Федеральное государственное бюджетное учреждение науки
Вычислительный центр им. А. А. Дородницына
Российской академии наук

На правах рукописи

**СОЛОГУБ РОМАН АРКАДЬЕВИЧ**

# АЛГОРИТМЫ ИНДУКТИВНОГО ПОРОЖДЕНИЯ И ТРАНСФОРМАЦИИ МОДЕЛЕЙ В ЗАДАЧАХ НЕЛИНЕЙНОЙ РЕГРЕССИИ

05.13.17 — теоретические основы информатики

**ДИССЕРТАЦИЯ**

на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
к.ф.-м.н., доц. В. В. Стрижов

МОСКВА – 2014

## Оглавление

## Введение

Данная работа направлена на решение проблемы автоматического построения и верификации количественных математических моделей. Модели предназначены для описания результатов измерений и прогнозирования экспериментов, составляющих неотъемлемую часть естественнонаучных исследований [2].

В работе исследуется фундаментальная проблема автоматического порождения моделей для решения задач анализа данных. Порождаемые модели предназначены для аппроксимации, анализа и прогнозирования результатов измерений. При порождении учитываются требования, предъявляемые экспертами-специалистами в предметной области к порождаемым моделям. Это дает возможность получения экспертно-интерпретируемых моделей, адекватно описывающих результат измерения.

Для создания адекватной модели измеряемых данных используются экспертно-заданные порождающие функции и набор правил порождения. Модель задается в виде суперпозиции порождающих функций. Правила порождения определяют допустимость суперпозиции и исключают порождение изоморфных моделей.

В работе предлагается развить существующие методы автоматического порождения моделей. В частности, при порождении моделей предлагается учитывать экспертные требования к виду моделей, ранжируя модели в соответствии с экспертными предпочтениями. Исследуются методы и алгоритмы порождения моделей, их свойства, сложность и устойчивость. Анализируется проблема возникновения различных топологически, но при этом равных функционально моделей. Предлагаются новые методы поиска изоморфных

суперпозиций, основанные на поиске изоморфных подграфов и подстановке подграфов по правилам.

Использование нелинейной регрессии для решения прикладных задач описывается в работах Дж. Себера [48, 49]. В них описывается построение и оценка параметров нелинейных моделей. Для оценки моделей используется алгоритм Левенберга-Марквадта [33]. Критерием качества при этом, как и в случае обычной линейной регрессии, остается среднеквадратичная ошибка. Данный критерий качества будет основным и в данной работе.

В прикладных задачах моделирования зачастую оказывается, что сведения о структуре модели, в том числе экспертные суждения о виде искомых зависимостей являются недостаточными для применения методов, способных обеспечить достаточное качество работы. Недостаток числа независимых переменных для построения математических моделей делает чрезвычайно перспективным для решения такого рода задач применение методов порождения признаков и моделей.

Идея метода порождения признаков заключается в создании дополнительных независимых переменных, являющихся образами исходных переменных относительно последовательно примененных наборов отображений. Такие отображения в рамках работы будут называться порождающими функциями.

Ранее работы, выполненные в рамках данного подхода, являлись в основном прикладными. Для различных задач экономики и промышленности на основе экспертного понимания проблемы выбирались примитивы и порождались наборы новых признаков для построения модели. При этом исследователями не ставился вопрос существования набора, полноты или корректности существующего

алгоритма.

В данной работе развивается теоретическое обоснование корректности и эффективности использования методов порождения суперпозиций для решения прикладных задач. Рассматриваются алгоритмы порождения суперпозиций, их сходимость, различные методы оптимизации структуры моделей. В работах А.Г. Ивахненко [8, 36] индуктивное порождение моделей строится с помощью метода группового учета аргументов. В линейной модели предлагается порождать новые признаки с помощью операции произведения. С помощью полиномов Колмогорова-Габора [3] алгоритм целенаправленно порождает и перебирает модели-претенденты различной сложности согласно ряду критериев. В результате находится модель оптимальной структуры в виде одного уравнения или системы уравнений [36].

Для индуктивного порождения моделей в работах Дж. Козы [32, 31], связанных с генетическим программированием [10, 38], осуществляется переход от строковой записи моделей к префиксной записи, таким образом вводится построение модели в виде графа-дерева. Данные деревья генерируются случайным образом, начиная от вершины. Для поиска оптимальной модели Коза организует процедуру, схожую с эволюционным процессом. Последовательно выполняются следующие шаги.

1. Случайным образом выбираются пары исходных моделей. На пару накладывается операция генетического скрещивания, таким образом порождается новая пара производных моделей.
2. Случайным образом выбираются исходные модели и элементы в них. С этими элементами проводится операция генетической мутации, что приводит к порождению производных моделей.

3. В соответствии с критерием качества модели сортируются, лучшие модели оставляются для дальнейших итераций.

В работах Дж. Козы процесс повторяется заданное количество раз, или до достижения достаточного качества модели. Данный теоретический метод позволил успешно решить задачу определения оптимальной формы проводной антенны [15].

Работы И. Зелинки [30], продолжающие работы Дж. Козы, связаны с аналитическим программированием — дальнейшим алгебраическим развитием методов генетического программирования. Автор использует строковое представление и цепочки логических предикатов в качестве элементов модели. Также в процессе построения моделей отсекаются циклические, а также имеющие комплексные или бесконечные значения.

В работах В.В. Стрижова [4, 6, 7] происходит отход от полностью случайного генетического поиска. Элементам модели в соответствии с теорией Байесовского вывода ставятся в соответствие гиперпараметры [35], и в соотвествии с их значениями определяется вероятность модификации того или иного элемента набора.

Построение прогностический модели в виде суперпозиции заданных функций, предложенное в работе Г.И. Рудого [1] позволяет получать интерпретируемые модели, а предложенный метод штрафования суперпозиций за сложность порождает менее точные, но более простые суперпозиции, что является позволяет получать экспертные интерпретации. Метод преобразования и упрощения суперпозиций по правилам, рассмотренный в работе [1], позволяет разделить построенные суперпозиции на классы эквивалентности и выбрать из каждого класса наиболее простую (то есть, имеющую наименьшее число структурных элементов) суперпозицию, что также позволя-

ет обосновать возможность экспертной интерпретации. Методы построения комбинаций прогностических моделей описаны в работых [30, 31, 53]

При рассмотрении генетического процедуры порождения деревьев важной задачей является оценка расстояния между моделями для понимания, насколько различные модели наивысшего качества отличаются друг от друга. Не существует общепринятой метрики, используемой в таких случаях. В рамках работы будет использоваться метрика, основанная на идеях, рассмотренных в работах Макарова [5].

Для недопущения эффекта переобученности структурная сложность моделей должна ограничиваться. Сложность моделей в рамках данной работы оценивается по методу, предложенному К. Владиславлевой [52, 53]. Для модели, представленной в виде дерева, её сложность равна количеству элементов во всех поддеревьях данного дерева.

В работах Т. Соула [50] рассматриваются методы, модифицирующие процедуру генетического отбора с целью решения проблем, связанных с чрезмерно быстрым ростом сложности моделей. Для этого модифицируется шаг генетического скрещивания структур моделей таким образом, что в случае, если производная модель не превосходит по качеству исходные модели, то она выбрасывается на шаге отбора моделей.

В работе П. Нордина [41] предлагается изменение вероятности генетического скрещивания в зависимости от качества моделей. Таким образом повышается вероятность появления в большом количестве моделей поддеревьев, которые есть в моделях наилучшего качества.

