление параметров механизма образования событий, каждое событие вносит свой вклад. Таким образом, показано, что

предлагаемый метод на основе моделей механизмов образования событий является конкурентным для передовых методов машинного обучения.

5.4 Сравнение с другими методами

В этом параграфе сравнивается предлагаемый подход, основанный на восстановлении параметров модели механизмов образования событий, с методами на основе потоков случайных событий, а также с адаптированными для получения прогноза события методами Кростона и Виллимейна.

Пуассоновский поток событий. В первой главе было сказано, что для получения прогноза следующего события правильно откладывать наиболее вероятное значение (моду), а не математическое ожидание. У пуассоновского потока интервалы времени между событиями подчинены экспоненциальному закону, наиболее вероятное значение которого находится в нуле. Тем не менее из-за популярности пуассоновского потока мы обязаны проверить его способность прогнозировать будущие события. Для этого будем откладывать математическое ожидание, равное $1/\lambda$, где λ – интенсивность событий. Однако интенсивность у пуассоновского потока может быть нестационарная $\lambda(t)$. Оценивать её будем с помощью ядерных функций (в качестве ядра используется нормальное распределение). Но для прогноза будет использоваться последнее значение этой нестационарной интенсивности, тем самым от предыдущего события откладывается $1/\lambda(t_n)$.

Поток Пальма (нормальное распределение). Потоки Пальма по определению всегда стационарные. Поэтому предполагается, что все интервалы между событиями подчинены одному и тому же закону распределения. В данном случае нормальному распределению, у которого наиболее вероятное значение совпадает с математическим ожиданием. Таким образом, от предыдущего события откладывается среднее значение предшествующих интервалов между событиями $\bar{\Delta}$.

Поток Пальма (распределение Эрланга $k = 2$). У распределения Эрланга мода определяется по формуле $(k - 1)/\lambda$, где λ – уже другая

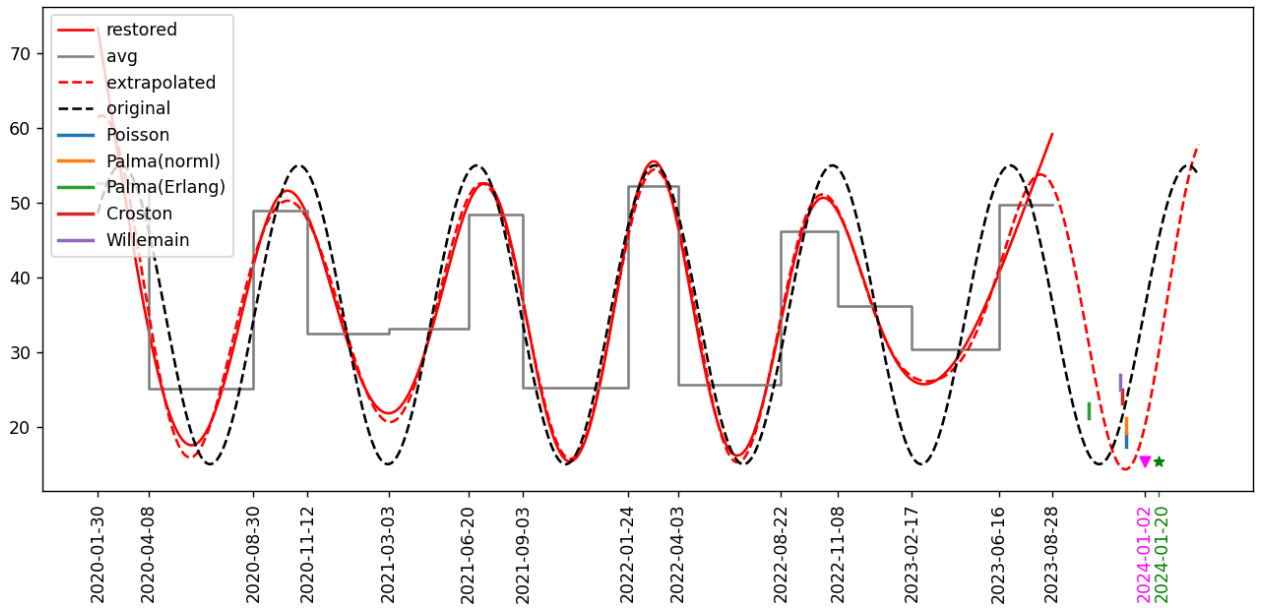
интенсивность. Эту интенсивность определяем методом моментов. Математическое ожидание распределения Эрланга $m = k/\lambda$ приравняем к среднему значению интервалов между событиями $\bar{\Delta}$, выражаем $\lambda = k/\bar{\Delta}$. Тогда мода $\bar{\Delta}(k - 1)/k$. Получаем прогноз, откладывая наиболее вероятный интервал времени (с ростом k мода приближается к среднему значению) от последнего события. Например, при $k = 2$ надо откладывать $\bar{\Delta}/2$.

Метод Кростона. В этом методе происходит экспоненциальное сглаживание интервалов времени между событиями (количества интервалов между ненулевым спросом, но если выбрать интервал, равный одному дню, то имеем количество дней между событиями). Для этого используем формулу (1.50) с параметром экспоненциального сглаживания $\alpha = 0,15$. От последнего события будем откладывать последнее сглаженное значение времени между событиями.

Метод Виллимейна. В этом методе оцениваются переходные вероятности марковской цепи с двумя состояниями, рисунок 1.22 в первой главе. Но предварительно надо опять разбить ось времени на интервалы, ширину которых выберем 7 дней. Далее подсчитываем количество переходов n_{00} (из состояния 0 в 0) и n_{01} (из 0 в 1), так как нас интересуют лишь переходные вероятности $P_{00} = n_{00}/(n_{00} + n_{01})$ и $P_{01} = n_{01}/(n_{00} + n_{01})$. Далее оценивается количество шагов до первого перехода из состояния 0 в состояние 1 по формуле (1.52), $\bar{t}_{0,1} = \frac{1}{P_{01}}$. Для прогноза будущего события откладываем $\bar{t}_{0,1}$ недель от последнего события.

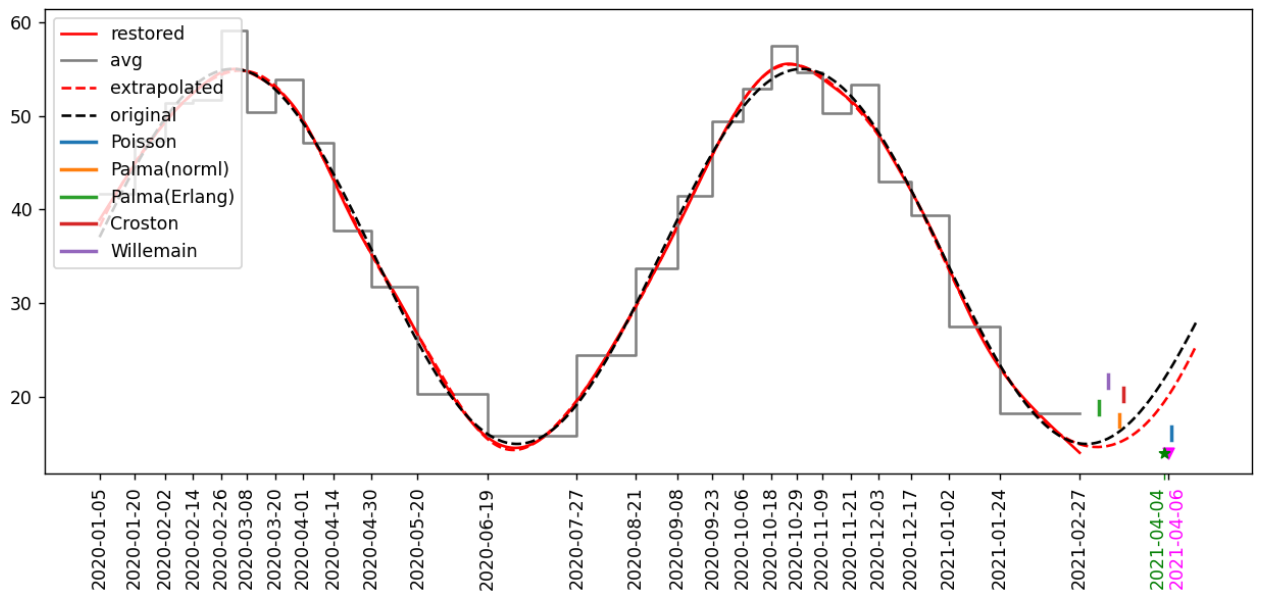
Результаты применения описанных методов следующие (зелёной звёздочкой ★ обозначается фактический прогноз, пурпурным треугольником ▼ обозначается прогноз по методу, основанному на восстановлении параметров механизма образования событий).

Простейшая модель. Результаты предсказания момента будущего события представлены на рисунке 5.44. Все методы дали прогноз хуже, чем у предлагаемого метода.



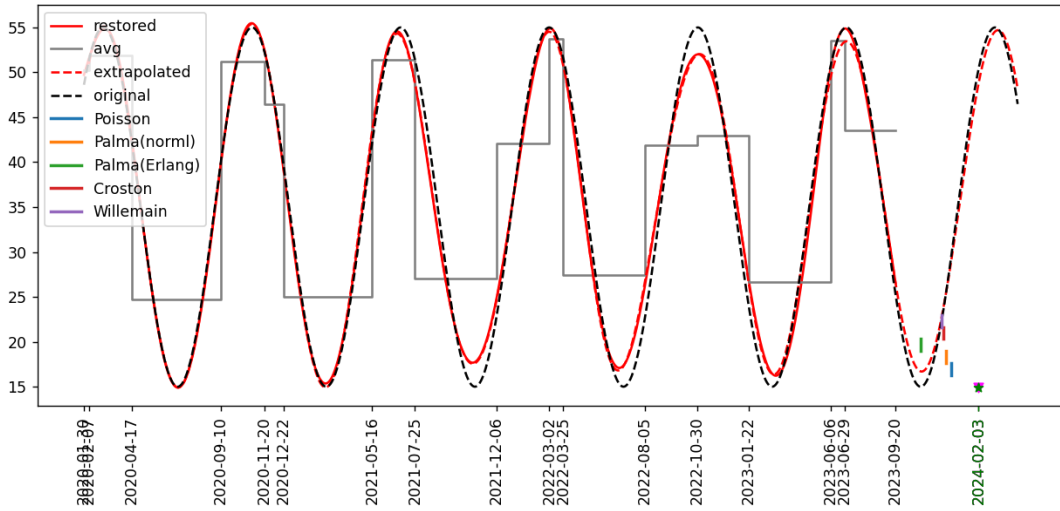
Источник: составлено автором.
Рисунок 5.44 – Простейшая модель

Простейшая модель, другие данные. Результаты предсказания момента будущего события представлены на рисунке 5.45. Прогноз с помощью пуассоновского потока событий оказался хорошим (2021-04-07). Остальные методы дали очень плохой прогноз.



Источник: составлено автором.
Рисунок 5.45 – Простейшая модель с другими данными

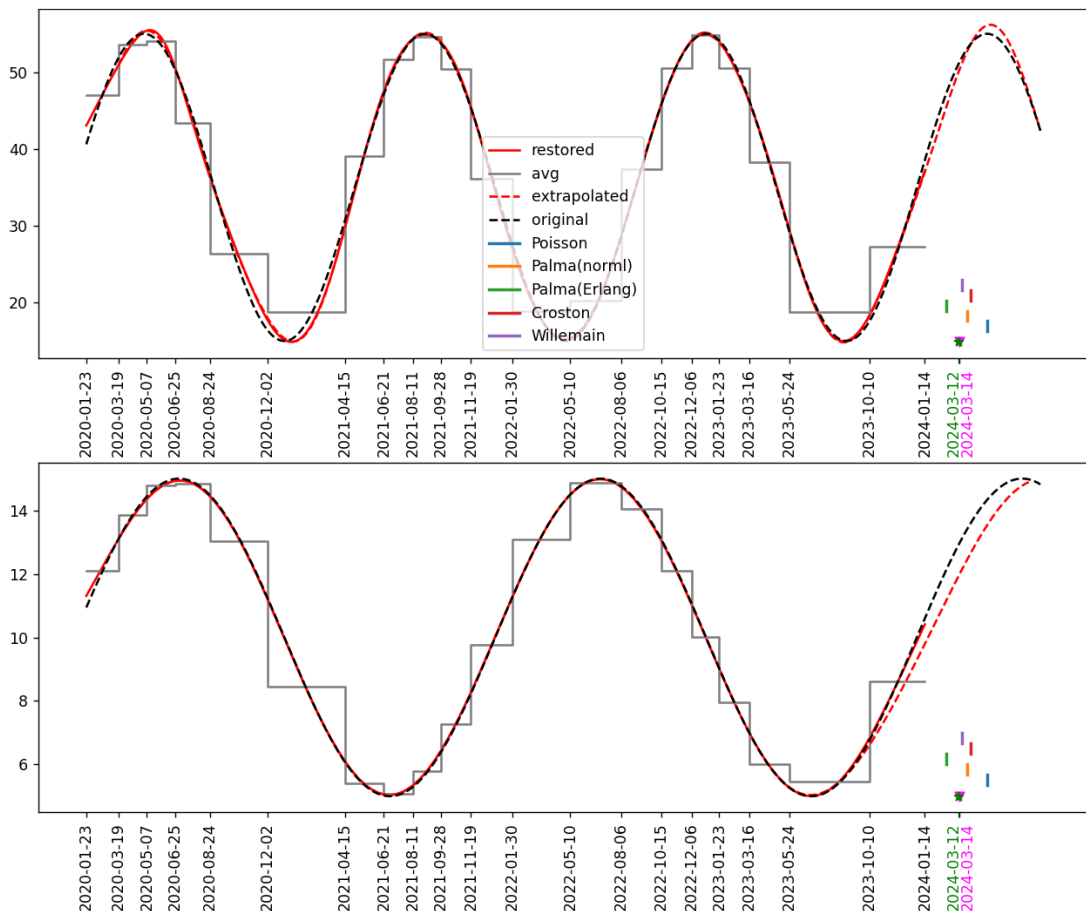
Модель с дополнительными событиями. Результаты предсказания момента будущего события представлены на рисунке 5.46. Все методы дали плохой прогноз.



Источник: составлено автором.