В середине 70-х годов Дж. Холланд [26] сформулировал и дока-

зал теорему схем, связанную с генетическими алгоритмами, в которых гены представляются в виде строк. Схемой называется подмножество множества всех возможных подстрок, возможных в данном наборе строк, заданное в виде подстроки с фиксированными значениями некоторых битов. Остальные биты могут принимать любые значения, образуя примеры схемы. Так, примерами схемы 00**1* являются подстроки 000010, 000011, 000110, 000111, 001010, 001011, 001110 и 001111. Количество фиксированных символов называется порядком схемы, а расстояние между крайними фиксированными позициями (т.е. разность их номеров) — её определяющей длиной. Порядок вышеприведённой схемы равен 3, а определяющая длина $5-1=4$. Функция пригодности схемы — это среднее значение функции качества всех строк, её содержащих. Теорема схем показывает происходящее при смене поколений экспоненциальное распространение схем высокой функцией пригодности с малыми порядком и определяющей длиной.

В работах Поли [43, 44] и Лангдона [42] теорема схем обобщается для алгоритмов, связанных с построением суперпозиций в виде деревьев. Рассматриваются различные операции замены поддеревьев, для них определяется вероятность сохранения поддерева заданной структуры с определенными порождающими функциями в вершинах.

Для упрощения структуры моделей используются методы теории трансформации графов, предложенные в работах Х. Эрига [21]. Для трансформации деревьев выделяются некоторые элементарные графы-шаблоны, для которых строятся оболочки изоморфных им графов более сложной структуры. Для упрощения модели производится рекурсивный поиск подграфов, изоморфных графам-

шаблонам, с их заменой на более простые подграфы.

Задача упрощения моделей, представленных в виде графов, рассматривается в работах Н. Мори [39]. Автор рассматривает два различных метода упрощения моделей. В первом анализируется структура моделей и выделяются элементы-подграфы, которые подходят под шаблоны упрощения (например, двойное отрицание). Альтернативным методом является вычисление значений элемента модели на исходной выборке. Если значения функции совпадают со значениями более простого шаблона, осуществляется замена элемента модели шаблоном.

Среди методов упрощения моделей также следует отметить метод оптимального прореживания, используемый в работах Я. Ле Куна [16] и Б. Хассиби [24]. Данный метод предполагал удаление из нейронных сетей избыточных вершин, влияние которых на качество модели было отрицательным. Подобный подход также распространяется на модели в генетическом программировании.

Переход от порождения моделей заданной структуры к моделям общего вида, являющимися суперпозициями заданных функций, также реализуется в работах, посвященных методологии Deep learning [40, 11]. Данный метод используется в задачах распознавания изображений, его идея предполагает, что строится многослойная модель, каждый уровень которой используется для распознавания различных сущностей. Например, на определенном уровне устанавливается наличие человека, другие уровни модели устанавливают параметры, которыми может быть описано лицо человека или его одежда.

В основе метода лежит идея создания многослойных нейронных сетей, при этом вместо функций активации нейронов скрытых слоев

могут использоваться различные нелинейные функции.

Каждый слой сети глубокого обучения представляет собой ограниченную машину Больцмана [47] –– вид стохастической рекуррентной нейронной сети, изобретенной Дж. Хинтоном и Т. Сейновски [25]. Ограниченная машина Больцамана является частным случаем машины Больцмана с тем ограничением, что нейроны сети должны составлять двудольный граф — каждый нейрон скрытого слоя должен быть с каждым входом нейронной сети (в отличие от обычной Больцмановской машины, где разрешены связи между нейронами скрытого слоя, что делает эти сети рекуррентными). Элементы данной сети строятся в соответствии с условными вероятностями, рассчитываемыми на основе исходных данных. При этом разрабатывается локально оптимальный алгоритм настройки данной сети по слоям.

**Целью работы** является исследование проблемы построения нелинейных регрессионных моделей как суперпозиций заданных параметрических функций.

**Основные положения, выносимые на защиту:**

1. Разработан алгоритм направленного порождения моделей. Разработаны новые алгоритмы вычисления структурной сложности порождаемых суперпозиций и алгоритмы вычисления расстояния между порождаемыми суперпозициями.
2. Разработан метод последовательного направленного порождения суперпозиций, исследованы свойства порождаемых суперпозиций.
3. Введено понятие изоморфных суперпозиций, разработан метод их обнаружения. Разработан алгоритм поиска изоморф-

ных подграфов, соответствующих порожденным суперпозициям.

4. Разработан новый метод порождения экспертно-интерпретируемых моделей. Создана базовая библиотека правил порождения экспертно-интерпретируемых моделей.

**Научная новизна:**

1. Предложен алгоритм индуктивного порождения регрессионных моделей, являющихся суперпозициями экспертно-заданных параметрических функций.
2. Предложен метод трансформации суперпозиций, представленных в виде категории на множестве направленных ациклических графов без самопересечений, соответствующих суперпозициям.
3. Предложен алгоритм индуктиного порождения регрессионных моделей, являющихся суперпозициями экспертно-заданных параметрических функций.

**Практическая значимость**

Предлагаемые в работе методы порождения моделей предназначены непосредственно для применения на практике. Алгоритмы порождения суперпозиций могут использоваться для решения задач обучения по прецедентам в различных прикладных областях, включая техническую диагностику, социологию, экономические задачи, задачи финансового рынка.

Финансовый рынок в целом характеризуется большим количеством ложных регрессий между ценами различных инструментов. В связи с этим, в качестве входных переменных, использование которых в модели ценообразования инструмента финансового рынка

оправдано, может выступать лишь небольшая группа основных факторов рынка [9]. В то же время цены и волатильности различных производных инструментов финансового рынка имеют ярко выраженную существенно-нелинейную зависимость от независимых переменных. В таких условиях применение алгоритмов порождения суперпозиций является оптимальным инструментом решения прикладных задач, стоящих перед экспертами финансового рынка.

**Достоверность** результатов подтверждена математическими доказательствами, экспериментальной проверкой полученных алгоритмов на реальных задачах регрессии; публикациями результатов исследования в рецензируемых научных изданиях, в том числе рекомендованных ВАК РФ. Результаты работы докладывались, обсуждались и получили одобрение специалистов на следующих научных конференциях и семинарах:

- Интеллектуализация Обработки Информации ИОИ-08 (Украина, Алушта, 2008);
- SIAM Conference on Financial Mathematics & Engineering 2008 (USA, New Bruinswick, 2008);
- Математика. Компьютер. Образование. 2009 (Россия, Пущино, 2009);
- EURO 2009 conference (Germany, Bonn, 2009);
- Математические Методы Распознавания Образов ММРО-14 (Россия, Суздаль, 2009);
- EURO 2010 conference (Portugal, Lisbon, 2010);
- EURO 2012 conference (Lithuania, Vilnus, 2012).

**Публикации.** Основные результаты по теме диссертации изложены в 6 печатных изданиях, 3 из которых изданы в журналах, рекомендованных ВАК, 3 — в тезисах докладов.

1. Сологуб Р.А. Алгоритмы порождения нелинейных регрессионных моделей // Информационные технологии, 2013. No 5. С. 8 – 12
2. Сологуб Р.А. Порождение регрессионных моделей поверхности волатильности биржевых опционов // Информационные технологии, 2012. No 8. С. 47 – 52
3. Стрижов В.В., Сологуб Р.А. Индуктивное порождение поверхности волатильности опционных торгов // Вычислительные технологии, 2009. No 5. С. 102—113.
4. Sologub R., Strijov V. The inductive generation of the volatility smile models // SIAM Financial Modeling 08 conference proceedings. P. 21.
5. Sologub R. Inductive generation of foreign exchange forecast models // 23rd European Conference On Operational Research proceedings. P. 162.
6. Sologub R. Model generation for equity-futures spread forecasting // 24th European Conference On Operational Research proceedings. P. 168.