Рисунок 5.46 – Модель с дополнительными событиями

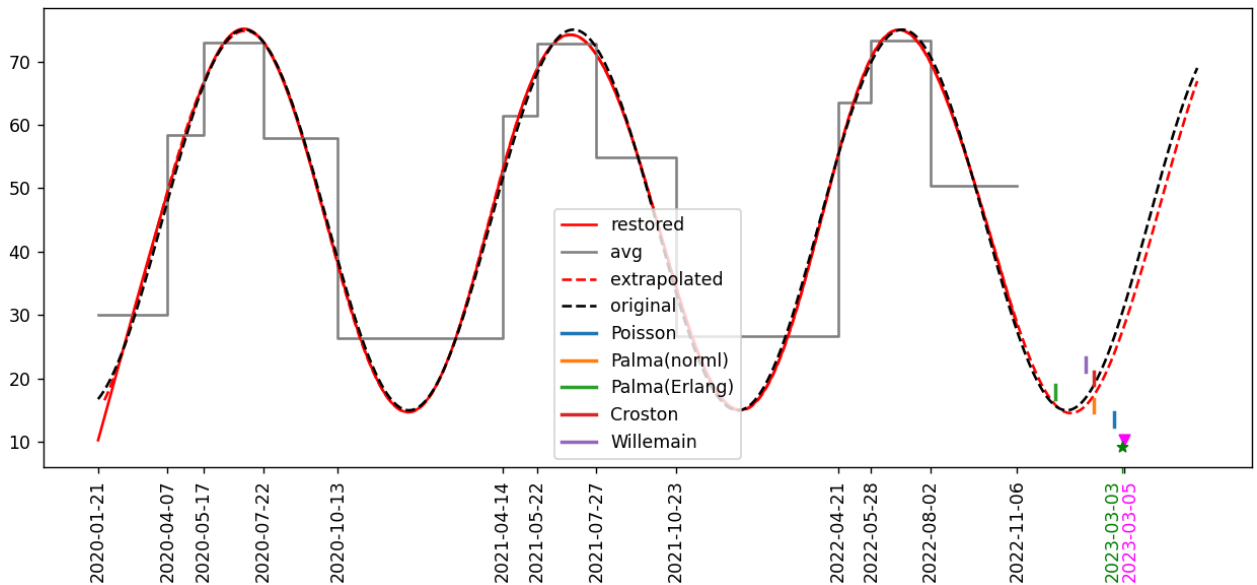
Модель с двумя продуктами. Результаты предсказания момента будущего события представлены на рисунке 5.47. В этот раз у метода Виллимейна и потока Пальма (с нормальным распределением) оказался хороший прогноз (2024-03-18 и 2024-03-26). Остальные методы дали плохой прогноз.



Источник: составлено автором.

Рисунок 5.47 – Модель с двумя продуктами

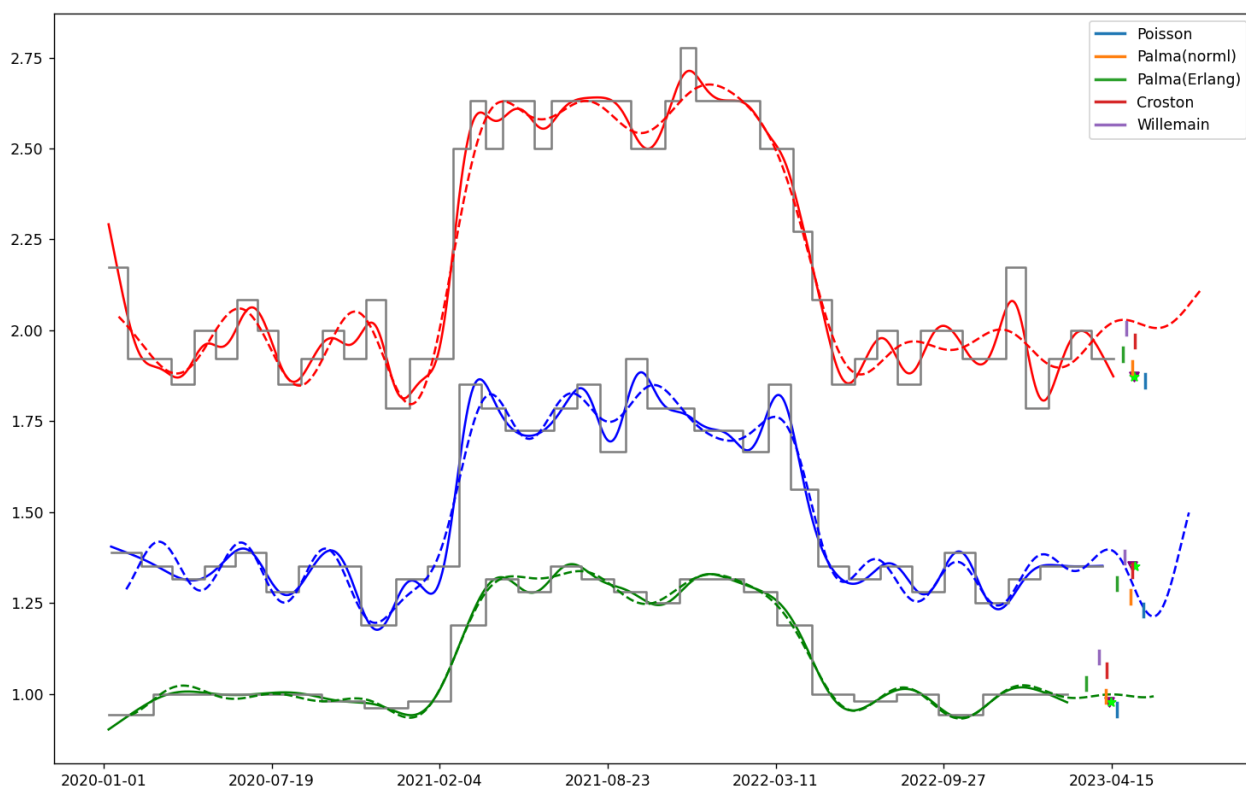
Модель с изменяющимся максимумом. Результаты предсказания момента будущего события представлены на рисунке 5.48. Прогноз с помощью пуассоновского потока событий (2023-02-21) оказался хорошим. Остальные методы дали плохой прогноз.



Источник: составлено автором.

Рисунок 5.48 – Модель с изменяющимся максимумом

Модель поставки запчастей для авиатранспорта. Результаты предсказания момента будущего события представлены на рисунке 5.49. Метод на основе потока Пальма (с нормальным распределением) и метод Кростона тут дают хороший прогноз. Остальные методы значительно ошибаются.

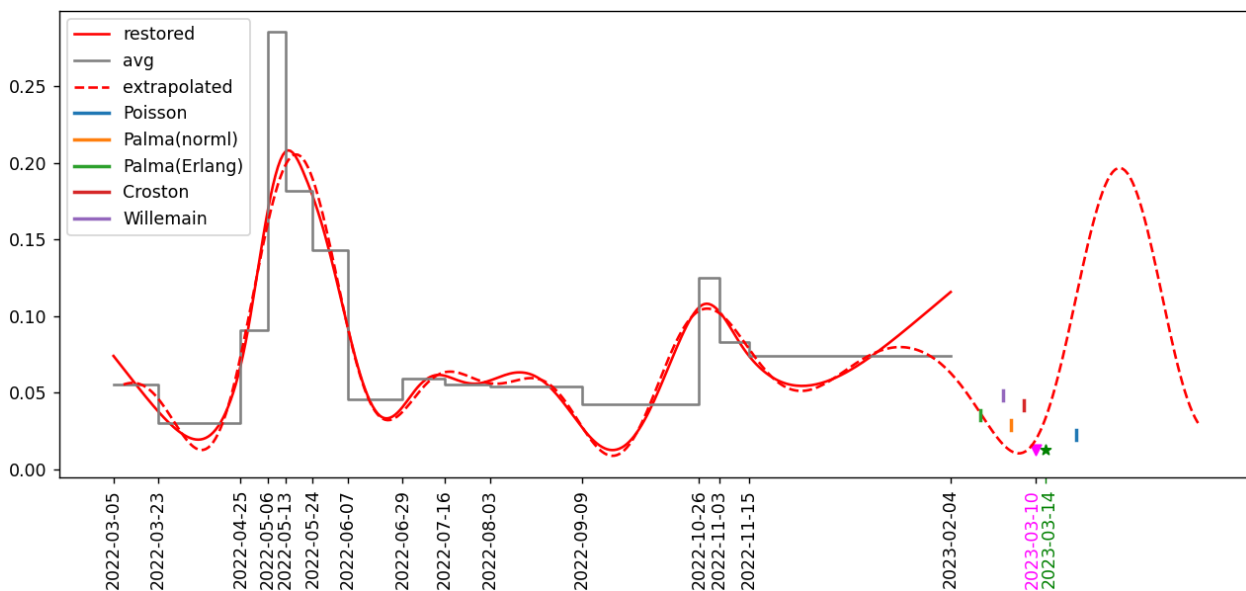


Источник: составлено автором.

Рисунок 5.49 – Модель поставки запчастей для авиатранспорта

Поставка биологически активных добавок в аптечные пункты.

Результаты предсказания момента будущего события представлены на рисунке 5.50. Все указанные методы дали плохой прогноз.

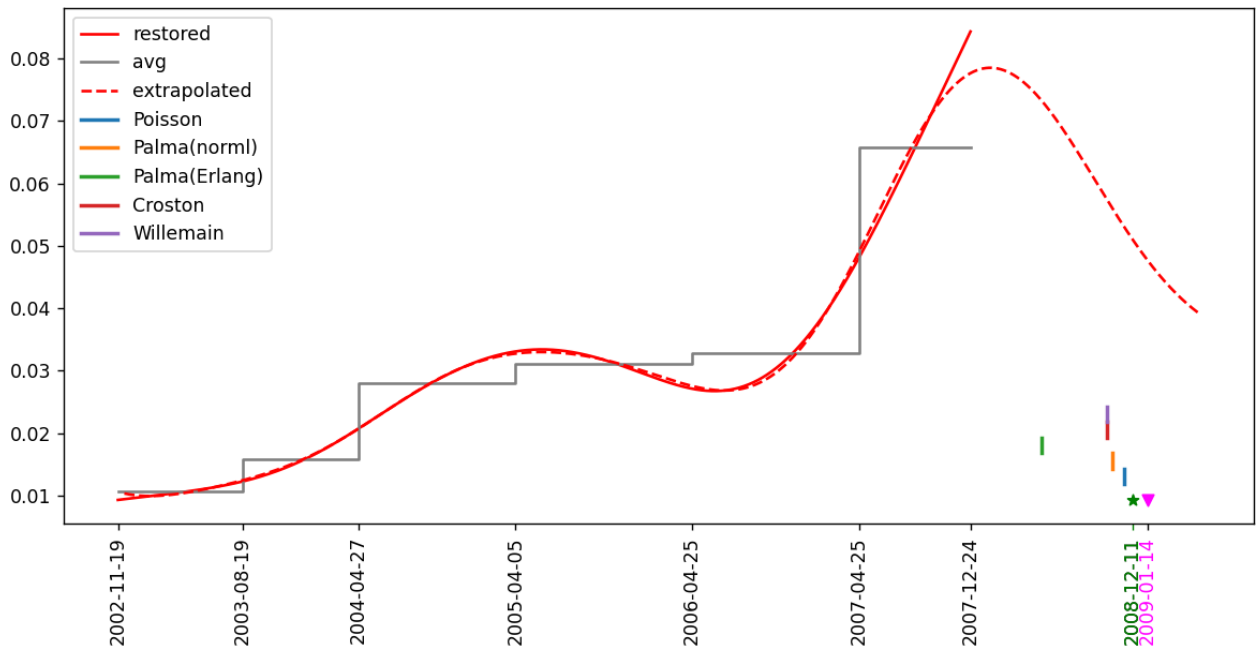


Источник: составлено автором.

Рисунок 5.50 – Прогноз следующего пополнения запаса Омега-3

Пример из сферы финансов, займы/облигации – Московская область. Результаты предсказания момента будущего события представлены

на рисунке 5.51. Прогноз с помощью пуассоновского потока событий (2008-11-24) оказался хорошим. Остальные методы дали плохой прогноз.



Источник: составлено автором.

Рисунок 5.51 – Прогноз следующего выпуска облигаций

В итоге на разобранных примерах было показано, что разработанный метод, основанный на моделях механизмов образования событий, практически всегда превосходит рассмотренные методы, основанные на потоках Пуассона и Пальма, а также на основе метода Кростона и Виллимейна. Заметим, что когда на одних данных некоторые из этих методов давали хороший прогноз, то на других данных эти же методы давали уже плохой прогноз. Никакого постоянства нет. В целом в большинстве случаев эти методы давали плохой прогноз.

Классификация задач и выбор метода прогнозирования событий

Можно выделить несколько классов задач, относящихся к прогнозу событий:

- классификация появления события на интервале времени – прогноз представляет собой либо бинарное значение 0 или 1 как факт появления события на интервале или вероятность появления события, например, 70%, что событие произойдет на следующей неделе;

- прогнозирование момента времени появления события или времени до события – прогноз представляет вещественное значение, например указывающее, что событие произойдет через 5 дней или в указанную дату;
- определение закона распределения для времени появления события – результатом является функция плотности распределения.

При этом класс задач может варьироваться в зависимости от того, какие данные анализируются:

- имеется одно-единственное событие как феномен, который требуется изучить – в этом случае необходимо кейс-исследование (детальное исследование конкретного случая в контексте процессов реального мира), системный анализ возможных причин и обстоятельств, сценарное или феноменологическое моделирование, экспертное оценивание, или же необходимо собрать больше данных о похожих случаях;
- имеется временной ряд некоторого показателя, например, финансового, а событие заключается в преодолении им некоторого порогового значения – в этом случае необходим анализ временных рядов, прогнозом тогда будет являться сам уровень временного ряда и его момент времени преодоления порога (для анализа временных рядов можно применять любые существующие методы – от выделения трендов и сезонности, построение авторегрессионных моделей зависимости показателя от предшествующих лаговых значений, более сложные модели машинного обучения или нейросетевые модели);
- имеется поток событий, как временные отметки, возможно, с сопутствующими дополнительными характеристиками – в этом случае подойдут методы, разрабатываемые в этой работе (как представленные в первой главе и адаптированные для прогнозирования событий, так и специализированный авторский подход, изложенный во второй и последующих главах).

Дополнительно можно различать события по таким факторам:

- имеется ли хоть какая-то информация, содержащаяся в наблюдаемых признаках и предвещающая появление событий, или же события появляются полностью без каких-либо предвещающих признаков;
- способны ли события от одного источника (например клиента) помочь объяснить появление событий в другом источнике, как похожим образом происходит в рекомендательных системах.