# Глава 1
# Постановка задачи

Пусть задана выборка $\mathfrak{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^m$. В зависимости от задачи выборка разбивается на обучающую и тестовую. Обозначим $\ell, C$ — множества индексов из $\{1, \ldots, W\} = \mathcal{I}$. Эти множества удовлетворяют условиям разбиения $\ell \cup C = \mathcal{I}, \ell \cap C = \emptyset$. Матрица $X_\ell$ состоит из тех векторов-строк $\mathbf{x}_n$, для которых индекс $n \in \ell$. Вектор $\mathbf{y}_\ell$ состоит из тех элементов $y_n$, для которых индекс $n \in \ell$. Разбиение выборки представляется в виде

$$X_W = \left(\frac{X_\ell}{X_C}\right), \quad \mathbf{y}_W = \left(\frac{\mathbf{y}_\ell}{\mathbf{y}_C}\right), \quad \text{где}$$

$$\mathbf{y}_W \in \mathbb{R}^{N\times 1}, X_W \in \mathbb{R}^{N\times m}, |\ell| + |C| = |W|.$$

Требуется построить функцию регрессии $\varphi(\mathbf{x}, \mathbf{w}) \mapsto \mathbf{y}$. Из множества функций $F$ требуется выбрать модель $f$ — отображение из декартова произведения множества свободных переменных $\mathbf{x} \in \mathbb{R}^n$ и множества параметров $\mathbf{w} \in \mathbb{R}^m$ в $\mathbb{R}^1$. Сужение модели есть функция регрессии $\varphi$ с заданными значениями $\mathbf{w} = \mathbf{w}_0$. Требуется оценить набор параметров $\mathbf{w}_0$, доставляющие минимум внешнему критерию качества [12] модели — квадратичной ошибке

$$S(\mathbf{w}|\mathfrak{D}, f) = ||f(\mathbf{x}, \mathbf{w}) - y||.$$

Выражение $S(\mathbf{w}|\mathfrak{D}, f)$ означает значение функции ошибки $S$, которое зависит от набора параметров $\mathbf{w}$ при заданной выборке $\mathfrak{D}$ и модели $f$. Такая модель называется оптимальной при условии, что её сложность $C(f)$ не превышает заданной. Сложность определяется как количество элементов во всех поддеревьях, которые можно выделить из дерева, представляющего модель.

Для выбора модели используется внутренний критерий качества $S$ [14], связанный с оценкой уровня правдоподобия модели в предположениях, что параметры модели и независимые величины принадлежат каким-либо заранее известным распределениям. Рассмотрим нормальное распределение зависимой переменной $y$ в качестве гипотезы порождения данных при восстановлении линейной или существенно-нелинейной регрессии:

$$\mathsf{E}(y|\mathbf{x}) = f(\mathbf{w}, \mathbf{x}).$$

Для нахождения наиболее правдоподобных параметров модели используется метод наибольшего правдоподобия. Пусть многомерная случайная величина $\mathbf{y} \sim \mathcal{N}(f, \mathbf{B})$ имеет нормальное распределение

$$p(y) = \frac{1}{(2\pi)^{\frac{2}{m}} \det^{\frac{1}{2}} \mathbf{B}^{-1}} \exp(-\frac{1}{2}(\mathbf{y} - f)^T \mathbf{B}(\mathbf{y} - f)) \qquad (1.1)$$

Рассмотрим частный случай гипотезы порождения данных: элементы вектора $\mathbf{y}$ не коррелируют и имеют одинаковую дисперсию, то есть обратная ковариационная матрица $\mathbf{B} = \beta \mathbf{I}_m$. Диагональные элементы этой матрицы $\beta_i$ обратны значениям дисперсии элементов случайной величины $\mathbf{y}$:

$$\sigma_i^2 = \frac{1}{\beta_i}, \quad \beta_i > 0, \ i \in I.$$

Так как правая часть выражения 1.1 зависит от вида регрессионной модели $f$, вектора параметров $\mathbf{w}$, независимой переменной $\mathbf{x}$ и от дисперсий $\beta$, перепишем данное уравнение в сокращенном виде

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, \beta, f) = \frac{\exp(-\mathsf{E}_{\mathfrak{D}})}{Z_{\mathfrak{D}}(\beta)},$$

где $Z_{\mathfrak{D}}$ — нормирующий коэффициент для плотности нормального распределения.

Функция ошибки, соответствующая матожиданию регрессионной модели при данной гипотезе, определена как

$$\mathsf{E}_{\mathfrak{D}} = \frac{1}{2}\left(\mathbf{y} - f\right)^T \mathbf{B}(\mathbf{y} - f)\big) = \frac{1}{2}\beta \sum_{i \in \mathcal{I}} (y_i - f(\mathbf{w}, x_i))^2 = \frac{1}{2}\beta \|\mathbf{y} - f\|_2.$$

Рассмотрим вектор параметров $\mathbf{w}$ модели $f$ — многомерную случайную величину. Согласно принятой гипотезе распределения 1.1 зависимой переменной и теореме о функциях связи распределений, распределение параметров $\mathbf{w} \sim \mathcal{N}(\mathbf{w}_{\mathrm{ML}}, \mathbf{A}^{-1})$ является нормальным с матожиданием $\mathbf{w}_{\mathrm{ML}}$, ковариационной матрицей $\mathbf{A}^{-1}$ и имеет вид

$$p(\mathbf{w}|\mathbf{A}, f) = \frac{\exp(-E_W)}{Z_W(\mathbf{A})}.$$

Данное выражение справедливо для линейных моделей. Для существенно-нелинейных моделей предполагается, что данное выражение будет справедливо в окрестности $\delta\mathbf{w}$ некоторой точки $\mathbf{w}_0$.

Нормирующий коэффициент $Z_w(\mathbf{A})$ равен

$$Z_W(\mathbf{A}) = (2\pi)^{\frac{n}{2}} \det{}^{\frac{1}{2}}(\mathbf{A}^{-1}), \tag{1.2}$$

где $n$ — число параметров модели $f$. Функция-штраф за большое значение параметров модели для принятого распределения определена как

$$E_W = \frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T \mathbf{A}(\mathbf{w} - \mathbf{w}_0). \tag{1.3}$$

Рассмотрим частный случай: дисперсии элементов $w_j$ вектора параметров $\mathbf{w}$ равны, обратная ковариационная матрица имеет вид $\mathbf{A} = \alpha\mathbf{I}_n$. В этом случае выражения (1.2) и (1.3) будут иметь вид

$$Z_W(\mathbf{A}) = (\frac{2\pi}{\alpha})^{\frac{n}{2}} \text{ и } E_W = \frac{1}{2}\alpha\|\hat{\mathbf{w}} - \mathbf{w}\|^2.$$

Отрицательный логарифм правдоподобия модели при нормальных распределениях векторов $\mathbf{y}$ и $\mathbf{w}$ :

$$S(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}_{ML})^T \mathbf{A}(\mathbf{w} - \mathbf{w}_{ML}) + \frac{1}{2}(\mathbf{y} - f(\mathbf{w}, \mathbf{x})^T \mathbf{B}(\mathbf{y} - f(\mathbf{w}, \mathbf{x}),$$

где $\mathbf{w}_{ML}$ — наиболее правдоподобные параметры. Для выбранных выше ограничений значение критерия $S$ записывается как

$$S(\mathbf{w}) = \frac{1}{2}\beta\|\mathbf{y} - f\|^2 + \frac{1}{2}\alpha\|\hat{\mathbf{w}} - \mathbf{w}\|^2.$$

Задано множество $G$ порождающих функций $g(\mathbf{w}, \mathbf{x})$. Для каждого элемента данного множества $g_i$ определены области аргументов $\mathbf{w} \in \mathbb{R}^m, \mathbf{x} \in \mathbb{R}^n$ и значений, при этом область значений принадлежит $\mathbb{R}^1$. В множество порождающих функций обязательно входит не имеющая аргументов функция $\mathrm{id}(\mathbf{x})$, значение которой тождественно значению свободной переменной, а также функция константы const.

Искомую модель $f$ мы будем искать среди множества суперпозиций функций $g \in G$. При этом накладываются ограничения на структуру суперпозиции:

**Определение 1.** Допустимой называется суперпозиция, удовлетворяющая следующим требованиям:

1. Элементами суперпозици $f$ могут являться только порождающие функции $g_j$ и свободные перменные $\mathbf{x}$.
2. Количество аргументов элемента суперпозиции равно арности соответствующей ему функции $g_j$.
3. Порядок аргументов элемента суперпозиции соотвествует порядку аргументов соответствующей функции $g_j$.
4. Для элемента $s_i$, аргументом которого является элемент $s_j$, область определения соответствующей порождающей функции

$g_i$ содержит область значений порождающей функции аргумента $g_j$: $\mathrm{dom}(g_i) \supseteq \mathrm{cod}(g_j)$;

Порождается множество моделей $f \in F$ — допустимых суперпозиций, состоящих из функций $g_i \in G$. Требуется выбрать модель, доставляющую минимум $S(f|\mathbf{w}^*_{\mathrm{ML}}, \mathfrak{D})$ при условии, накладываемом на сложность $C(f) < C^*$. Различные методы определения сложности модели будут рассмотрены в главе 3.

Следует заметить, что выборка вместе с суперпозициями составляют категорию $\mathfrak{F}$, т.к. для данной конструкции выполняются все аксиомы теории категорий:

1. $\mathfrak{F}$-объектами данной категории являются множества независимых перменных $x$ и зависимых переменных $y$.
2. $\mathfrak{F}$-стрелками в данной категории являются суперпозиции $f_i$.
3. Функции $\mathrm{dom}(f)$ и $\mathrm{cod}(f)$ для суперпозиции $f$ определяются естественным образом как область определения и область значений соответствующей суперпозиции.
4. Если для пары суперпозиций $\langle f_1, f_2 \rangle$ выполняется условие $\mathrm{cod}(f_1) = \mathrm{dom}(f_2)$, то суперпозиция $f_2$ имеет область определения $dom(f_2) \in \mathbb{R}^1$. Суперпозиция, в которой вместо независимых переменных из $f_2$ будет использоваться суперпозиция $f_1$, будет допустимой, т.е. композиция существует и входит в множество $\mathfrak{F}$-стрелок. Ассоциативность следует из того факта, что замена в суперпозиции одного аргумента на другой является ассоциативной операцией. Вообще все множество $\mathfrak{F}$-стрелок состоит из элементов $G$ и их композиций.
5. Наличие единицы обеспечивается обязательным существованием в $G$ функции $\mathrm{id}(\mathbf{x})$. Для этой функции выполняется закон тождества по определению.

## 1.1. Описание структуры модели

Условимся считать, что каждой суперпозиции $f$ сопоставлено дерево $\Gamma_f$, эквивалентное этой суперпозиции и строящееся следующим образом:

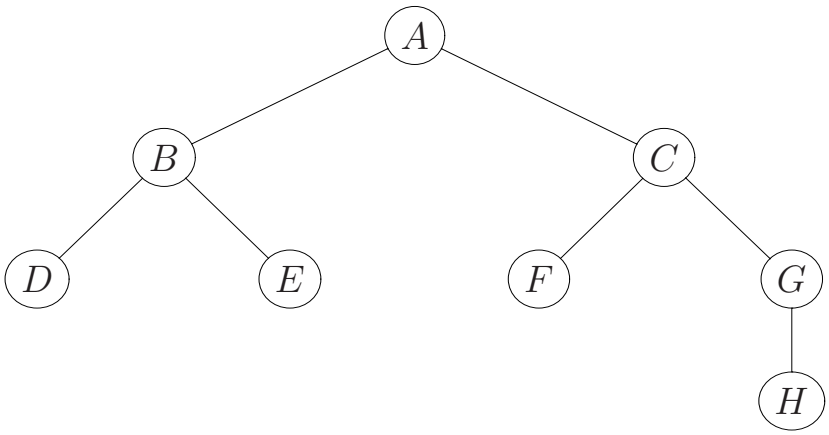
- В вершинах $v_i$ дерева $\Gamma_f$ находятся соответствующие порождающие функции $g_j$.
- Число дочерних вершин у некоторой вершины $v_i$ равно арности соответствующей ей функции $g_j$.
- Порядок дочерних вершин вершины $v_i$ соотвествует порядку аргументов соответствующей функции $g_j$.
- Листьями дерева $\Gamma_f$ являются свободные переменные $x_i$ либо числовые параметры $w_i$.

Таким образом, вычисление значения выражения $f$ в некоторой точке с данным вектором параметров $\mathbf{w} = \{w_1, w_2, \ldots, w_k\}$ эквивалентно подстановке соответствующих значений свободных переменных $x_i$ и параметров $w_i$ в дерево $\Gamma_f$, где $x_i$ — элементы вектора свободных переменных $\mathbf{x}$.

Заметим важное свойство таких деревьев: каждое поддерево $\Gamma'_f$ дерева $\Gamma_f$, корнем которого является вершина $v_i$, также соответствует некоторой суперпозиции, являющейся составляющей исходной суперпозиции $f$.

Актуальной задачей, встающей при использовании сопоставления деревьев моделям, является способ представления деревьев, за-

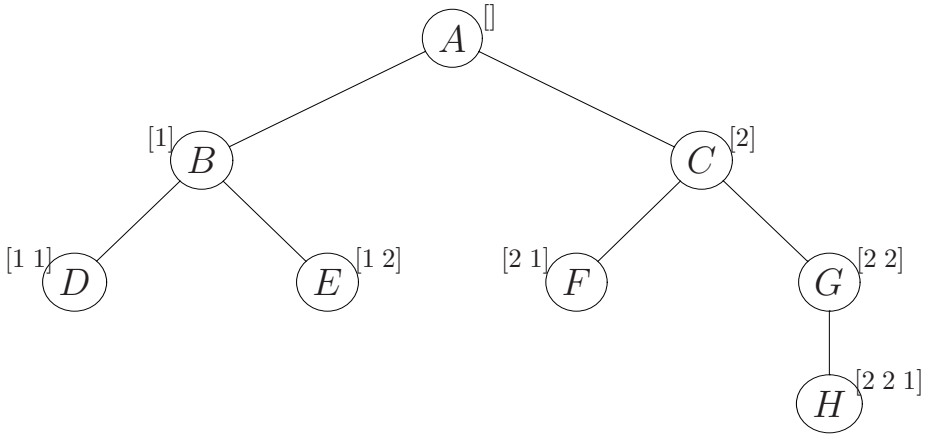
дающих суперпозиции, на плоскости:




(1.4)

Рассмотрим дерево $\Gamma_0$, изображенное на диаграмме 1.4. Простейший метод записи деревьев — перечисление вершин при обходе дерева «в глубину», при этом потомки вершины записываются внутри скобок, подобно аргументам функции. В такой нотации дерево $\Gamma_0$ будет соответствовать записи $(A(B(CD)E(FG(H))))$.

Одним из методов введения координат для вершин является метод пути [18], в котором система координат имеет изменяемую размерность. Разметка дерева в этой системе координат изображена на следующей диаграмме:

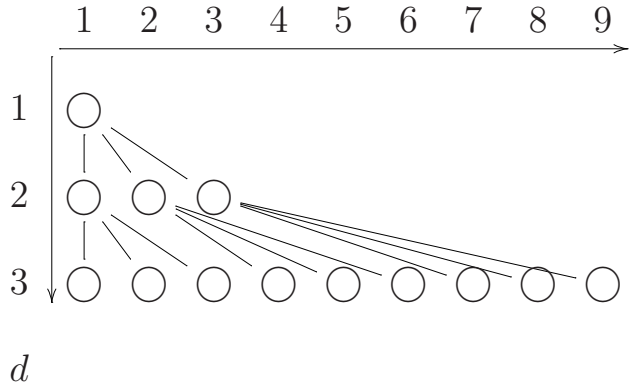



(1.5)

К примеру, вершине $H$ дерева $\Gamma_0$ соответствует список координат [221], что значит, что эта вершина является первым потомком второго потомка второго потомка корня дерева. Подобная система записи плохо подходит для работы с вершинами дерева в контексте дан-

ной работы, потому что обращение к определенной вершине дерева оказывается затрудененным — размер координатной строки оказывается порядка $\ln(|\Gamma_0|)$.