Можно еще выделить горизонт планирования: краткий, средний или долгий. Однако это определяется не столько классом задачи, сколько прикладной областью и характером данных. Некоторые задачи могут оперировать днями, некоторые – месяцами или годами. Для одних задач прогноз на месяц может быть долгосрочным, когда для других задач прогноз на год будет всё ещё краткосрочным. Таким образом, горизонт планирование не определяет класс задачи.

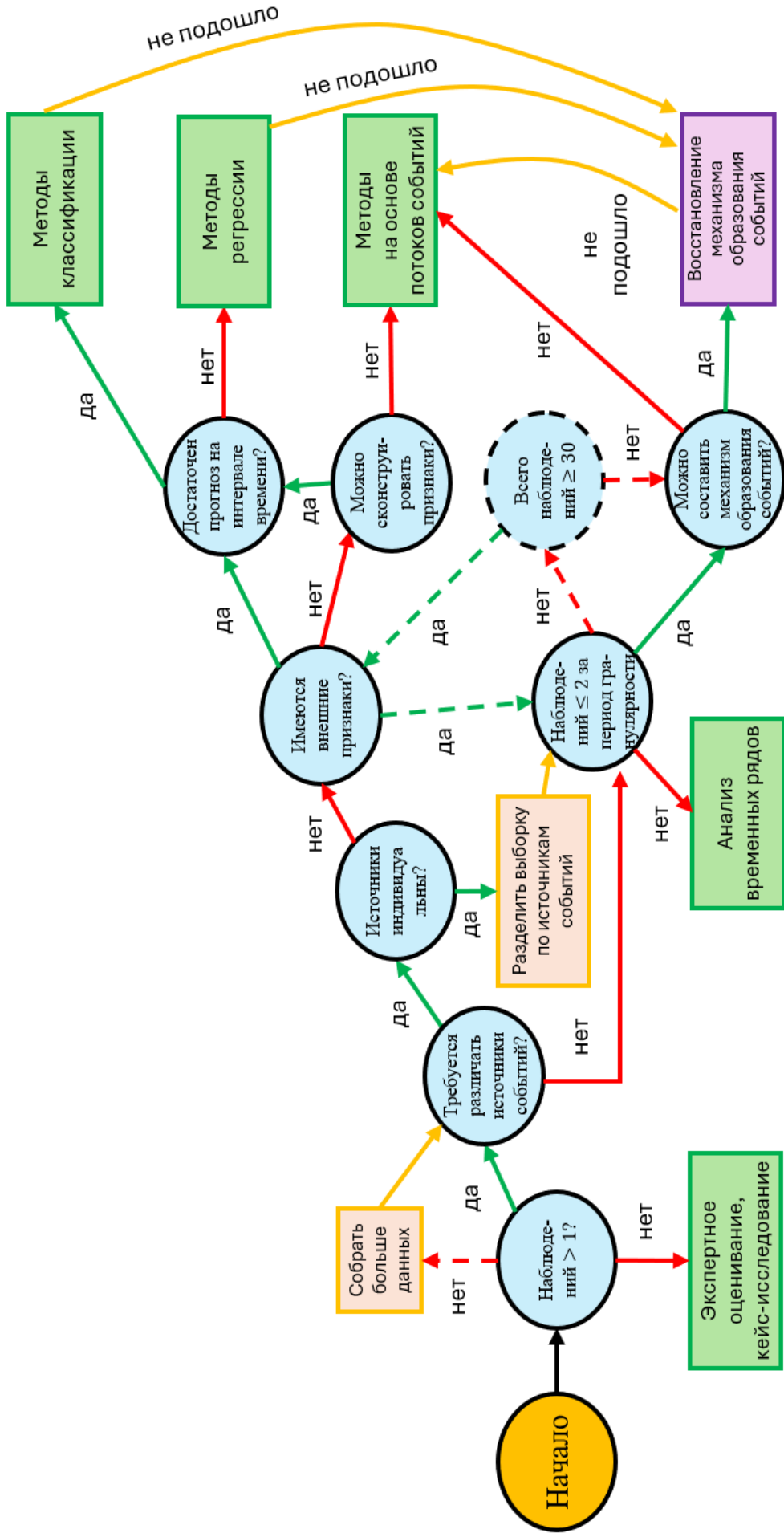
Для перечисленных классов задач разработан алгоритм выбора метода прогнозирования события, схема которого представлена на рисунке 5.52. Пунктирными стрелочками обозначены альтернативы, подразумевая, что можно поступать несколькими способами. В этом случае следует сопоставлять качество прогноза несколькими альтернативными методами по метрикам, составленным в подразделе 2.8, таблица 2.7, или по другим метрикам оценки экономического эффекта, которые обсуждаются в конце второй главы (бизнес-метрики, которые переводят стратегические цели в измеримые показатели).

Рекомендации по выбору метода прогнозирования следующие.

В случае, если требуется определить, произойдет событие на определенном интервале времени или нет, могут подойти методы классификации, при условии, что признаки содержат информацию, позволяющую прогнозировать события в будущем. Однако методами классификации нельзя спрогнозировать характеристики события y_i .

Методами регрессии и предложенным новым подходом можно дать прогноз для момента времени t_i появления события и характеристик события y_i . Зная момент времени t_i , тривиально определяется появление события на

определённом интервале времени. Поэтому методы регрессии предпочтительнее, чем методы классификации. Опять же, для методов регрессии требуется, чтобы наблюдаемые признаки несли информацию о возникновении события в будущем (о ширине интервала времени до следующего события). Зачастую сами признаки могут отсутствовать, а иметься лишь выборка событий. Построение искусственных признаков, таких как средняя частота событий, средняя ширина интервалов и др., может не помогать в этом деле. В случае, если известен механизм образования событий, предложенный подход, основывающийся на восстановлении по выборке событий параметров этого механизма, способен давать более точный прогноз. Другим преимуществом является то, что такой прогноз будет более понятен и обоснован. В методах машинного обучения зачастую непонятно, почему метод выдал то или иное значение.



Источник: составлено автором.
 Рисунок 5.52 – Алгоритм выбора метода прогнозирования событий

Если же нет ни признаков, способных объяснить интервалы между событиями, ни предположений о механизмах образования событий, когда события являются как бы случайно, то остаётся только использовать методы на основе потоков событий из теории случайных процессов. Однако, если выборка событий мала, что характерно для редких событий, может быть затруднительно определять характеристики закона распределения интервала времени между событиями. Вместе с тем, откладывая моду (или среднее значение, что не рекомендуется) закона распределения от предыдущего события, нет возможности узнать, отклонится ли событие в меньшую или большую сторону, на сколько отклониться. Данный метод будет давать менее обоснованный прогноз, чем все остальные методы. В этом случае лучше подобрать механизм образования событий перебором разных схем операторов.

В большинстве случаев нас интересует прогноз лишь одного будущего события. Но что, если нам интересно получить прогноз сразу двух или трёх событий? Методами классификации и регрессии такое не получится, так как наблюдаемые или искусственно сконструированные признаки меняться не будут (в том смысле, что после последнего события больше нет наблюдений и анализировать нечего). Методы на основе потоков событий будут каждый раз давать один и тот же прогноз для интервала времени между событиями (как наиболее вероятное значение), так как выборка событий, а следовательно, и распределение, не меняется. Вряд ли можно назвать корректным прогнозом простое откладывание от последнего события равных интервалов времени. Если в имеющейся выборке события шли нерегулярно, то не стоит ожидать, что в будущем они будут идти через одинаковые интервалы. Только в новом подходе на основе восстановления механизмов образования событий можно получить несколько прогнозов событий, которые справедливо будет назвать прогнозами.

Если говорить о точности прогнозов, то всё упирается в информацию, которую задействуют методы. Если наблюдаемые признаки могут объяснить

интервал времени до будущего события, то методы регрессии и классификации (машинного обучения) могут давать хороший результат. Если по выборке событий можно построить достаточно узкое распределение, в котором подавляющее большинство событий лежит в малой области, то фактическое событие будет иметь малый разброс относительно моды (среднего значения). Если имеющиеся события подчиняются одному и тому же механизму образования событий, чьи параметры можно восстановить и правильно экстраполировать на будущее, то прогноз оказывается достаточно точным. Как было показано выше, наблюдаемые признаки могут отсутствовать, искусственные признаки могут содержать недостаточно информации. Часто только событие является известным фактом. Факт появления события уже несёт информацию. Извлекать всю информацию из самого факта появления событий можно в новом предлагаемом подходе.

Можно составить следующую таблицу возможностей для разных способов прогнозирования событий, таблица 5.19.

Таблица 5.19 – Прогноз будущего выпуска облигаций

Особенность	Методы			
	Классификации	Регрессии	Потоки событий	Восстановление механизмов
Прогноз события на интервале времени	Да	Да	Да	Да
Прогноз момента времени t_i	Нет	Да	Да	Да
Прогноз характеристик события y_i	Нет	Да	Нет	Да
Способность прогнозировать исключительно по выборке событий	Да/Нет	Да/Нет	Да	Да
Способность дать прогноз сразу нескольких событий	Нет	Нет	Нет	Да
Информация о событиях задействуется напрямую	Нет	Нет	Да	Да
Обоснованность прогнозов	Да/Нет	Да/Нет	Нет	Да

Источник: составлено автором.

Так какой же метод следует использовать при прогнозе событий? На основе вышесказанного следует отдать предпочтение новому предлагаемому методу, основанному на восстановлении механизмов образования событий.

Однако в современной науке анализа данных принято использовать не один метод, а целое множество. Применяется сразу множество методов, оцениваются метрики качества созданных моделей, выбирается самая лучшая модель из множества. Тогда рекомендацией будет – использовать все методы сразу. Но тут надо сделать предупреждение, что когда используется множество методов, какой-то из методов может по случайным обстоятельствам оказаться лидирующим. В примерах выше было видно, что на одних данных одни методы машинного обучения давали хороший прогноз, но на других данных эти же методы давали уже плохой прогноз. Не было постоянства. Чем больше будет применяться разных методов и моделей, тем больше будет усугубляться эта картина. Поэтому надо с осторожностью относиться к выбору показавшего наилучший результат метода или модели из ансамбля, выбирать ту модель/метод, который работал стабильно хорошо.

Другим подходом будет использование ансамбля, когда результаты каждого из метода/моделей будут задействоваться для получения нового прогноза, превосходящего по точности каждый из методов в отдельности. Однако эта тема дальнейших исследований.

5.5 Экономический эффект

Отдельно опишем экономический эффект от разработанного подхода прогнозирования экономических событий, основанного на моделях механизмов образования события и восстановлении их параметров. Стоит заметить, что в отличие от других работ данная работа не посвящена конкретно одной экономической отрасли или деятельности. Наоборот, полученные теоретические и практические результаты можно применять практически в любой отрасли, причём в различных аспектах деятельности. Экономический эффект выражается не в большой прибыли или сокращении убытков в одной из отраслей, а в возможности широкого применения во множестве отраслей, на множестве уровней; получения прибыли или сокращения убытков множества совершенно разных организаций,

государственных или частных компаний, отдельных индивидуумов, физических лиц.

Экономический эффект от повышения точности прогноза событий можно рассчитать только косвенными способами. Повышение точности предсказания событий может принести пользу в широком спектре человеческой деятельности. Так как большинство приведённых примеров относятся к торговле, то в первую очередь нужно отметить возможность сокращения затрат на содержание запасов.

Согласно [106, с. 235], затраты на содержание запасов составляют около 37% от всех логистических издержек, которые, в свою очередь, в среднем составляют около 12% всего ВВП [106, с. 132]. Для многих компаний запасы составляют внушительную часть в совокупных активах, приблизительно 30% (от 8 до 46% в зависимости от вида деятельности компании) [106, с. 228]. Затраты на содержание запасов состоят из: 1) капитальных затрат (определяется процентной ставкой банков, по которой берётся кредит для формирования запасов или по которой могли быть размещены средства на вкладах), 2) налогов на товар (если товар облагается налогом), 3) затрат на страхование (например, когда товар взрывоопасен и необходимо страховать риски за его размещение и хранение), 4) затрат, связанных с износом и списанием (когда товар портится или морально устаревает при длительном хранении), 5) непосредственно расходов за хранение на складах [106, с. 236]. Согласно [106, с. 236], в среднем получается 20% от стоимости самих запасов. В текущих условиях (на момент написания этого раздела), когда ключевая ставка ЦБ составляет 21%, совокупные затраты на содержание запасов могут приближаться к 30%. Таким образом, годовые затраты на запасы можно рассчитать приблизительно как 30% от стоимости самих запасов. Тем самым очень значительная часть выручки заложена в затраты на содержание запасов. Сокращение запасов всего на несколько процентов может дать значительное повышение прибыли [106, с. 227].

Из расчёта 365 дней в году получаем, что 1 день содержания запасов обходится в среднем в 0,082% от стоимости запасов (товаров), и если запасы содержатся 12 дней, то затраты составляют приблизительно 1%. Отсюда можно получить приблизительную оценку экономического эффекта от увеличения точности прогноза события (когда запасы размещаются к указанному дню возникновения спроса). Например, в модели с изменяющимся максимумом прогноз по предложенному методу оказался аж на 62 дня ближе к фактическому, чем у метода XGBRegressor (ошибка с 64 дней упала до 2 дней). Если бы формирование запасов осуществлялось к указанному дню, то сокращение затрат на запасы могло составить около 5,1% (с 5,248% сократилась до 0,164%). В примере на реальных данных с поставкой биологически активной добавки Омега-3 прогноз по методу DecisionTreeRegressor оказывается хуже на 16 дней (ошибка с 20 дней сокращается до 4 дней), что может быть эквивалентно приблизительно 1,3% стоимости запасов (с 1,640% сокращается до 0,328%), а по методу GradientBoostingClassifier, когда прогнозируется интервал выпадения события, прогноз хуже на 5–8 дней, что эквивалентно приблизительно 0,41–0,66% стоимости запасов, а прогноз на основе Пуассоновского потока событий хуже на 8 дней, что эквивалентно приблизительно 0,66% стоимости запасов.

Заметим, что все указанные значения выражены в стоимости самих запасов (товаров), снижение затрат на запасы приводит к значительному повышению чистой прибыли. Также нужно отметить, что редко кто планирует запасы к указанной дате, обычно запасы планируют к началу месяца или квартала. Прогноз событий обращения клиентов за товаром позволяет более тонко управлять запасами. Отметим, что планировать можно не только запасы готовой продукции на складах и пунктах распространения, но также запасы сырья или деталей, из которых производится эта продукция, эти запасы также нужно содержать и где-то хранить. В свою очередь, на стороне поставщиков этого сырья или деталей можно также осуществлять такое планирование, то

есть задействуется вся цепочка производства и распространения товаров. Как было сказано выше, содержание запасов в среднем обходится в $0,37 \times 12 = 4,44\%$ ВВП. Если хотя бы на процент удастся сократить расходы на запасы, то это может привести к огромной экономии.