Более подходящей альтернативой является введение координатной системы, в которой вершины дерева были бы организованы в слои увеличивающейся глубины (в порядке, в котором обычно деревья изображаются). Вершины упорядочены слева направо, каждой вершине в слое присваивается номер. Тогда номер слоя $d$ и индекс вершины в слое $i$ определяют прямоугольную систему координат. Например, вершина $G$ дерева $\Gamma_0$ в такой системе координат будет иметь координаты (3,4), т.к. это третья вершина во втором слое дерева. Проблемой, возникающей при использовании данной системы координат, является невозможность определения родительской вершины по координатам потомка. К примеру, вершина $H$ дерева $\Gamma_0$ имеет координаты (4,1) в независимости от того, какая из вершин третьего слоя является её родительской вершиной:




(1.6)

Предлагается построить схожую систему координат, для которой данай проблема будет решена. Заранее задается максимальная арность вершин $a_{\max}$. Максимальное дерево, которое можно построить при таком условии, будет иметь 1 вершину на первом слое, $a_{\max}$ на втором, $a_{\max}^2$ на третьем и так далее. Пример координатной сетки

для деревьев с $a_{\max} = 3$ представлен на диаграмме (1.6). В данной системе координат вершина $G$ имела бы координаты $(d, i) = (2, 4)$, а вершина $H$ — $(3, 12)$. По координатам вершины однозначно определяется родительская вершина, и можно проследить всю цепочку вершин до корня по координате вершины.

Следует заметить, что данная плоская система координат может быть сведена к линейной. Формула перехода при этом строится естественным образом — вершины считаются в порядке слева направо сверху вниз. Тогда вершине с координатами $(d, i)$ будет соответствовать число $\sum_{j=0}^{d-1} a_{\max}^j + i$. В таком случае для немаксимальных деревьев не каждому индексу будет соответствовать вершина, однако в некоторых случаях такая нотация тоже будет использоваться.

## 1.2. Расстояние между моделями

При построении моделей в виде суперпозиций заданных функций оказывается, что небольшие изменения в структуре модели могут значительно изменить занчения функций. В дальнейшем при изменении моделей будет модифицироваться именно структура, поэтому помимо расстояния как нормы разности функций следует каким то образом определить структурное расстояние, позволяющее оценить степень схожести моделей и количество модификаций, необходимых для превращения одной модели в другую. В данной работе используется частный случай метрики, введенной Л.А. Макаровым [5].

Пусть дан граф-дерево $\Gamma(V, X)$, состоящего из множества вершин $V = \{v_i\}$ и множества ребер $X = \{x_{ij}\}$. Каждой вершине соответствует единственная вершина-родитель, т.е. существует инъективная функция $V \to X$, ставящая в соответствия ребрам графа те

вершины-потомки, им соответствующие. Таким образом, для обозначения графа оказывается достаточно множества его вершин.

**Определение 2.** Дерево $\Gamma'(V')$ называется поддеревом дерева $\Gamma(V)$, если его множество вершин $V'$ является подмножеством множества $V$.

**Определение 3.** Два дерева $\Gamma_1$ и $\Gamma_2$ называются изоморфными, если между их множествами вершин существует взаимно однозначное отображение, сохраняющее метки вершин.

**Определение 4.** Дерево $\Gamma_0$ называется общим поддеревом деревьев $\Gamma_1$ и $\Gamma_2$, если в них существуют поддеревья $\Gamma'_1$ и $\Gamma'_2$, изоморфные дереву $\Gamma_0$.

**Определение 5.** Общее поддерево двух деревьев называется наибольшим, если в нем содержится наибольшее число вершин среди других общих поддеревьев. Наибольшее поддерево деревьев $\Gamma_i(V_i)$ и $\Gamma_j(V_j)$ обозначается как $\Gamma_{ij}(V_{ij})$. Символом $p$ будет обозначаться количество элементов в множестве $V$: $|V_i| = p_i$, $|V_j| = p_j$, $|V_{ij}| = p_{ij}$.

Рассмотрим абсолютную функцию расстояния $r$ между $\Gamma_i$ и $\Gamma_j$, зависящую от их размеров и наибольшего общего подграфа,

$$r_{ij} = p_i + p_j - 2p_{ij},$$

и нормированное расстояние $R$ между этими графами:

$$R_{ij} = r_{ij}/(p_i + p_j).$$

Для практического использования данных функций расстояний необходимо, чтобы эти функции удовлетворяли аксимомам метрики.

**Определение 6.** Функция $r$, определенная в пространстве $\Gamma$ и принимающая значения в множестве вещественных чисел называется метрикой, если для точек $\Gamma_1, \Gamma_2, \Gamma_3 \in \Gamma$

1) $d(\Gamma_1, \Gamma_2) = 0 \Leftrightarrow \Gamma_1 = \Gamma_2$ (в данном случае равенство соответствует изоморфности деревьев
2) $d(\Gamma_1, \Gamma_2) = d(\Gamma_2, \Gamma_1)$
3) $d(\Gamma_1, \Gamma_3) \geq d(\Gamma_1,\ \Gamma_2) + d(\Gamma_2,\ \Gamma_3)$

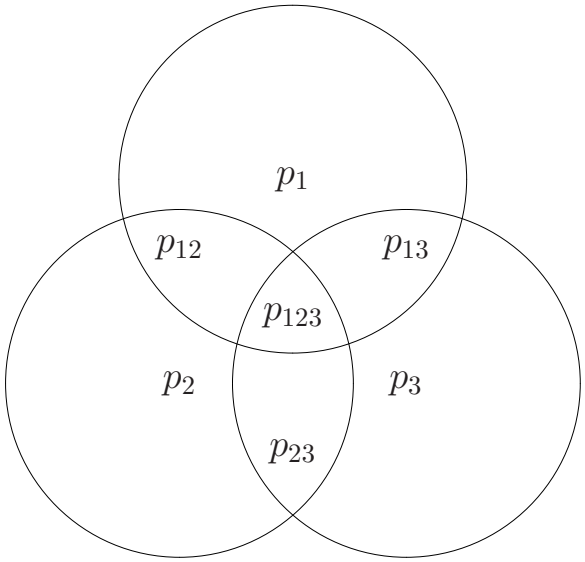
**Теорема 1.** Условия метрики выполняются для функции $r_{ij}$.

Свойства 1 и 2 доказываются следующим образом: для равных деревьев есть их общее поддерево, равное им же, и так как у него такое же число вершин, расстояние будет нулевым. Также и обратное: если расстояние равно нулю, значит мощность общего поддерева равна мощности обоих деревьев, и значит деревья изоморфны.

Докажем третье условие. Пусть даны графы $\Gamma_1$, $\Gamma_2$ и $\Gamma_3$. Необходимо доказать неравенство треугольника для попарных расстояний между ними:

$$\begin{aligned} r_{12} + r_{23} &\geq r_{13}, \\ p_1 + p_2 - 2p_{12} + p_2 + p_3 - 2p_{23} &\geq p_1 + p_3 - 2p_{13}, \\ p_2 - p_{12} - p_{23} + p_{13} &\geq 0. \end{aligned} \tag{1.7}$$

Рассмотрим последнее неравенство. Если графы и их общие поддеревья рассмотреть в виде диаграммы Эйлера (см. рис. 1.8), легко заметить что перекрывающаяся часть поддеревьев $p_{12}$ и $p_{23}$ не может быть больше, чем $p_{13}$. Таким образом, в левой части неравенства обязательно находится неотрицательное число.$\square$




(1.8)

Рассмотрим некоторые свойства расстояний между деревьями в контексте порождения моделей.