Маркетинг. Помимо сокращения затрат на содержание запасов, прогноз будущих событий, например моментов обращения клиентов за продукцией, позволяет применять маркетинговые методы. Зная, что у определённого клиента подходит к концу его запас продукции, можно заблаговременно с ним связаться и скоординировать с ним очередное пополнение запаса, возможно, с предложением дополнительной скидки. Это позволит не потерять существующего клиента, спланировать его будущее обращение, создаст возможность управлять объёмом приобретаемого клиентом товара. Прогнозирование событий может быть связано не только с приобретением товаров, но также с приобретением услуг [201]. В этом случае прогноз будущего обращения за услугой позволит заблаговременно записать клиента к нужному мастеру или спланировать загруженность мастеров. Однако прогнозировать события можно не только для своих клиентов, но и для чужих клиентов, если известны их события обращения за теми же товарами или услугами. В этом случае можно переманивать чужих клиентов к себе.

На различных торговых площадках информация о грядущем событии, например о подходе к концу запаса некоторого товара какого-нибудь крупного покупателя, может позволить игрокам использовать ситуацию в свою пользу, например, создавая дефицит и увеличивая цену, только лишь для своего последующего личного обогащения.

На уровне государства нужно также отметить определённый эффект. Это возможное снижение цены на товары, что скажется на экономическом благополучии населения; более высокий уровень снабжения территорий из-за более точного планирования потребления; обеспечение более эффективного расходования ресурсов или площадей под эти ресурсы.

Отдельно надо выделить социально-экономическую значимость. Успешное прогнозирование экономических событий может снизить экономические потери, сохранить рабочие места, обеспечить стабильность критической инфраструктуры и социальную защищённость населения. Результаты исследования могут стать фундаментом обеспечения экономической безопасности.

Конечно же, надо отметить, что сам подход к прогнозированию событий с помощью составления алгоритмических моделей механизмов их образования можно выделить в отдельное направление исследований, которое будем называть «Эвентометрика». Это направление будет ответвлением, дополнением или обобщением классической эконометрики. Происходит сдвиг парадигмы от непрерывных функций, лежащих в основе практически всего математического и экономического анализа, к дискретным событиям. Вместо алгебраических моделей строятся алгоритмические модели. Заметим, что алгоритмические модели могут быть не сводимыми к алгебраическим (к формулам), в то время как алгебраические всегда можно свести к алгоритмическим. Параметры модели могут быть не только статичными, но и динамичными. Входными данными для идентификации параметров модели являются не значения регрессоров, а дискретные события. Для определения параметров алгоритмических моделей могут потребоваться методы оптимизации недифференцируемых функций. Результатом прогнозирования с помощью построенной модели будет не значение непрерывной функции в новой точке, а дискретное событие. Отличия нового направления «Эвентометрика» от эконометрики представлены в таблице 5.20.

Появление нового направления может дать мощный толчок в исследовании экономических процессов. Позволит объяснить реальные траектории показателей, которые в основном разрывны. Получить новые качественные или количественные выводы в экономических исследованиях применительно к любым уровням любых отраслей. Причём в отличие от современного машинного обучения, на основе которого зачастую строятся

непрозрачные модели в виде чёрного ящика, в новом подходе делается акцент на интерпретируемых алгоритмических моделях, объясняющих причину и следствие появления событий. Это подтолкнет развитие в исследовании причинно-следственных связей сложных экономических систем.

Таблица 5.20 – Сравнение эвентометрики и эконометрики

-	Эконометрика	Эвентометрика
Модель	Алгебраическая $f(X) = a_0 + a_1x_1 + a_2x_2 + \dots + a_mx_m$	Алгоритмическая $Program(X, P)$
Входные данные	Наблюдения (показатели/признаки) $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$	События (t_i, y_i)
Параметры	Статичные $\{a_0, a_1, a_2, \dots, a_m\}$	Статичные или динамичные $\{P_1(t), P_2, P_3, \dots\}$
Методы	В основном МНК или градиентные методы	МНК или оптимизация недифференцируемых функций
Прогноз	Непрерывная функция $y = f(X)$	Дискретное событие (t_{n+1}, y_{n+1})

Источник: составлено автором.

Отдельный научный и практический интерес может представлять не столько прогноз экономических событий, сколько сами значения восстанавливаемых параметров моделей. Как в эконометрике параметры могут обладать важным экономическим смыслом, так и в новом направлении параметры могут иметь интерпретацию, позволяющую на основе значений этого параметра принимать значимые управленческие решения.

С помощью разработанной методологии можно анализировать экономические процессы, объекты и системы похожим образом, как это делается в эконометрике. Определять требуемые значения параметров (факторов) для появления определённых событий. Определять значимость параметров при влиянии их на характеристики событий и т.д. Это тема для дальнейших исследований.

Выводы по главе 5

1) Проведено сравнение предлагаемого подхода прогнозирования экономических событий с распространёнными методами классификации на искусственных примерах и на основе реальных данных. Такими методами являются SVC, LogisticRegression, KNeighborsClassifier, GaussianNB, DecisionTreeClassifier, RandomForestClassifier, GradientBoostingClassifier, MLPClassifier, XGBClassifier. Показано, как адаптировать и как строить модели классификации для прогнозирования событий. Показано, что новый подход зачастую превосходит современные методы классификации (в некоторых примерах ошибка сокращается с 19 до 2 дней, с 40 до 16 дней). Методы классификации на одних данных могут дать хороший прогноз, когда на других данных могут дать очень плохой прогноз.

2) Проведено сравнение предлагаемого подхода прогнозирования экономических событий с распространёнными регрессионными методами, такими как LinearRegression, SVR, KernelRidge, KNeighborsRegressor, GaussianProcessRegressor, PLSRegression, DecisionTreeRegressor, GradientBoostingRegressor, MLPRegressor, XGBRegressor. Показано, как адаптировать и как строить регрессионные модели для прогнозирования событий. На искусственных примерах и на примерах на основе реальных данных показано, что разработанный метод, основанный на восстановлении моделей механизмов образования событий, зачастую превосходит современные регрессионные методы, тем самым он является конкурентным для передовых методов машинного обучения (в некоторых примерах ошибка сократилась с 33 дней до 0, с 43 до 16 дней, с 64 до 2 дней).

3) Проведено сравнение предлагаемого подхода прогнозирования экономических событий с методами на основе потоков случайных событий, таких как поток Пуассона, поток Пальма с нормальным распределением и поток Пальма с распределением Эрланга, а также с адаптированными для получения прогноза события методами Кростона и Виллимейна (в некоторых примерах ошибка сократилась с 45 дней до 18 и с 45 до 0 дней). Отмечается

отсутствие постоянства у существующих методов, на одних данных некоторые из этих методов давали хороший прогноз, на других данных эти же методы давали уже плохой прогноз.

4) Описан экономический эффект результатов и теоретических положений, полученных в диссертационном исследовании. Отмечается возможность широкого применения во множестве отраслей и на множестве уровней. Описан ожидаемый экономический эффект от увеличения точности прогноза экономических событий на разных примерах. Увеличение точности прогноза события на 1 день приводит к среднему сокращению затрат на содержание запасов на 0,082% от их стоимости (в некоторых примерах сокращение ошибки составило с 64 до 2 дней, что соответствует 5,1% стоимости продукции). Из-за возможности применения результатов исследования на разных уровнях возможно значительное сокращение затрат, выражаемое в процентах от всего ВВП. Отмечается возможность применения маркетинговых методов при ожидании будущего события для управления своими или чужими клиентами. Описывается польза прогноза событий для торгов, для государственного обеспечения, социально-экономического благополучия. Отдельно отмечается образование нового направления исследования «Эвентометрика», в котором происходит сдвиг парадигмы от непрерывных функций к дискретным событиям.

Заключение

В данной диссертационной работе разработана новая методология прогнозирования событий, в том числе редких экономических событий. Проведён анализ существующих методов, таких как различные методы машинного обучения, выполняющие задачи классификации и регрессии, методы на основе потоков событий, и других методов, даны рекомендации по адаптации этих методов для прогнозирования событий, рассмотрены их недостатки и границы применимости. Предложен новый подход, который основан на построении и восстановлении параметров алгоритмических моделей механизмов образования событий, состоящих из последовательности операторов над внутренними переменными и параметрами. Переменные инициализируются и изменяются внутри механизма, параметры изменяются вовне механизма. Параметры могут быть динамичными, т.е. быть функцией времени. Для относительно простых моделей восстановление параметров может быть осуществлено аналитически. Для восстановления параметров сложных моделей надо использовать численные методы. Для получения прогноза будущего события параметры механизма образования событий необходимо экстраполировать на будущую перспективу, чтобы механизм функционировал в новых условиях. Прогноз события получается с помощью моделирования механизма с установленными значениями параметров.

Разработан аналитический способ восстановления по выборке событий динамических параметров механизмов, основанный на монотонной сплайновой коллокации, в которой восстанавливается функция по разным функционалам, таким как наблюдаемые с погрешностью значения искомой функции, её первые и вторые производные, а также определённые интегралы. Положительность восстанавливаемой функции обеспечивается либо способом, в котором сплайн помещается в степень экспоненты, либо с помощью сведения к задаче квадратичного программирования с итерационно добавляемыми условиями неотрицательности.

Для экстраполяции динамических параметров механизмов образования событий предлагается использовать перенос тенденций с прошлого периода времени на будущий, регрессионные методы и разложение на сумму ограниченного количества гармонических функций с помощью алгоритма Куинна-Фернандеса.

Для определения оптимальных гиперпараметров методов восстановления и экстраполяции параметров механизма образования событий используется комбинирование поиска на сетке и локальной оптимизации в границах ячейки сетки алгоритмом Нелдера-Мида. В качестве минимизируемой целевой функции выступает смесь относительных отклонений даты прогноза t и характеристики события Y с некоторым весовым коэффициентом. Причём эта ошибка усредняется по нескольким предыдущим событиям. Предложены ещё несколько метрик на основе абсолютных и относительных средних или среднеквадратичных отклонений.

Предложено несколько способов построения доверительных интервалов для прогнозируемых событий. В первом случае дисперсия даты прогноза выводится из дисперсии восстановленных параметров. Во втором способе дисперсия рассчитывается по формуле зависимости от уровня приглушения наблюдений, которая была получена в результате имитации 1000 моделей. В третьем способе доверительный интервал строится в результате моделирования механизма образования событий, в котором параметры этого механизма отклоняются от своего среднего значения на заданную величину в большую и меньшую сторону. Если фактическое будущее событие попадает в построенный доверительный интервал, то можно принять гипотезу об адекватности модели механизма образования событий, а при промахе гипотеза отвергается, что свидетельствует о неадекватности модели.

Для восстановления параметров сложных моделей механизмов образования событий разработан специальный способ, основанный на численной оптимизации. Неизвестные динамические параметры кодируются

кубическим сплайном через начальное значение, значение первой и третьей производной в самом первом узле и через значения третьей производной в последующих узлах сплайна. В качестве метода оптимизации используется комбинация поиска на сетке и метода Нелдера-Мида. Функция потерь состоит из смеси суммы квадратов относительных ошибок прогноза даты события и характеристики события, а также из штрафа на кривизну функции. Используется специальный приём со скользящим окном событий, который позволяет уменьшить размерность получающейся оптимизационной задачи.

Предложены способы подбора самих алгоритмических моделей механизмов образования событий, основанные на комбинаторном подходе и на генетическом программировании. В комбинаторном подходе при заданных ограничениях на количество операций сравнения и количество основных операторов рекурсивно строится модель из универсальных операторов, которые могут быть четырёх видов и содержать внутри себя другие универсальные операторы. В генетическом программировании терминальные листы предлагается выбирать из внутренних переменных или параметров, а среди функций обязательно должен быть оператор формирования событий.

Проведено сравнение на разных примерах нового предлагаемого подхода прогнозирования событий с существующими методами, такими как методы классификации, методы регрессии, методами на основе потоков событий и методами Кростона и Виллимейна, адаптированными для прогнозирования событий. Почти во всех примерах новый подход значительно обходит по точности существующие методы. Делается вывод о том, что новый метод является конкурентным для передовых методов машинного обучения.

Все разработанные в диссертационной работе методы и алгоритмы имеют программную реализацию на современных языках программирования. Метод восстановления функции по разным функционалам с помощью монотонной сплайновой коллокации реализован на языке Python в виде функции `FunctionalSmoothingSpline`. Разработана собственная библиотека `eventometrics` на языке Python, которая является фреймворком для

прогнозирования событий. В библиотеке реализованы основные методы и алгоритмы, предложенные в диссертационном исследовании. Библиотека имеет интерфейс, схожий с библиотекой машинного обучения `sklearn`. Основные вычисления оптимизированы с помощью `Cython`. Библиотека опубликована в репозитории `PyPi` для свободного использования. Предложенный способ восстановления параметров алгоритмических моделей механизмов образования событий с помощью численных методов реализован на языке `R`. Также на языке `R` реализован способ автоматического перебора самих моделей механизма образования событий. Для системы бизнес-аналитики `Logiном` разработан компонент, реализующий предложенный новый подход прогнозирования событий с помощью восстановления механизмов их образования.

Поставленные в диссертационном исследовании задачи были решены полностью, а исследование получилось комплексным. В результате исследования получилось заложить основы фундамента для нового направления исследований, которое было названо эвентометрика. Созданы соответствующие технологии для прогнозирования экономических событий. Разработанная методология может создать критически важный научно-практический прорыв в экономических исследованиях. В будущем возможно образование специального направления обучения студентов, которые специализировались бы на построении моделей механизмов формирования экономических событий и оценке их параметров.

Список литературы

1. Талеб, Н.Н. Чёрный лебедь. Под знаком непредсказуемости / Н. Талеб ; перевод с английского В. Сонькина [и др.]. – Москва : КоЛибри, 2010. – 525 с. : ил. – ISBN 978-5-389-00573-0.
2. Guidotti, R. A Survey of Methods for Explaining Black Box Models / A. Monreale, F. Turini, D. Pedreschi, F. Giannotti // ACM Computing Surveys. – 2018. – № 5. Volume 51. – P. 1–42. – ISSN 0360-0300.
3. Gunning, D. XAI – Explainable artificial intelligence / D. Gunning, M. Stefik, J. Choi [et al.] // Science Robotics. – 2019. – № 37. Volume 4. – ISSN 2470-9476.
4. Cuomo, S. Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What’s Next / V.S. Di Cola, F. Giampaolo, G. Rozza [et al.] // Journal of Scientific Computing. – 2022. – № 88. Volume 92. – ISSN 0885-7474.
5. Колмогоров, А.Н. Основные понятия теории вероятностей / А.Н. Колмогоров. – 2-е издание. – Москва : Наука, 1974. – 120 с. – ISBN отсутствует.
6. Ширяев, А.Н. Вероятность: в 2-х книгах / А.Н. Ширяев – 4-е издание, переработанное и дополненное — Москва : МЦНМО, 2007. – ISBN 978-5-94057-036-3.
7. Виноградов, В.В. История слов: около 1500 слов и выражений и более 5000 слов, с ними связанных / В.В. Виноградов ; Российская академия наук. Отделение литературы и языков. Научный совет «Русского языка». Институт русского языка им. В.В. Виноградова. – Москва, 1999. – 1138 с. – ISBN 5-88744-033-3.
8. Даль, В.И. Толковый словарь Даля 2008–2022 : [сайт] – Текст : электронный. – URL: <http://slovardalja.net/> (дата обращения: 19.01.2021).
9. Шеннон, К.Э. Работы по теории информации и кибернетике : сборник статей ; перевод с английского / с предисловием А.Н. Колмогорова ; под редакцией Р.Л. Добрушина и О.Б. Лупанова. – Москва : Издательство иностранной литературы, 1963. – 829 с. – ISBN отсутствует.
10. Ниров, А. Как использовать гранулярность прогноза в ритейле и от чего она зависит / Наполеон IT : [сайт]. – 2024. – Текст : электронный. – URL: <https://napoleonit.ru/blog/kak-ispolzovat-granulyarnost-prognoza-v-riteyle-i-ot-chego-ona-zavisit> (дата обращения: 26.06.2025).
11. Вентцель, Е.С. Теория случайных процессов и её инженерные приложения : учебное пособие для вузов / Е.С. Вентцель, Л.А. Овчаров. –

2-е издание, стереотипное. – Москва : Высшая школа, 2000. – 383 с.: ил. – ISBN 5-06-003831-9.

12. Савельев, А.А. Теория пространственных точечных процессов в задачах экологии и природопользования (с применением пакета R) : учебное пособие / А.А. Савельев, С.С. Мухарамова, Н.А. Чижикова, А.Г. Пилюгин. – Казань : Издательство Казанского университета, 2014. – 146 с. – ISBN 9785-00019-311-2.

13. Embrechts, P. Modelling extreme events for insurance and finance. Application of mathematics, stochastic modelling, and applied probability / P. Embrechts, C. Kluppelberg, T. Mikosch. – New York : Springer Berlin, Heidelberg, 1997. – 648 p. – ISBN 978-3-540-60931-5.

14. Carreno, A. Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework / A. Carreno, I. Inza, J.A. Lozano // Artificial Intelligence Review. – 2020. – Volume 53. – P. 3575–3594. – ISSN 0269-2821.

15. Halim, S.Z. Time-trend analysis of offshore fire incidents using nonhomogeneous Poisson process through Bayesian inference / S.Z. Halim, N. Quaddus, H. Pasman // Process Safety and Environmental Protection. – 2021. – № 147. – P. 421–429. – ISSN 1744-3598.

16. Физика и экономика. Гносеологическая разница и её проявление в IT / Хабр : [сайт]. – 2019. – Текст : электронный. – URL: <https://habr.com/ru/articles/460833/> (дата обращения: 26.06.2025).

17. Introduction to Apache Kafka / Apache Kafka : [сайт]. – 2024. – Текст : электронный. – URL: <https://kafka.apache.org/intro> (дата обращения: 26.06.2025).

18. Predictive Analytics — всё, что нужно знать (обзор ключевых моментов / Хабр : [сайт]. – 2024. – Текст : электронный. – URL: <https://habr.com/ru/articles/808423/> (дата обращения: 26.06.2025).

19. Howson, C. Successful Business Intelligence, Second Edition: Unlock the Value of BI & Big Data / C. Howson. – 2nd Edition. New York : McGraw Hill, 2013. – 336 p. – ISBN 978-0071809184.

20. Berg, A. The Challenges of Predicting Economic Crises / A. Berg, C. Pattillo ; International Monetary Fund : [сайт]. – 2000. – Текст : электронный. – URL: <https://www.imf.org/external/pubs/ft/issues/issues22/> (дата обращения: 26.06.2025).

21. Lun, Xi. A machine-learning-dependent early warning system for corporate financial crises / Xi Lun, Y. Wang, Q. Fu. [et al.] // Journal of Computational Methods in Sciences and Engineering. – 2025. – № 11(25). – ISSN 1472-7978.

22. Вапник, В.Н. Об одном классе перцептронов / В.Н. Вапник, А.Я. Червоненкис // Автоматика и телемеханика. – № 1(25). – 1964. – С. 112–120. – ISSN 0005-2310.
23. Burges, C.A Tutorial on Support Vector Machines for Pattern Recognition / C. Burges // Data Mining and Knowledge Discovery. – Том 2. – 1998. P. 121–167. – ISSN 1384-5810.
24. Коротеев, М.В. Основы машинного обучения на Python : учебник / М.В. Коротеев. – Москва : КНОРУС, 2024. – 432 с. – ISBN 978-5-406-12673-8.
25. Писарук, Н.Н. Исследование операций / Н.Н. Писарук. – Минск : БГУ. – 289 с. – ISBN отсутствует. – Текст : электронный. – URL: <https://pisaruk.narod.ru/books/OR.pdf> (дата обращения: 14.05.2024).
26. Katta, G.M. Linear complementarity linear and nonlinear programming. Internet Edition / G.M. Katta, Yu Feng-Tien // Web archive. – 2010. – 629 p. – Текст : электронный. – URL: https://web.archive.org/web/20100605193200if_/http://ioe.engin.umich.edu/people/fac/books/murty/linear_complementarity_webbook/ (дата обращения: 14.05.2024).
27. Wright, M.H. The interior-point revolution in optimization: History, recent developments, and lasting consequences / M.H. Wright // Bulletin of the American Mathematical Society. – 2004. – Volume 42. – P. 39–57. – ISSN 0273-0979.
28. Cramer, J.S. The origins of logistic regression / J.S. Cramer // Tinbergen Institute Working Paper. – 2002. – № 119/4. – Текст : электронный. – DOI: 10.2139/ssrn.360300. – URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=360300 (дата обращения: 25.11.2022).
29. Berkson, J. Application of the Logistic Function to Bio-assay / J. Berkson // Journal of the American Statistical Association. – 1944. – № 227. – Volume 9. – P. 357–365. – ISSN 0162-1459.
30. Bliss, C.I. The method of probits / C.I. Bliss // Science. – 1934. – № 2037. Volume 79. – P. 38–39. – ISSN 0036-8075.
31. Bliss, C.I. The method of probits – A correction / C.I. Bliss // Science. – 1934. – № 2053. Volume 79. – P. 409–410. – ISSN 0036-8075.
32. Vannucci, M. Classification of unbalanced datasets and detection of rare events in industry: Issues and solutions / M. Vannucci, V. Colla // Communications in Computer and Information Science. – 2016. – № 629. – P. 337–351. – ISSN 1865-0929.
33. Hosmer, D.W. Applied logistic regression / D.W. Hosmer, S. Lemeshow. – Chichester : Wiley, 2000. – 2nd ed. – ISBN 9781118548394.

34. Goodfellow, I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. – MIT Press, 2016. – ISBN 0262035618. – Текст : электронный. – URL: <http://www.deeplearningbook.org> (дата обращения: 25.11.2022).
35. Hastie, T. The Elements of Statistical Learning. Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman. – 2nd ed. – New York : Springer-Verlag, 2009. – ISBN 978-0-387-84857-0.
36. Kashnitsky, Y. Открытый курс машинного обучения. Тема 4. Линейные модели классификации и регрессии / Y. Kashnitsky // Хабр : [сайт]. – 2017. – Текст : электронный. – URL: <https://habr.com/ru/company/ods/blog/323890/> (дата обращения: 25.11.2022).
37. Cover, T. Nearest neighbor pattern classification / T. Cover, P. Hart // IEEE Transactions on Information Theory. – 1967. – № 1. Volume 13. – P. 21–27. – ISSN 1557-9654.
38. Fix, E. Discriminatory Analysis. Nonparametric Discrimination / E. Fix, J.L.Jr. Hodges ; University of California; Berkley // USAF School of Aviation Medicine; Randolph Field; Texas; Project 21-49-004; Report 4; Contact AF41(128)-31. – California : Berkley, 1951. – ISBN отсутствует.
39. Pekalska, E. Who invented the nearest neighbor rule? / E. Pekalska, B. Duin (P.W. Robert) // 37 Steps. Understanding, teaching and developing tools for pattern recognition : [сайт]. – 2014. – Текст : электронный. – URL: <https://37steps.com/4370/nn-rule-invention/> (дата обращения: 25.11.2022).
40. Pelillo, M. Alhazen and the nearest neighbor rule / M. Pelillo // Pattern Recognition Letters. – 2014. – Volume 38. – P. 34–37. – ISSN 0167-8655.
41. Метрики дистанции библиотеки scikit-learn / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.DistanceMetric.html#sklearn.metrics.DistanceMetric.get_metric (дата обращения: 14.05.2024).
42. Lindberg, D.C. Theories of Vision from Al-Kindi to Kepler / D.C. Lindberg. – Chicago : The University of Chicago Press, 1976. – P. 324. – ISBN 0226482340, 9780226482347.
43. Bentley, J.L. Multidimensional binary search trees used for associative searching / J.L. Bentley // Communications of the ACM. – 1975. – № 9. Volume 18. – P. 509–517. – ISSN 0001-0782.
44. Omohundro, S.M. Five Balltree Construction Algorithms / S.M. Omohundro // International Computer Science Institute Technical Report. – 1989. – P. 1–23. – ISBN отсутствует.
45. Singh, S. Forecasting using Fuzzy Nearest Neighbor Method / S. Singh // Proc. 6th International Conference on Fuzzy Theory and Technology, Fourth Joint

Conference on Information Sciences (JCIS'98) (23–28 October, 1998). – North Carolina, 1998. – Volume 1. – P. 80–83. – ISBN отсутствует.

46. Breiman, L. Classification and Regression Trees / L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone // Belmont : Wadsworth, 1984. – 368 p. – ISBN 978-0412048418.

47. Loh, W. Fifty Years of Classification and Regression Trees / W.Y. Loh // International Statistical Review. – 2014. – № 3. Volume 82. – P. 329–348. – ISSN 0306-7734.

48. Morgan, J.N. Problems in the analysis of survey data, and a proposal / J.N. Morgan, J.A. Sonquist // Journal of the American Statistical Association. – 1963. – Volume 58. – P. 415–434. – ISSN 0162-1459.

49. Who invented the decision tree? / Cross Validated : [сайт]. – 2017. – Текст : электронный. – URL: <https://stats.stackexchange.com/questions/257537/who-invented-the-decision-tree> (дата обращения: 22.05.2024).

50. Belson, W.A. Matching and Prediction on the Principle of Biological Classification / W.A. Belson // Journal of the Royal Statistical Society. Series C (Applied Statistics). – 1959. – № 2. Volume 8. – P. 65–75. – ISSN 0162-1459.

51. Breiman, L. Bagging predictors / L. Breiman // Machine Learning. – 1996. – № 2. Volume 26. – P. 123–140. – ISSN 0885-6125.

52. Breiman, L. Random Forests / L. Breiman // Machine Learning. – 2001. – Volume 45. – P. 5–32. – ISSN 0885-6125.

53. Landwehr, N. Logistic Model Trees / N. Landwehr, M. Hall, E. Frank // Machine Learning. – 2005. – Volume 59. – P. 161–205. – ISSN 0885-6125.

54. Schapire, R.E. The Boosting Approach to Machine Learning: An Overview / R.E. Schapire // Nonlinear Estimation and Classification. Lecture Notes in Statistics / D.D. Denison, M.H.Hansen, C.C. Holmes [et al.] (eds.). – 2003. – Springer, New York, NY. 2003. – Volume 171. – P. 149–171. – ISBN 978-0-387-95471-4.

55. Kearns, M. Learning Boolean formulae or finite automata is as hard as factoring / M. Kearns, L.G. Valiant // Technical Report TR-14-88. – Harvard University, Aiken Computation Laboratory, August 1988. – ISBN отсутствует.

56. Kearns, M. Cryptographic limitations on learning Boolean formulae and finite automata / M. Kearns, L.G. Valiant // Journal of the Association for Computing Machinery. – 1994. – № 1. Volume 41. – P. 67–95. – ISSN 00045411.

57. Schapire, R.E. The strength of weak learnability / R. E. Schapire // Machine Learning. – 1990. – № 2. Volume 5. – P. 197–227. – ISSN 08856125.

58. Freund, Y. Boosting a weak learning algorithm by majority / Y. Freund // *Information and Computation*. – 1995. – № 2. Volume 121. – P. 256–285. – ISSN 0890-5401.

59. Freund, Y. A decision-theoretic generalization of on-line learning and an application to boosting / Y. Freund, R.E. Schapire // *Journal of Computer and System Sciences*. – 1997. – № 1. Volume 55. – P. 119–139. – ISSN 0022-0000.

60. Friedman, J. Greedy Function Approximation: A Gradient Boosting Machine / J. Friedman // Technical report, Department of Statistics, Stanford University. – 1999.

61. Friedman, J. Greedy Function Approximation: A Gradient Boosting Machine / J. Friedman // *The Annals of Statistics*. – 2001. – № 5. Volume 29. – P. 1189 – 1232. – ISSN 0090-5364.

62. Rosenblatt, F. The Perceptron: A Probabilistic Model For Information Storage And Organization in the Brain / F. Rosenblatt // *Psychological Review*. – 1958. – № 6. Volume 65. – P. 386–408. – ISSN 0033-295X.

63. Neural network (machine learning) // Википедия. Свободная энциклопедия : [сайт]. – Текст : электронный. – URL: [https://en.wikipedia.org/wiki/Neural_network_\(machine_learning\)](https://en.wikipedia.org/wiki/Neural_network_(machine_learning)) (дата обращения: 27.05.2024).

64. History of artificial neural networks // Википедия. Свободная энциклопедия : [сайт]. – Текст : электронный. – URL: https://en.wikipedia.org/wiki/History_of_artificial_neural_networks (дата обращения: 27.05.2024).

65. Борокин, А. Основные функции активации в нейронных сетях / А. Борокин // Your To – Do : [сайт]. – 15 января 2024. – Текст : электронный. – URL: <https://www.yourtodo.ru/posts/osnovnyie-funktsii-aktivatsii-v-nejronnyih-setyah/> (дата обращения: 27.05.2024).