**Теорема 2.** При замене в дереве $\Gamma_0(V_0)$ поддерева $\Gamma'_0(V'_0)$ поддеревом $\Gamma'_0(V'_0)$ дерева $\Gamma_1(V_1)$ получается дерево $\Gamma_2(V_2)$. Расстояния между деревьями $r_{02} = r(\Gamma_0(V_0), \Gamma_2(V_2))$ и $r_{01} = r(\Gamma_0(V_0), \Gamma_1(V_1))$:

$$\begin{aligned} r_{02} &\leq p'_0 + p'_1, \\ r_{12} &\leq p_1 + p_0 - p'_0 - p'_1. \end{aligned} \tag{1.9}$$

Размер дерева $\Gamma_2$ $|V_2| = p_0 - p'_0 + p'_1$. Размеры общих поддеревьев $p_{02}$ и $p_{12}$ равны $p_{02} = p_0 - p'_0$ и $p_{12} = p'_1$. Отсюда

$$\begin{aligned} r_{12} &\leq p_1 + p_0 - p'_0 + p'_1 - 2 * (p'_1), \\ r_{02} &\leq 2p_0 - p'_0 + p'_1 - 2 * (p_0 - p'_0). \end{aligned} \tag{1.10}$$

При сокращении получаем искомые формулы.

**Теорема 3.** Расстояние между исходным деревом $\Gamma_0(V_0)$ и порожденным деревом $\Gamma_1(V_1)$, $|V_0| = |V_1| = p_0$, полученным с помощью операции замены одной вершиной дерева, не более чем $2(p_0 - \frac{p_0 - 1}{k+1})$, где $k$ — максимальное число аргументов среди порождающих функций $g$, составляющих деревья $\Gamma_0$ и $\Gamma_1$.

**Доказательство.** Рассмотрим вершину дерева, подвергшуюся операции замены. Максимальное количество ребер, соединенных с этой вершиной, составляет $k+1$ - одно ребро приходит в эту вершину из родительской вершины, остальные соответствуют её потомкам. Таким образом, при изъятии этой вершины и всех ребер, соединенных с ней, из графа, получится не более $k+1$ несвязных графов. Согласно принципу Дирихле, количество вершин в максимальном по размеру из этих графов не менее, чем $\frac{p_0-1}{k+1}$. Такой граф будет общим подграфом графов $\Gamma_0$ и $\Gamma_1$. Значит, расстояние между графами будет не более, чем $2(p_0 - \frac{p_0-1}{k+1})$.□

## 1.3. Сложность суперпозиций

Качество регрессионной модели $f$ оценивается по её внешнему критерию качества $S$. В случае, когда отсутствует разбиение выборки $D$ на тестовую и контрольную, предполагается, что улучшение значения критерия качества $S$ модели $f$ может быть достигнуто с помощью порождения более гладких в смысле условия Липшица (1.11) с меньшим показателем Гёльдера $\alpha$ моделей с небольшим увеличением значения функции ошибки $S$. Также более гладкие модели имеют тенденцию показывать лучшее качество при использовании внутреннего критерия.

**Определение 7.** Функция $f$ называется удовлетворяющей условию Липшица с показателем Гёльдера $\alpha$, если существует такая константа $L$, что:

$$|f(x) - f(y)| < L|x - y|^{\alpha}. \tag{1.11}$$

Следует заметить, что при порождении регрессионных моделей, являющихся многочленами от независимых переменных, показатель Гёльдера соответствует степени многочлена.

Описание суперпозиции, представленной в виде дерева, определяется следующими параметрами.

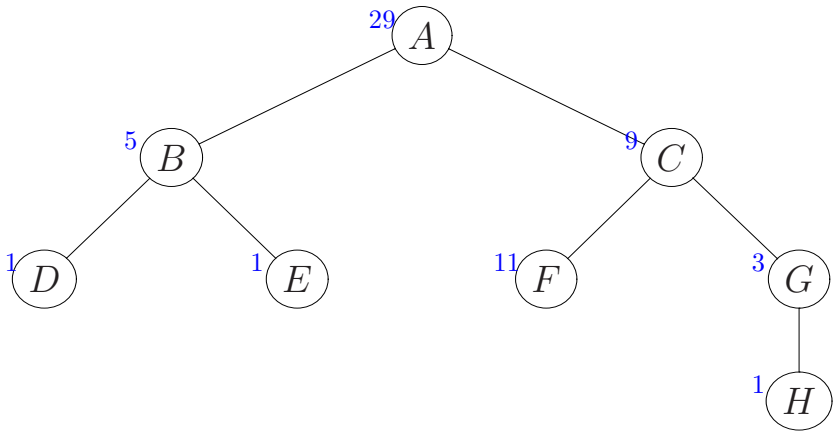
1. Число вершин $v_i$ в дереве Г.
2. Количество порождающих функций $g_i$, не являющихся терминальными вершинами $x_j$.
3. Число элементов вектора весов $\mathbf{w}$.

Для того, чтобы различать суперпозиции с различными показателями Гельдера, предлагается построить определение сложности суперпозиции, которое будет штрафовать длинные вложенные выражения. Иначе, в случае порождения многочленов, схожая сложность будет у суммы нескольких слагаемых и длинного выражения вложенных функций, а показатели Гельдера у этих функций разные.

Предложим определение сложности суперпозиции, позволяющее штрафовать суперпозиции с большим числом вложенных функций. Введем понятие сложности вершины.

**Определение 8.** Сложность $C$ суперпозиции $f$ равна сложности дерева Г, соответствующего ей, и определяется как сумма количества элементов во всех поддеревьях дерева Г.

Таким образом штрафуется суперпозиция, содержащая большое число вложенных функций. Определение позволяет вычислять сложность, производя обход дерева снизу вверх обратно обходу дерева «в глубину» — сложность родительской вершины равна удвоенной сложности вершин потомков плюс единица. Сложность корня и будет сложностью всей суперпозиции, $C(1,1) = C(f)$.




(1.12)

**Теорема 4.** Для частного случая, в котором единственными функциями порождения являются функции сложения, умножения и константы (с помощью данного множества можно породить любой многочлен), дерево, соотвутствующее многочлену степени $k$ будет иметь минимальную сложность порядка $2^k$.

Данное утверждение легко доказывается по индукции, показателем которой является степень многочлена $k$. Отсюда следует, что функции с большим показателем Гёльдера, получают штраф к сложности.

Выведем формулу сложности дерева, являющегося результатом процедуры обмена поддеревьями.

**Теорема 5.** Сложность $C_{\Gamma_1}$ дерева $\Gamma_1$, полученного заменой в дереве $\Gamma_0$ поддерева $\Gamma'_0$ на поддерево $\Gamma'_1$ с корнем в вершине $(d, i)$. В случае подобной замены сложность $C_{\Gamma_1}$ дерева $\Gamma_1$ будет равна

$$C_{\Gamma_1} = C_{\Gamma_0} + d(C_{\Gamma'_1} - C_{\Gamma'_0}).$$

При подсчете сложности дерева $C_{\Gamma_1}$ размер всех поддеревьев, содержавших $C_{\Gamma'_0}$ увеличивается на $C_{\Gamma'_1} - C_{\Gamma'_0}$. При этом поддерево $C_{\Gamma'_0}$ встречается в $d$ деревьях, каждое из которых соответствует одной из

вершин на пути от корня дерева до вершины $(d, i)$. Таким образом, получается требуемая формула.□

## 1.4. Нелинейность моделей

Альтернативным способом определения сложности модели может быть задание её сложности как минимальной степени полинома, достаточным образом приближающего модель. Для поиска данного полинома наиболее удобным способом является использование системы ортогональных полиномов. Последовательностью ортогональных полиномов называют бесконечную последовательность действительных многочленов $p_n(x)$, где каждый многочлен $p_i(x)$ имеет степень $i$, а также любые два различных многочлена этой последовательности ортогональны друг другу в смысле некоторого скалярного произведения, заданного в пространстве $L^2$. В данной работе будут использоваться ряд ортогональных полиномов Чебышева, их скалярное произведение имеет вид

$$(f, g) = \int_{-1}^{1} \sqrt{1 - x^2} f(x) g(x) dx. \tag{1.13}$$

Эта система полиномов также является ортонормированной при заданной операции скалярного произведения (1.13). Многочлен Чебышёва $T_n(x)$ характеризуется как многочлен степени $n$ со старшим коэффициентом $2^{n-1}$, который меньше всего отклоняется от нуля на интервале $[-1, 1]$. Многочлены Чебышёва $T_n(x)$ могут быть определены с помощью рекуррентного соотношения:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x). \end{aligned}$$

Данные полиномы обладают несколькими важными свойствами:

1. Многочлены чётных степеней являются чётными функциями, нечётных — нечётными функциями.
2. Ортогональность по отношению к соответствующим скалярному произведению (с весом $\frac{1}{\sqrt{1-x^2}}$).
3. Среди всех многочленов, значения которых на отрезке [-1,1] не превосходят по модулю 1, многочлен Чебышёва имеет наибольший старший коэффициент.