66. Каширина, И.Л. Исследование и сравнительный анализ методов оптимизации, используемых при обучении нейронных сетей / И.Л. Каширина, М.В. Демченко // Вестник воронежского государственного университета. Серия: системный анализ и информационные технологии. – 2018. – № 4. – С. 123–132. – ISSN 1995-5499.

67. Siarshai. Методы оптимизации нейронных сетей // Хабр : [сайт]. – 4 января 2017. – Текст : электронный. – URL: <https://habr.com/ru/articles/318970/> (дата обращения: 28.05.2024).

68. Choe, W. Detection of rare events by neural networks / W. Choe, O.K. Ersoy, M. Blina // *Intelligent Engineering Systems Through Artificial Neural Networks*. – 1998. – P. 5–10. – ISBN отсутствует.

69. Kaiser, L. Learning to remember rare events / L. Kaiser, A. Roy, O. Nachum, S. Bengio // Conference Track Proceedings : 5th International Conference on Learning Representations (ICLR) 2017 (France, Toulon, April 24–26, 2017). – France : Toulon, 2017. – ISBN отсутствует.

70. Cross-validation: evaluating estimator performance // SciKit-Learn. Machine Learning in Python : [сайт]. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation (дата обращения: 30.05.2024).

71. Падве, В.А. Метод наименьших квадратов (история и развитие) / В.А. Падве, Б.Т. Мазуров // Интерэкспо ГЕО-Сибирь. – 2017. – № 1. Том 1, – С. 50–154. – ISSN 2618-981X.

72. Merriman, M. On the History of the Method of Least Squares / M. Mansfield // The Analyst. – 1877. – № 2. Volume 4. – P. 33–36. – ISSN отсутствует.

73. Legendre, A. M. Nouvelles méthodes pour la détermination des orbites des comètes; avecun supplément contenant divers perfectionnemens de ces methodes et lauf application aux deux comètes de 1803/ A.M. Legendre. – Paris : Courcier, 1806. – ISBN отсутствует.

74. Theoria motus corporum coelestium in sectionibus conicis solem ambientium / C.F. Gauss. – Hamburg, 1809. – ISBN отсутствует.

75. Избранные геодезические сочинения. Том 1 / К.Ф. Гаусс ; под редакцией, с введением и комментариями Г.В. Багратуни ; перевод с латинского и немецкого Н.Ф. Булаевского. – Москва : Геодиздат, 1958. – 152 с. – ISBN отсутствует.

76. Adrain, R. Research concerning the probabilities of the errors which happen in making observations / R. Adrain // The Analyst. – 1808. – № 4. – P. 93–109. – ISSN отсутствует.

77. Гнеденко, Б.В. Математические методы в теории надежности / Б.В. Гнеденко, Ю.К. Беляев, А.Д. Соловьев. – Москва : Наука, 1965. – 524 с. : ил. – ISBN отсутствует.

78. Барлоу, Р. Математическая теория надежности / Р. Барлоу, Ф. Прошан ; перевод с английского И.А. Ушакова ; под редакцией Б.В. Гнеденко. – Москва : Советское радио, 1969. – 488 с.: ил. – ISBN отсутствует.

79. Лукинский, В. Методы управления запасами: расчёт показателей запаса для товарных групп, относящихся к редким событиям (часть I) / В. Лукинский, Д. Замалетдинова // Логистика. – 2015. – № 1. Том 98. – С. 28–33. – ISSN 2219-7222.

80. Лукинский, В. Методы управления запасами: расчёт показателей запаса для товарных групп, относящихся к редким событиям (часть II) /

В. Лукинский, Д. Замалетдинова // Логистика. – 2015. – № 2. Том 99. – С. 24–27. – ISSN 2219-7222.

81. Вожжов, А.П. Формирование страховых запасов предприятия при пуассоновском характере поступающих и выдаваемых потоков / А.П. Вожжов, О.В. Луняков, С.П. Вожжов // Экономика и управление: теория и практика. – 2015. – № 1. Том 1. – С. 30–35. – ISSN 2412-8376.

82. Henderson, D.J. Applied Nonparametric Econometrics / D.J. Henderson, C.F. Parmeter. – New York : Cambridge University Press, 2015. – 367 p. – ISBN 978-1-107-01025-3.

83. Multivariate Density Estimation: Theory, Practice, and Visualization / D.W. Scott. – 2nd ed. – New Jersey : Wiley, 2015. – 384 p. – ISBN 978-0-471-69755-8.

84. Li, Q. Nonparametric Econometrics: Theory and Practice / Q. Li, J.S. Racine. – Princeton University Press, 2007. – 768 p. – ISBN 978-0-691-12161-1.

85. Syntetos, A.A. Forecasting of compound Erlang demand / A.A. Syntetos, M.Z. Babai, S. Luo // Journal of the Operational Research Society. – 2015. – № 12. – Volume 66. – P. 2061–2074. – ISSN 0160-5682.

86. Shale, E.A. Forecasting for intermittent demand: the estimation of an unbiased average / E.A. Shale, J.E. Boylan, F.R. Johnston // Journal of the Operational Research Society. – 2006. – № 57. – P. 588–592. – ISSN 0160-5682.

87. Nadaraya, E.A. On Estimating Regression / E.A. Nadaraya // Teoriya Veroyatnosti i Primeneniya. – 1964. – № 1 (9). – P. 157–159. – ISSN 0040-361X; Theory of Probability and its Applications. – 1964. – № 1 (9). – P. 141–142. – ISSN 0040-585X.

88. Watson, G.S. Smooth Regression Analysis / G.S. Watson // Sankhyā: The Indian Journal of Statistics, Series A. – 1964. – № 4. – Volume 26. – P. 359–372. – ISSN 0581-572x.

89. Croston, J.D. Forecasting and stock control for intermittent demands / J.D. Croston // Journal of the Operational Research Society. – 1972. – № 3. – Volume 23. – P. 289–303. – ISSN 1476-9360.

90. Qingzheng, X. A Review of Croston's method for intermittent demand forecasting / X. Qingzheng, W. Na, S. Heping // Proceedings – 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2012. – 2012. – P. 1468–1472. – ISBN 978-1-4673-0025-4.

91. Recycling supply chain optimization using simulation software / Anylogic : [сайт]. – 2022. – Текст : электронный. – URL: <https://www.anylogic.com/resources/case-studies/recycling-supply-chain-optimization-using-simulation->

software/?utm_source=youtube&utm_medium=social-organic (дата обращения: 05.06.2024).

92. Optimization of a Reverse Logistics Supply Chain with an end-to-end Digital Twin Solution : [видеозапись] // Anylogic : официальный аккаунт на YouTube. Время воспроизведения: 29 ноября 2022. – URL: <https://www.youtube.com/watch?v=jIaTDGSyIeA> (дата обращения: 05.06.2024).

93. Efron, B. Bootstrap methods: another look at jackknife / B. Efron // The Annals of Statistics. – 1979. – № 1. Volume 7. – P. 1–26. – ISSN 0090-5364.

94. Efron, B. A leisurely look at the bootstrap, the jackknife, and cross-validation / B. Efron, G. Gong // American Statistician. – 1983. – № 1. Volume 37. – P. 36–48. – ISSN 1537-2731.

95. Efron, B. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy / B. Efron, R. Tibshirani // Statistical Science. – 1986. – № 1. Volume 1. – P. 54–75. – ISSN 0883-4237.

96. Efron, B. An Introduction to the Bootstrap / B. Efron, R.J. Tibshirani. – New York : Chapman & Hall, 1993. – 456 p. – ISBN 0-412-04231-2.

97. Большой энциклопедический словарь [Текст] / Главный редактор А.М. Прохоров. — 2-е издание, переработанное и дополненное. — Москва : Большая Российская энциклопедия; Санкт-Петербург : Норинт, 2000. — 1434 с.: ил. — ISBN 5-85270-160-2, 5-7711-0004-8.

98. Willemain, T.R. A new approach to forecasting intermittent demand for service parts inventories / T.R. Willemain, C.N. Smart, H.F. Schwarz // International Journal of Forecasting. – 2004. – № 3. Volume 20. – P. 375–387. – ISSN 0169-2070.

99. Willemain, T.R. Forecasting intermittent demand in manufacturing: a comparative evaluation of Croston's method / T.R. Willemain, C.N. Smart, J.H. Shockor, P.A. DeSautels // International Journal of Forecasting. – 1994. – № 10. – P. 529–538. – ISSN 0169-2070.

100. Kaya, G.O. Intermittent demand forecasting: a guideline for method selection / G.O. Kaya, M. Sahin, O. Demirel // Sadhana – Academy Proceedings in Engineering Sciences. – 2020. – № 1. Volume 45. – P. 45–51. – ISSN 0973-7677.

101. Pince, C. Intermittent demand forecasting for spare parts: A Critical review / C. Pince, L. Turrini, J. Meissner // Omega (United Kingdom). – 2021. – Volume 105. – P. 102513. – ISSN 0305-0483.

102. Burioni, R. Rare events in stochastic processes with sub-exponential distributions and the big jump principle / R. Burioni, A. Vezzani // Journal of Statistical Mechanics: Theory and Experiment. – 2020. – № 034005. – ISSN 1742-5468.

103. Holl, M. Big jump principle for heavy-tailed random walks with correlated increments / M. Holl, E. Barkai // *The European Physical Journal B*. – 2021. – № 216. Volume 94. – ISSN 1434-6028.

104. Vezzani, A. Single-big-jump principle in physical modeling / A. Vezzani, E. Barkai, R. Burioni // *Physical Review E*. – 2019. – № 012108. Volume 100. – ISSN 2470-0045.

105. Embrechts, P. Modelling extremal events for insurance and finance / P. Embrechts, C. Klüppelberg, T. Mikosch. – Berlin : Springer-Verlag, 1997. – 648 p. – ISBN 978-3-540-60931-5.

106. Бауэрсок, Д.Д. Логистика: интегрированная цепь поставок / Д.Д. Бауэрсок, Д.Д. Клосс. – 2-е издание. – Москва : ЗАО «Олимп-Бизнес», 2008. – 640 с.: ил. с. – ISBN 978-5-9693-0124-5.

107. Supervised learning / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/supervised_learning.html (дата обращения: 04.07.2024).

108. Support Vector Machines / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: <https://scikit-learn.org/stable/modules/svm.html> (дата обращения: 04.07.2024).

109. Kernel ridge regression / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/kernel_ridge.html (дата обращения: 04.07.2024).

110. Nearest Neighbors / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: <https://scikit-learn.org/stable/modules/neighbors.html> (дата обращения: 04.07.2024).

111. Gaussian Process Regression / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/gaussian_process.html (дата обращения: 04.07.2024).

112. Cross decomposition / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/cross_decomposition.html (дата обращения: 04.07.2024).

113. Decision Trees / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: <https://scikit-learn.org/stable/modules/tree.html> (дата обращения: 04.07.2024).

114. Ensembles: Gradient boosting, random forests, bagging, voting, stacking / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: <https://scikit-learn.org/stable/modules/ensemble.html> (дата обращения: 04.07.2024).

115. Neural network models (supervised) / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/neural_networks_supervised.html (дата обращения: 04.07.2024).

116. LightGBM documentation / LightGBM : [сайт]. – 2024. – Текст : электронный. – URL: <https://lightgbm.readthedocs.io/en/latest/index.html> (дата обращения: 04.07.2024).

117. XGBoost Documentation / DMLC XGBoost : [сайт]. – 2024. – Текст : электронный. – URL: <https://xgboost.readthedocs.io/en/stable/index.html> (дата обращения: 04.07.2024).

118. Numerical differentiation of noisy time series data in python / Derivative documentation : [сайт]. – 2024. – Текст : электронный. – URL: <https://derivative.readthedocs.io/en/stable/> (дата обращения: 04.07.2024).

119. Haas, J.S. derivative documentation / GitHub : [Исходный код]. – 2023. – Текст : электронный. – URL: <https://github.com/andgoldschmidt/derivative/blob/c90ef7c8fdfe41c1e15a95a2f2e5bd7aa1358d50/derivative/dglobal.py> (дата обращения: 04.07.2024).

120. Киреев, В.И. Интегральный метод приближения функций алгебраическими многочленами и биквадратными сплайнами / В.И. Киреев // Вестник Московского авиационного института. – 1994. – № 1. Том 1. – С. 48–58. – ISSN 0869-6101.

121. Киреев, В.И. Полиномиальные интегродифференциальные одномерные и двумерные сплайны / В.И. Киреев, Т.К. Бирюкова // Вычислительные технологии. – 1998. – № 3. Том 3. – С. 19–34. – ISSN 1560-7534.

122. Киреев, В.И. Интегродифференциальный метод обработки информации и его применение в численном анализе / В.И. Киреев, Т.К. Бирюкова. – Москва : ИПИ РАН, 2014. – 267 с. – ISBN 978-5-91993-040-2.

123. Бирюкова, Т.К. Методы численного дифференцирования и восстановления сеточных функций по интегралам, основанные на интегродифференциальных сплайнах / Т.К. Бирюкова, В.И. Киреев, М.М. Гершкович // Системы компьютерной математики и их приложения : материалы XVII Международной научной конференции. – Смоленск, 2016. – С. 106–112. – ISSN отсутствует.

124. Dagnino, C. Spline quasi-interpolating projectors for the solution of nonlinear integral equations / C. Dagnino, A. Dallefrate, S. Remogna // Journal of Computational and Applied Mathematics. – 2019. – Volume 354. – P. 360–372. – ISSN 0377-0427.

125. Самойлова, Э.Н. Сплайновые приближения решения сингулярного интегродифференциального уравнения / Э.Н. Самойлова // Известия высших учебных заведений. Математика. – 2001. – № 11 (474). – С. 35–45. – ISSN 0021-3446.

126. Zemlyanova, A.Y. A new B-spline collocation method for singular integro-differential equations of higher orders / A.Y. Zemlyanova, A. Machina // Journal of Computational and Applied Mathematics. – 2020. – Volume 380. – P. 112949. – ISSN 0377-0427.

127. Boor, C.D. A Practical Guide to Splines [Текст] / C.D. Boor. – Revised ed. – Berlin : Springer Verlag, 2001. – 366 p. – ISBN 0-387-95366-3.

128. Федорова, О.П. Об одном подходе к приближению функции сплайнами / О.П. Федорова // Вестник Томского государственного университета. Математика и механика. – 2008. – № 2 (3). – С. 61–66. – ISSN 1998-8621.

129. Федорова, О.П. Метод построения сплайна, сохраняющего интеграл функции двух переменных по области её задания / О.П. Федорова // Научный альманах. – 2016. – № 1-3 (15). – С. 31–35. – eISSN 2411-7609.

130. Ahlberg, J.H. The Theory of Splines and Their Applications / J.H. Ahlberg, E.N. Nilson, J.L. Walsh. – New York : Academic Press, 1967. – P. 284. – ISBN 978-0-12-044750-3.

131. Schoenberg, I.J. Contributions of the problem of approximation of equidistant data by analytic function / I. Schoenberg // Quarterly of Applied Mathematics. – 1946. – № 4(1). – P. 45–99, 112–141. – ISSN 1552-4485.

132. Кто первым записал сплайн-функцию? Как техническая механика «предвосхитила» достижения теории сплайнов / Хабр : [сайт]. – 2024. – Текст : электронный. – URL: <https://habr.com/ru/articles/797545/> (дата обращения: 07.07.2025).

133. Бубнов, И.Г. Строительная механика корабля. Часть I / И.Г. Бубнов – Санкт-Петербург : типография Морского министерства, 1912. – 331с. – ISBN отсутствует.

134. Whittaker, E.T. On a new method of graduation / E.T. Whittaker // Proceedings of the Edinburgh Mathematical Society. – 1923. – № 41. – P. 63–75. – ISBN отсутствует.

135. Schumaker, L.L. Spline function: basic theory / L.L. Schumaker. – New York : Wiley, 1981. – 553 p. – ISBN 978-0-47176-475-5

136. Kimeldorf, G. Some results on Tchebychefian spline functions / G. Kimeldorf, G. Wahba // Journal of mathematical analysis and applications. – 1971. – Volume 33. – P. 82–95. – ISSN 0022-247X.

137. Green, P.J. Nonparametric regression and generalized linear models. A roughness penalty approach / P.J. Green, B.W. Silverman. – New York : Chapman & Hall, 1994. – 194 p. – ISBN 0-412-30040-0.

138. Тихонов, А.Н. Методы решения некорректных задач / А.Н. Тихонов, В.Я. Арсенин. – 2-е издание. – Москва : Наука, 1979. – 285 с. – ISBN отсутствует.

139. Hansen, P.C. Analysis of Discrete Ill-Posed Problems by Means of the L-Curve / P.C. Hansen // *SIAM Rev.* – 1992. – № 4. Volume 34. – P. 561–580. – ISSN 0036-1445.

140. Hansen, P.C. The L-Curve and its Use in the Numerical Treatment of Inverse Problems / P.C. Hansen // In book: *Computational Inverse Problems in Electrocardiology*; edited by: P.R. Johnston, Griffith University, Australia – Southampton : WIT Press, 2001. – P. 119–142. – ISBN 978-1-85312-614-7.

141. Craven, P. Smoothing noisy data with spline functions – Estimating the correct degree of smoothing by the method of generalized cross-validation / P. Craven, G. Wahba // *Numerische Mathematik.* – 1978. – № 31. Volume 4. – P. 377–403. – ISSN 0945-3245.

142. Morozov, A. Methods for solving incorrectly posed problems / A. Morozov. – New York : Springer-Verlag, 1984. – 257 p. – ISBN 978-0-387-96059-3.

143. Mueller, J.J. Linear and Nonlinear Inverse Problems with Practical Applications / J.J. Mueller, S. Siltanen. – Philadelphia : SIAM, 2012. – 338 p. – ISBN 978-1-61197-233-7.

144. Reinsch, Ch.H. Smoothing by spline function / Ch.H. Reinsch // *Numerische Mathematik.* – 1967. – № 10. – P. 177–183. – ISSN 0945-3245.

145. Elfving, T. An Algorithm for Computing Constrained Smoothing Spline Functions / T. Elfving, L. Andersson // *Numerische Mathematik.* – 1988. – № 52. – P. 583–595. – ISSN 0945-3245

146. Vlachkova, K. A Newton-type algorithm for solving an extremal constrained interpolation problem / K. Vlachkova // *Numerical Linear Algebra With Applications.* – 2000. – № 7. – P. 133–146. – ISSN 1099-1506.

147. Greiner, H. A survey on univariate data interpolation and approximation by splines of given shape / H. Greiner // *Mathematical and Computer Modelling.* – 1991. – № 15. – P. 97–106. – ISSN 0895-7177.

148. Ramsay, J.O. Monotone regression splines in action / J.O. Ramsay // *Statistical Science.* – 1988. – № 3. Volume 4. – P. 425–461. – ISSN 0883-4237.

149. Wright, I.W. Isotonic, Convex and Related Splines / I.W. Wright, E.J. Wegman // *The Annals of Statistics.* – 1980. – № 8(5). – P. 1023–1035. – ISSN отсутствует.

150. Curry, H.B. On Polya frequency functions. IV. The fundamental spline functions and their limits / H.B. Curry, I.J. Schonberg // *Journal d'Analyse Mathématique*. – 1966. – № 17. – P. 71–107. – ISSN 0021-7670. – DOI: 10.1007/BF02788653.

151. Ramsay, J.O. *Functional Data Analysis. Second Edition* / J.O. Ramsay, B.W. Silverman // *Springer Series in Statistics*. – 2005. – P. 432. – ISBN 978-0387-40080-8.

152. Nagahara, M. Monotone smoothing splines using general linear systems / M. Nagahara, C.F. Martin // *Asian Journal of Control*. – 2013. – № 15(2). – P. 461–468. – ISSN 1561-8625.

153. Nocedal, J. *Numerical Optimization* / J. Nocedal, S.J. Wright. – New-York : Springer, 2006. – P. 651. – ISBN 0-387-98793-2.

154. Katta, G.M. 2006. *Linear complementarity linear and nonlinear programming. Internet Edition* / G.M. Katta, Feng-Tien Yu. – 2006. – 629 p. – Текст : электронный. – URL: https://web.archive.org/web/20100605193200if_/http://ioe.engin.umich.edu/people/fac/books/murty/linear_complementarity_webbook/ (дата обращения: 10.03.2024).

155. Caron, S. *Quadratic programming in Python* / Stéphane Caron Blog : [сайт]. – Текст : электронный. – URL: <https://scaron.info/blog/quadratic-programming-in-python.html> (дата обращения: 10.03.2024).

156. Chan, V. Efficient estimation of smoothing spline with exact shape constraints / V. Chan, K.W. Tsui, Y. Wei [et al.] // *Statistical Theory and Related Fields*. – 2020. – № 1(5). – DOI: 10.1080/24754269.2020.1722604.

157. Diakin. *Практическое применение преобразования Фурье для анализа сигналов. Введение для начинающих* // Хабр : [сайт]. – 2015. – Текст : электронный. – URL: <https://habr.com/ru/post/269991/> (дата обращения: 17.11.2020).

158. Quinn, B.G. A Fast Efficient Technique for the Estimation of Frequency / B.G. Quinn, J.M. Fernandes // *Biometrika*. – 1991. – № 3. Volume 78. – P. 489–497. – ISSN 0006-3444.

159. Quinn, B.G. *The Estimation and Tracking of Frequency* / B.G. Quinn, E.J. Hannan. – Cambridge : Cambridge University Press, 2001. – 278 p. – ISBN 0-521-80446-9.

160. *Beautiful, but Lost Algorithms – Episode 1 – Quinn & Fernandez Algorithm (1991)* // AlephZero : аккаунт на YouTube : [видеозапись]. Время воспроизведения: 11 июля 2024. – URL: <https://www.youtube.com/watch?v=jfTzsw1U3Q8> (дата обращения: 11.07.2024).

161. Vladimir. Экстраполяция цен методом Фурье – индикатор для MetaTrader 5 / Vladimir, gpwr // MQL5 community : [сайт]. – 2010. – Текст : электронный. – URL: <https://www.mql5.com/ru/code/130> (дата обращения: 17.11.2020).

162. Nelder, J.A. A Simplex Method for Function Minimization / J.A. Nelder, R. Mead // The Computer Journal. – 1965. – № 4. Volume 7. – P. 308–313. – ISSN 0010-4620.

163. Lagarias, J.C. Convergence properties of the Nelder-Mead simplex method in low dimensions / J.C. Lagarias, J.A. Reeds, M.H. Wright, P.E. Wright // SIAM Journal on Optimization. – 1998. – № 1. Volume 9. – P. 112–147. – ISSN 1052-6234.

164. Kelley, C.T. Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition / C.T. Kelley // SIAM Journal on Optimization. – 1999. – № 1. Volume 10. – P. 43–55. – ISSN 1052-6234.

165. Goldberg, D.E. Genetic algorithms in search, optimization, and machine learning / D.E. Goldberg. – Addison-Wesley, 1989. – 432 p. – ISBN 0-201-15767-5.

166. Саймон, Д. Алгоритмы эволюционной оптимизации / Д. Саймон ; перевод с английского А.В. Логунова. – Москва : ДМК Пресс, 2020. – 1002 с. – ISBN 978-5-97060-707-7.

167. Laguna, M. Scatter Search – Methodology and Implementations in C / M. Laguna, R. Marti. – Boston : Kluwer Academic Publishers, 2003.– 291 p. – ISBN 978-1-4020-7376-2.

168. Laguna, M. Scatter Search / M. Laguna, R. Marti // Metaheuristic Procedures for Training Neural Networks ; ed. E. Alba, R. Marti. – New York : Springer, 2006.– P. 139–152. – ISBN 978-0-387-33415-8.

169. Banzhaf, W. Genetic Programming ~ An introduction : on the automatic evolution of computer programs and its application / W. Banzhaf, P. Nordin, R.E. Keller, F.D. Francone. – San Francisco: Morgan Kaufmann, 1998. – P. 470. – ISBN 1-55860-510-X.

170. Samuel, A. Some studies in machine learning using the game of checkers / A. Samuel // Computers and Thoughts – New York : McGraw-Hill, 1963. – ISBN отсутствует

171. Koza, J.R. Genetic Programming: On the Programming of Computers by Natural Selection / J.R. Koza. – Cambridge : MIT Press, 1992. – P. 819. – ISBN 978-0-2621-1170-6.

172. Poli, R.A Field Guide to Genetic Programming / R. Poli, W.B. Langdon, N.F. McPhee. – Издательство Lulu.com, 2008. – ISBN 978-1-

4092-0073-4. – Текст : электронный. – URL: <http://www.gp-field-guide.org.uk/> (дата обращения: 20.10.2024).

173. Scikit-Learn. Machine Learning in Python / Scikit-Learn : [сайт]. – 2025. – Текст : электронный. – URL: <https://scikit-learn.org/stable/index.html> / (дата обращения: 11.08.2025).

174. Loginom – аналитическая low-code платформа / Loginom : [сайт]. – 2025. – Текст : электронный. – URL: <https://loginom.ru/platform> (дата обращения: 11.08.2025).

175. Cython. C-Extension for Python / Cython : [сайт]. – 2025. – Текст : электронный. – URL: <https://cython.org/> (дата обращения: 12.08.2025).

176. Cython 3.2.0a0 documentation. Getting Started / Cython : [сайт]. – 2025. – Текст : электронный. – URL: <https://cython.readthedocs.io/en/latest/src/quickstart/overview.html> (дата обращения: 12.08.2025).

177. GridSearchCV. Scikit-Learn API Reference/ Scikit-Learn : [сайт]. – 2025. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV (дата обращения: 14.08.2025).

178. Logistic regression/ Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (дата обращения: 05.08.2024).

179. Gaussian Naive Bayes / Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes (дата обращения: 05.08.2024).

180. Random forests and other randomized tree ensembles/ Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: <https://scikit-learn.org/stable/modules/ensemble.html#random-forests-and-other-randomized-tree-ensembles> (дата обращения: 05.08.2024).

181. Сорнетте, Д. Как предсказывать крахи на финансовых рынках: критические события в сложных финансовых системах / Д. Сорнетте ; [перевод с французского Н. Запорович и Т. Чепраковой] ; Библиотека Принстонского университета. – 2-е издание стереотипное – Москва : SmartBook : Изд-во «И-трейд», 2008. – 400 с. : ил., табл. – ISBN 978-5-9791-0042-5.

182. Nagesh, S.C. Predict customers probable purchase / Kaggle : [сайт]. – 2022. – Текст : электронный. – URL:

<https://www.kaggle.com/code/nageshsingh/predict-customers-probable-purchase>
(дата обращения: 07.04.2024).

183. Ordinary Least Squares/ Scikit-learn. Machine Learning in Python : [сайт]. – 2024. – Текст : электронный. – URL: https://scikit-learn.org/stable/modules/linear_model.html#ordinary-least-squares (дата обращения: 17.08.2024).

Работы автора по теме исследования

184. Кораблев, Ю.А. Альтернативное определение спроса / Ю.А. Кораблев // Россия 2030 глазами молодых учёных : материалы III Всероссийской научной конференции (Москва, 26 апреля 2012 г.). – Москва, 2012. – С. 137–140. – ISBN 978-5-91290-192-8.

185. Кораблев, Ю.А. Ёмкостный метод определения функции скорости потребления / Ю.А. Кораблев // Экономика и менеджмент систем управления. – 2015. – № 1.1. Том 15. – С. 140–150. – ISSN 2223-0432.

186. Кораблев, Ю.А. Обоснование ёмкостного метода определения спроса / Ю.А. Кораблев // Экономика, статистика и информатика. Вестник УМО. – 2015. – № 5. – С. 100–104. – ISSN 1994-7844.

187. Кораблев, Ю.А. Ёмкостный метод анализа редких событий в экономике / Ю.А. Кораблев // Экономика: теория и практика. – 2016. – № 4 (44). – С. 59–64. – ISSN 2224-042X.

188. Кораблев, Ю.А. Исследование точности определения скорости потребления продукции с помощью ёмкостного метода при прохождении товара через цепочку распространителей / Ю.А. Кораблев // Современные проблемы моделирования социально-экономических систем : материалы VIII международной научно-практической конференции (1–10 апреля 2016 г.). – Бердянск : Ткачук А.В., 2016. – С. 17–21. – ISBN 978-617-7291-55-7.

189. Кораблев, Ю.А. Ёмкостный метод анализа редких продаж в Excel / Ю.А. Кораблев // Экономика и управление: проблемы, решения. – 2017. – № 6. Том 3 (66). – С. 224–229. – ISSN 2227-3891.

190. Кораблев, Ю.А. Разбор причин и оценка погрешности аномальных картин в ёмкостном методе анализа редких событий / Ю.А. Кораблев // Экономика и управление: проблемы, решения. – 2017. – № 8. Том 6 (68). – С. 8–12. – ISSN 2227-3891.