Следует задать способ определения полинома, наилучшим образом приближающего модель.

**Определение 9.** Полином $P(x)$ приближает непрерывную функцию $f(x)$ на интервале $[a, b]$ с точностью $\varepsilon$, если

$$max\, x \in [a, b]|f(x) - P(x)| \leq \varepsilon. \tag{1.14}$$

В контексте задачи данное определение меняется в связи с ограниченным количеством элементов $x \in [a, b]$. Помимо этого, данное определение должно быть усилено тем, что приближающий полином обладает минимальной степенью среди всех полиномов, для которых выполняется неравенство 1.14. Данное понятие позволяет нам индуктивно определить степень нелинейности модели.

**Определение 10.** Пусть множество порождающих функций $G$ суперпозиции $f$ состоит из элементарных функций, а также простых бинарных операций и операции возведения в степень. Для данной точности $\varepsilon$ и данного отрезка $[a, b]$, на котором будут рассматриваться значения всех порождаемых суперпозиций, нелинейность суперпозиции $f$ с деревом $\Gamma$, определяется пошагово от листьев дерева по следующим правилам:

1. Нелинейность вершины, порождающая функция которой является константой, равна 0:

$$\mathrm{nl}(\mathrm{const}) = 0.$$

2. Нелинейность вершины, порождающая функция которой является свободной переменной, равна 1:

$$\mathrm{nl}(x_i) = 1.$$

3. Нелинейность внутренней вершины $v_1$, соответствующей унарной функции $g_i \in G$, соотносится со сложностью дочерней вершины $v_2$ согласно следующему правилу:

$$\mathrm{nl}(v_1) = \mathrm{nl}(v_2) \cdot n_{g_i},$$

где $n_{g_i}$ является минимальной степенью $P$ полинома Чебышева, приближающего функцию $g_i$ с ошибкой аппроксимации $\varepsilon$. Отдельно рассматривается нелинейность внутренней вершины, соответствующей функции $g_{\exp}(x, w) = w^x \equiv e^{x \ln w}$. Для такой порождающей функции сложность вершины $v_1$ определеяется следующим образом:

$$\mathrm{nl}(v_1) = \mathrm{nl}(e^{v_2 \ln w}) = \mathrm{nl}(v_2) \cdot n_{g_{\exp}},$$

где $n_{g_{\exp}}$ является минимальной степенью $P$ полинома Чебышева, приближающего функцию $e^{x \ln w}$ с ошибкой аппроксимации $\varepsilon$.

4. Нелинейность внутренней вершины $v_1$, соответствующей порождающей функции суммы $g_+$ или разности $g_-$, определяется как максимум нелинейностей её дочерних вершин $v_2$ и $v_3$:

$$\mathrm{nl}(v_1) = \mathrm{nl}(v_2 + v_3) = max(\mathrm{nl}(v_2), \mathrm{nl}(v_3)),$$
$$\mathrm{nl}(v_1) = \mathrm{nl}(v_2 - v_3) = max(\mathrm{nl}(v_2), \mathrm{nl}(v_3)).$$

5. Нелинейность внутренней вершины $v_1$, соответствующей порождающей функции произведения $g_*$, определяется как сумма нелинейностей её дочерних вершин $v_2$ и $v_3$:

$$\mathrm{nl}(v_1) = \mathrm{nl}(v_2 * v_3) = \mathrm{nl}(v_2) + \mathrm{nl}(v_3)$$

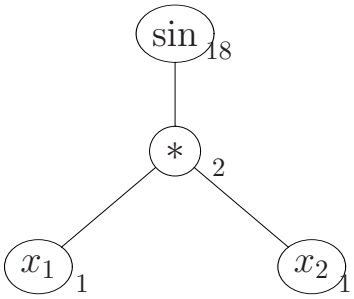
6. Нелинейность внутренней вершины $v_1$, соответствующей порождающей функции деления $g_\div$, определяется следующим образом:

$$\mathrm{nl}(v_1) = \mathrm{nl}(v_2 * v_3) = \mathrm{nl}(v_2) + \mathrm{nl}(v_3) * n_\div,$$

   где $n_\div$ является минимальной степенью полинома Чебышева, приближающего функцию $1/x$ с заданной точностью $\varepsilon$.

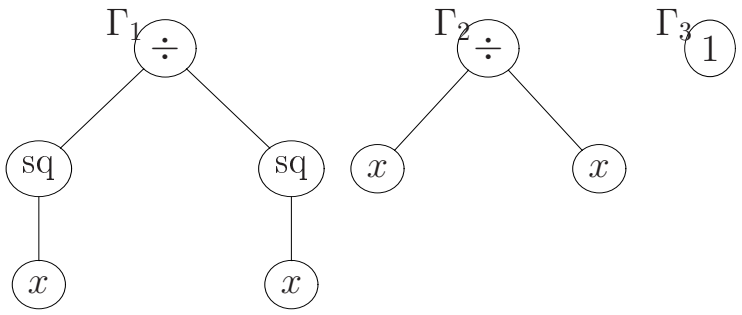
7. Нелинейность корневой вершины соответствует нелинейности модели.

Рассмотрим примеры подсчета нелинейности для различных суперпозиций




(1.15)

Пример подсчета нелинейности модели двух переменных рассматривается на диаграмме 1.15. Рассматривается интервал [0,4]. Нелинейность листьев равна 1. Нелинейность произведения согласно правилу 5 — двум. Нелинейность sin на выделенном интервале равна 9 при точности $\varepsilon = 10^{-6}$. Таким образом, нелинейность корневой вершины равна 18.




(1.16)

$\mathrm{nl} = 12$ $\mathrm{nl} = 6$ $\mathrm{nl} = 0$

В примере 1.16 показывается, что нелинейность модели может зависить от способа записи — на интервале $[1{,}2]$ три модели $\Gamma_1, \Gamma_2, \Gamma_3$ задают одну и ту же поверхность, однако нелинейность данных моделей различна. Таким образом, ограничение нелинейности позволяет нам накладывать штрафы на излишне сложные модели, имеющие менее сложные аналоги.

# Глава 2
# Порождение суперпозиций

Для решения прикладных задач машинного обучения недостаточно определить свойства моделей, необходимо указать конкретный алгоритм построения регрессионных моделей. Интуитивным способом выбора модели является метод перебора моделей — для заданных ограничений строится все множество допустимых суперпозиций, и из них согласно критерию качества выбирается наилучшая модель. Далее будет показано, что такой способ реализуем, однако имеет большую вычислительную сложность. Альтернативным методом порождения модели является алгоритм направленного поиска модели. Данный алгоритм является рандомизированным и не гарантирует нахождение лучшей модели в заданных ограничениях, однако качество его работы оказывается достаточным для решения прикладных задач.

## 2.1. Построение всех возможных суперпозиций

Опишем итеративный алгоритм $\mathfrak{F}$, порождающий суперпозиции ограниченной сложности, не содержащие параметров [1, 51]. Описанный алгоритм породит любую допустимую суперпозицию ограниченной сложности за конечное число шагов.