191. Кораблев, Ю.А. Информационная экономика: этапы развития, методы управления, модели (Пункт 3.6. Влияние позиции в цепочке распространителей на точность ёмкостного метода) : монография / Ю.А. Кораблев ; под редакцией д.э.н., профессора В.С. Пономаренко, д.э.н.,

профессора Т.С. Клебановой. – Харьков : ВШЭМ – ХНЭУ имени С. Кузнеца, 2018. – С. 502–523. – 668 с. – ISBN 978-80-89654-45-1.

192. Кораблев, Ю.А. Исследование точности ёмкостного метода от позиции в цепочке распространителей / Ю.А. Кораблев // Экономика и управление: проблемы, решения. – 2018. – № 5. – Том 7 (77). – С. 106–121. – ISSN 2227-3891.

193. Кораблев, Ю.А. Влияние позиции в цепочке распространителей на точность емкостного метода / Ю.А. Кораблев // Научный журнал «Мягкие измерения и вычисления». – 2018. – № 10. Том 11. – С. 47–62. – ISSN 2618-9976.

194. Кораблев, Ю.А. Дисперсия ёмкостного метода от позиции в цепочке распространителей / Ю.А. Кораблев // Системный анализ в экономике-2018 : сборник трудов V Международной научно-практической конференции-биеннале ; под общей редакцией Г.Б. Клейнера, С.Е. Щепетовой. – Москва : Прометей, 2018. – С. 197–200. – ISBN 978-5-907100-80-0.

195. Кораблев, Ю.А. Анализ и прогнозирование редких событий в экономике / Ю.А. Кораблев // Современная математика и концепции инновационного математического образования : труды VI Международной научно-практической конференции (23 мая 2019). – Москва : Издательский дом МФО, 2019. – С. 123–132. – ISSN 2412-9895.

196. Кораблев, Ю.А. Ёмкостный метод анализа редких событий в торговле различными товарами / Ю.А. Кораблев, П.С. Голованова, Т.А. Кострица // Бизнес. Образование. Право. Вестник Волгоградского института бизнеса. – 2019. – № 3 (48). – С. 121–131. – ISSN 1990-536X.

197. Кораблев, Ю.А. Погрешность ёмкостного метода анализа редких событий, удалённость от конечного потребителя / Ю.А. Кораблев // Известия Кабардино-Балкарского научного центра РАН. – 2019. – № 3 (89). – С. 48–77. – ISSN 1991-6639.

198. Кораблев, Ю.А. Восстановление потребности в заёмных средствах с помощью емкостного метода анализа редких событий / Ю.А. Кораблев // Научный журнал «Мягкие измерения и вычисления». – 2019. – № 6. – Том 19. – С. 22–37. – ISSN 2618-9976.

199. Кораблев, Ю.А. Ёмкостный метод анализа редких событий, оценка погрешности вследствие конкуренции или потери данных / Ю.А. Кораблев // Современная экономика: проблемы и решения. – 2019. – № 10 (118). – С. 18–31. – ISSN 2078-9017.

200. Кораблев, Ю.А. Ёмкостный метод анализа редких событий, допущения при использовании в сфере услуг / Ю.А. Кораблев,

П.С. Голованова, Т.А. Кострица // Современная экономика: проблемы и решения. – 2019. – № 11 (119). – С. 22–32. – ISSN 2078-9017.

201. Кораблев, Ю.А. Ёмкостный метод анализа редких событий в сфере услуг / Ю.А. Кораблев, П.С. Голованова, Т.А. Кострица // Экономическая наука современной России. – 2020. – № 3 (90). – С. 132–142. – ISSN 1609-1442.

202. Кораблев, Ю.А. Метод восстановления функции по интегралам для анализа и прогнозирования редких событий в экономике / Ю.А. Кораблев // Экономика и математические методы. – 2020. – № 3. – Том 56. – С. 113–124. – ISSN 0424-7388.

203. Кораблев, Ю.А. Исследование точности ёмкостного метода анализа редких событий от неопределённости внутри процесса образования событий (Часть 1) / Ю.А. Кораблев // Известия Кабардино-Балкарского научного центра РАН. – 2020. – № 5 (97). – С. 49–67. – ISSN 1991-6639.

204. Кораблев, Ю.А. Программный продукт, реализующий метод восстановления функции по последовательности интегралов, которые наблюдаются с погрешностью, основанный на интегральных сплайнах со штрафной функцией : Свидетельство о государственной регистрации программы для ЭВМ. – № 2020617973, 15.07.2020. – Заявка № 2020617018 от 06.07.2020.

205. Кораблев, Ю.А. Ёмкостный метод анализа редких событий в экономике, интегральный сплайн в Excel / Ю.А. Кораблев // Научный журнал «Мягкие измерения и вычисления». – 2020. – № 10. Том 35. – С. 32–45. – ISSN 2618-9976.

206. Кораблев, Ю.А. Использование ёмкостного метода для анализа социальных событий / Ю.А. Кораблев, П.С. Голованова, Т.А. Кострица // Научный журнал «Мягкие измерения и вычисления». – 2021. – № 6. Том 43. – С. 81–92. – ISSN 2618-9976.

207. Кораблев, Ю.А. Использование ёмкостного метода для анализа исторических событий / Ю.А. Кораблев, П.С. Голованова, Т.А. Кострица // KANT. – 2021. – № 1 (38). – С. 27–32. – ISSN 2222-243X.

208. Кораблев, Ю.А. Кросс-валидация для интегрального сплайна, восстанавливающего скорость потребления по редким событиям / Ю.А. Кораблев // Системный анализ в экономике-2020 : сборник трудов VI Международной научно-практической конференции-биеннале ; под общей редакцией Г.Б. Клейнера, С.Е. Щепетовой. – Москва : Издательский дом «Наука», 2021. – С. 410–413. – ISBN 978-5-6046256-0-6.

209. Korablev, Yu.A. Restoration of function by integrals with cubic integral smoothing spline in R = Восстановление функции по интегралам с помощью кубического интегрального сглаживающего сплайна в R / Yu.A. Korablev //

ACM Transactions on Mathematical Software. – 2022. – № 2. Volume 48. – P. 1–17. – ISSN 0098-3500.

210. Korablev, Yu.A. Restoration of the product consumption rate with integral cubic smoothing spline, study of the best smoothing parameter choice = Восстановление скорости потребления продукции с помощью интегрального кубического сглаживающего сплайна, исследование выбора лучшего параметра сглаживания / Yu.A. Korablev // Acta Applicandae Mathematicae. – 2022. – № 180 (8). – P. 1–30. – ISSN 1572-9036.

211. Кораблев, Ю.А. Об одном алгоритме восстановления функции по разным функционалам для прогнозирования редких событий в экономике / Ю.А. Кораблев // Финансы: теория и практика / Finance: Theory and Practice. – 2022. – № 3. Том 26. – С. 196–225. – ISSN 2587-5671.

212. Кораблев, Ю.А. Определение параметров процесса образования редких событий в экономике для их последующего прогнозирования / Ю.А. Кораблев // Экономика и математические методы. – 2022. – № 2. – Том 58. – С. 80–91. – ISSN 0424-7388.

213. Кораблев, Ю.А. Восстановление параметров процесса образования событий в экономике, заданного алгоритмической моделью / Ю.А. Кораблев // Известия Кабардино-Балкарского научного центра РАН. – 2022. – № 4 (108). – С. 96–114. – ISSN 1991-6639.

214. Кораблев, Ю.А. Алгоритм автоматического перебора произвольных моделей процесса образования редких событий в экономике / Ю.А. Кораблев // Научный журнал «Мягкие измерения и вычисления». – 2022. – № 9. Том 58. – С. 62–88. – ISSN 2618-9976.

215. Кораблев, Ю.А. Ёмкостный метод анализа и прогнозирования редких событий в экономике : монография / Ю.А. Кораблев. – Москва : РУСАЙНС, 2023. – 296 с. – ISBN 978-5-466-04159-0.

216. Кораблев, Ю.А. Ёмкостный метод анализа редких событий в экономике : специальность 5.2.2. Математические, статистические и инструментальные методы в экономике : диссертация на соискание ученой степени доктора экономических наук / Кораблев Юрий Александрович ; Финансовый университет при Правительстве Российской Федерации. – Москва, 2023. – 342 с. – Библиогр.: с. 312–325.

217. Кораблев, Ю.А. Ёмкостный метод анализа редких событий в экономике : автореферат диссертации на соискание ученой степени доктора экономических наук : 5.2.2. Математические, статистические и инструментальные методы в экономике / Кораблев Юрий Александрович ; Финансовый университет при Правительстве Российской Федерации. – Москва, 2023. – Библиография 52 с.

218. Кораблев, Ю. А. Monotone smoothing cubic spline and spline collocation for restoring the product consumption rate and predicting events (FunctionalSmoothingSpline) = Монотонный сглаживающий кубический сплайн и сплайновая коллокация для восстановления скорости потребления продукции и прогнозирования событий / Ю. А. Кораблев. – [Препринт. Версия 1]. – Research Square, 2024. – 20 August. – URL: <https://doi.org/10.21203/rs.3.rs-4742025/v1> (дата обращения: 25.02.2026). – Режим доступа: свободный. – Текст : электронный.

219. Кораблев, Ю.А. Распространённые ошибки использования машинного обучения при прогнозировании событий и новый подход на основе моделей механизмов образования событий / Ю.А. Кораблев, В.А. Судаков // Экономика и математические методы. – 2025. – № 1. Том 61. – С. 25–37. – ISSN 0424-7388.

Приложение А.

(информационное)

Формальное определение механизма образования событий через аксиомы

Колмогорова

Для большей строгости можно дать некоторую математическую трактовку механизмам образования событий. Во-первых, можно сказать, что таким механизмом будет являться неслучайный (детерминированный, явный) процесс \mathbb{F} , состоящий из внутренних переменных X , параметров P , операторов, ветвлений, условий (триггеров), развивающийся со временем t и приводящий к образованию событий. Под детерминированным процессом в математике понимают процесс, который можно описать явными математическими формулами¹⁾. Однако такой детерминированный процесс всё же будет содержать некоторую неопределённость, так как параметры процесса будут иметь некоторый разброс из-за невозможности их абсолютно точного восстановления. Кстати, в математике имеется понятие функции случайных переменных. В нашем случае выходит нечто похожее, только вместо функции над случайными переменными у нас будет детерминированный (явный) процесс, развивающийся со временем, использующий параметры, имеющие случайный характер. Из-за этого получаемые события и интервалы между ними хотя и будут в большей мере объясняться самим механизмом, но всё же будут обладать некоторым разбросом. Поэтому можно дать ещё одно альтернативное определение, сказав, что у нас имеется случайный процесс, где интервалы времени между событиями являются случайными величинами, при условии, что среднее (а лучше мода) этих интервалов определяется по детерминированному (явному) механизму \mathbb{F} . Эти два определения говорят об одном и том же, только с разных сторон. Первое – с начала, описывая рождение событий, второе – с конца, описывая полученную в результате выборку. И в том, и в другом случае интервалы времени между событиями T_i можно выразить двумя способами

$$T_i = \mathbb{F}(P_1 + \varepsilon_{P_{1i}}, P_2 + \varepsilon_{P_{2i}}, \dots, P_n + \varepsilon_{P_{ni}}), \quad (\text{A. 1})$$

$$T_i = \mathbb{F}(P_1, P_2, \dots, P_n) + \varepsilon_i, \quad (\text{A. 2})$$

где \mathbb{F} – детерминированный процесс, P_j – параметры, $\varepsilon_{P_{ji}}$ – погрешность параметра j , ε_i – погрешность интервала времени.

В обоих способах записи интервал времени между событиями T_i будет обладать случайным характером. Однако, если в записи (A.1) можно представить, как ведёт себя погрешность параметров $\varepsilon_{P_{ji}}$ (априори предположив закон распределения), то в записи

¹⁾ Бендат Дж., Пирсол А. Прикладной анализ случайных данных / пер. с англ. – М. : Мир, 1989. – 540 с. – С. 14.

(А.2) заранее (априори) определить, как ведёт себя погрешность интервалов времени ε_i , будет неправильно. Мы будем придерживаться больше первого определения, однако второе определение также допустимо. В обоих случаях можно составить формальное определение, которое базируется на аксиомах Колмогорова.

В случае, если параметры стационарны, пусть:

$X = \{X_1, X_2, \dots, X_{n_x}\}$ – множество внутренних переменных, $X_i \in R$;

$P = \{P_1, P_2, \dots, P_{n_p}\}$ – множество параметров, которые могут иметь вероятностную природу (случайные величины), т.е. (аксиомы Колмогорова):

Ω_i – множество элементов ω , называемых элементарными событиями;

\mathfrak{F}_i – множество подмножеств из Ω_i ;

1) \mathfrak{F}_i – сигма алгебра;

2) каждому множеству A из \mathfrak{F}_i поставлено неотрицательное число $P(A)$;

3) $P(\Omega_i) = 1$;

4) Если A и B не пересекаются, то $P(A + B) = P(A) + P(B)$;

5) Для убывающей последовательности $A_1 \supseteq A_2 \supseteq \dots \supseteq A_n$ из \mathfrak{F}_i , такой что $\bigcap_n A_n = \emptyset$ имеет место равенство $\lim_n P(A_n) = 0$.

Определение случайной величины

$P_i = P_i(\omega)$ – измеримая вещественная функция, определённая на множестве Ω_i .

\mathbb{F} – детерминированный процесс, состоящий из математических операторов и условий (триггеров) над внутренними переменными X и параметрами P , развивающийся со временем t ;

Процедуры детерминированного процесса описываются явными математическими формулами (имеют аналитическое выражение).

\mathfrak{F}_{X_i} – множество подмножеств из R (возможные значения внутренней переменной X_i для наступления события);

\mathfrak{F}_{P_i} – множество подмножеств из Ω_i (возможные значения параметра P_i для наступления события);

\mathfrak{F}_{X_i} и \mathfrak{F}_{P_i} – сигма алгебра.

Элементы множества $\mathfrak{F}_{X,P} = (\mathfrak{F}_{X_1}, \dots, \mathfrak{F}_{X_{n_x}}, \mathfrak{F}_{P_1}, \dots, \mathfrak{F}_{P_{n_p}}, t)$ назовём событием в момент времени t .

$Y_t = Y(X, P, t)$ – назовём значением события, возникшим в момент времени t .

В случае, если параметры нестационарные (динамические), в предыдущем определении заменяем

$P = \{P_1, P_2, \dots, P_{n_p}\}$ – множество параметров, которые могут являться случайными процессами, т.е.

T – некоторое подмножество на числовой прямой;

$P_i = (\xi_t)_{t \in T}$ – совокупность случайных чисел на числовой прямой называется случайным процессом.