Пусть дано множество примитивных функций

$$G = \{g_1, \ldots, g_l, \mathrm{id}, \mathrm{Const}\}$$

и множество свободных переменных

$$X = \{x_1, \ldots, x_n\}. \tag{2.1}$$

Алгоритм $\mathcal{F}$ порождения суперпозиций будет итеративным. Перед первым шагом построим начальное значения множества моделей $F_0$ (индекс множества соответствует шагу, на котором получены соответствующие модели):

$$F_0 = \{X, \text{Const}\},$$

где $X$ соответствует выборке (2.1), а Const соотвествует порождающей функции константы. Далее на каждом шаге для множества $F_i$ построим вспомогательное множество $U_i$, состоящее из суперпозиций, полученных в результате применения функций $g_i \in G$ к элементам $f_j \in F_{i-1}$:

$$U_i = \{g_i(f_{j_1}, \dots f_{j_k}), \mid g_i \in G_u, f \in F_{i-1}\}.$$

Тогда множество $F_i$

$$F_i = F_{i-1} \cup U_i.$$

**Теорема 6.** Алгоритм $\mathcal{F}$ породит любую допустимую суперпозицию ограниченной сложности за конечное число шагов.

Данное утверждение можно доказать по индукции, где показателем является высота $h$ дерева $\Gamma$, соответствующего суперпозиции $f$. База индукции: деревья высоты 1 порождаются на нулевом шаге алгоритма. Пусть алгоритм $\mathcal{F}$ порождает все суперпозиции $F_h$, которым соответствуют деревья $\Gamma$ высоты $h$. Возьмем произвольную суперпозицию $f_{h+1}$, дерево $\Gamma_{h+1}$ которой имеет высоту $h+1$. Для каждого поддерева $\Gamma_h$, корни которых являются потомками корня $\Gamma_{h+1}$, утверждение теоремы верно. Значит существует шаг алгоритма $\ell$, на котором все эти деревья уже были порождены. Тогда на шаге $\ell+1$ алгоритма в множестве $U_{\ell+1}$ будет элемент, в котором порождающей функцией $g_j$ будет порождающая функция корня $\Gamma_{h+1}$,

а аргументами — суперпозиции, соответствующие деревьям $\Gamma_h$. Эта суперпозиция и будет искомой суперпозицией $f_{h+1}$.

## 2.2. Количество возможных суперпозиций

Оценим количество суперпозиций, получаемых после каждой итерации алгоритма. Пусть дано $n$ независимых переменных: $|X| = n$, а мощность множества $G$ выразим через мощности его подмножеств функций соответствующей арности:

$$|G_1| = l_1, |G_2| = l_2, \ldots, |G_p| = l_p.$$

На нулевой итерации имеем $P_0 = n$ суперпозиций.

На первой итерации дополнительно порождается:

$$P_1 = l_1 n + l_2 n^2 + \cdots + l_n n^q = \sum_{i=1}^{p} l_i P_0^i.$$

Суммарное число суперпозиций после первой итерации:

$$\hat{P}_1 = P_1 + P_0 = \sum_{i=1}^{p} l_i P_0^i + P_0.$$

Как было замечено ранее, суперпозиции, порожденные на $k$-ой итерации, будут также порождены и на любой следующей после $k$ итерации, поэтому суммарное число суперпозиций после второй итерации будет равно:

$$\hat{P}_2 = \sum_{i=1}^{p} l_i \hat{P}_1^i.$$

После $k$-ой итерации будет порождено

$$\hat{P}_k = \sum_{j=1}^{p} l_i \hat{P}_{k-1}^i.$$

Оценим порядок количества суперпозиций, порожденных после $k$-ой итерации. Пусть в множестве примитивных функций $G$ содержится $l_p$ функций максимальной арности $p > 1$, и имеется $n > 1$ независимых переменных.

**Теорема 7.** Справедлива следующая оценка количества суперпозиций, порожденных алгоритмом $\mathfrak{F}$ после $k$-ой итерации:

$$|\mathcal{F}_k| = \mathcal{O}(l_p^{(p^k-1)/(p-1)} n^{p^k}).$$

Оценим сначала порядок роста для случая, когда есть лишь одна $m$-арная функция и $n$ свободных переменных.

После первой итерации алгоритма будет порождено $n^m + n$ суперпозиций. После второй — $(n^m+n)^m+n^m+n$, что можно оценить как $(n^m)^m = n^{m^2}$. Таким образом, после $k$-ой итерации количество суперпозиций можно оценить как $n^{m^k}$.

Видно, что для оценки скорости роста количества порожденных суперпозиций следует учитывать только функции с наибольшей арностью.

Рассмотрим теперь случай, когда имеется не одна функция арности $m$, а $l_m$ таких функций. Тогда на первой итерации порождается $l_m n^m + n$ суперпозиций, на второй:

$$l_m(l_m n^m + n)^m + l_m n^m + n \approx l_m^{m+1} n^{m^2},$$

на третьей, с учетом этого приближения:

$$l_m(l_m^{m+1} n^{m^2})^m = l_m l_m^{m(m+1)} n^{m^3} = l_m^{m^2+m+1} n^{m^3}.$$

Количество порожденных суперпозиций после $k$-ой итерации можно оценить как как:

$$|F_k| = \mathcal{O}(l_m^{(m^k-1)/(m-1)} n^{m^k}).$$

Таким образом, получаем оценку в общем случае, когда в множестве $G$ содержится $l_p$ функций максимальной арности $p$:

$$|F_k| = \mathcal{O}(l_p^{(p^k-1)/(p-1)} n^{p^k}). \tag{2.2}$$

Следует заметить, что количество моделей растет более чем экспоненциально.□

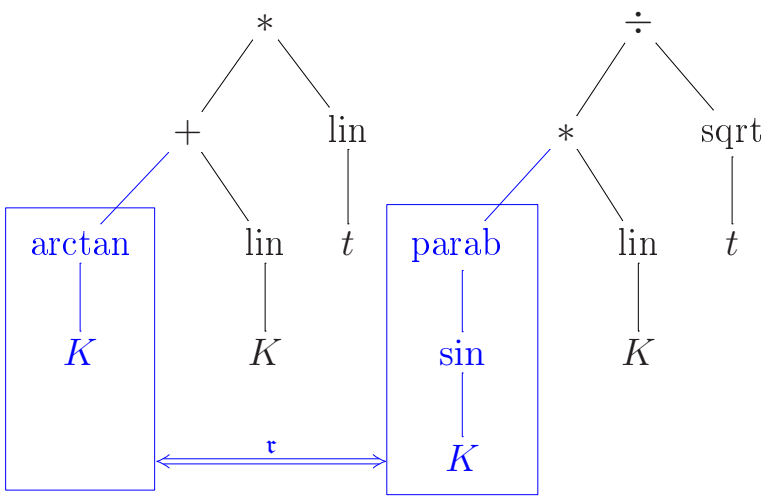
## 2.3. Алгоритм последовательного порождения моделей

Алгоритм $\mathfrak{F}$ порождения все суперпозиций заданной сложности имеет слишком высокую вычислительную сложность из-за большого числа порождаемых нелинейных моделей (2.2) даже при невысокой сложности $C$ порождаемых моделей и поэтому не подходит для решения прикладных задач. В связи с этим предлагается использовать алгоритм $\mathfrak{G}$ направленного поиска оптимальных моделей с использованием случайных операций.

Выбор оптимальной модели происходит на множестве порождаемых моделей на каждой итерации алгоритма $\mathfrak{G}$. Задан начальный набор конкурирующих моделей $F$, в котором каждая модель $f_i$ есть суперпозиция порождающих функций $g_i \in G$. Алгоритм $\mathfrak{G}$ работает итерационно, пока не будет достигнут необходимый уровень значения ошибки $S$ или через определенное число итераций.

1. Некоторым методом оптимизации параметров минимизируются функции ошибки $S_i(w)$ для каждой модели $f_i$. Отыскиваются параметры $\mathbf{w}$ и вычисляется значение функции ошибки $S$ каждой модели.
2. Заданы следующие правила построения производных моделей $f_j$. Для модели $f_j$ строится тождественная ей модель $f'_j$. В дереве $\Gamma'_j$, соответствующем модели $f'_j$, произвольно выбирается

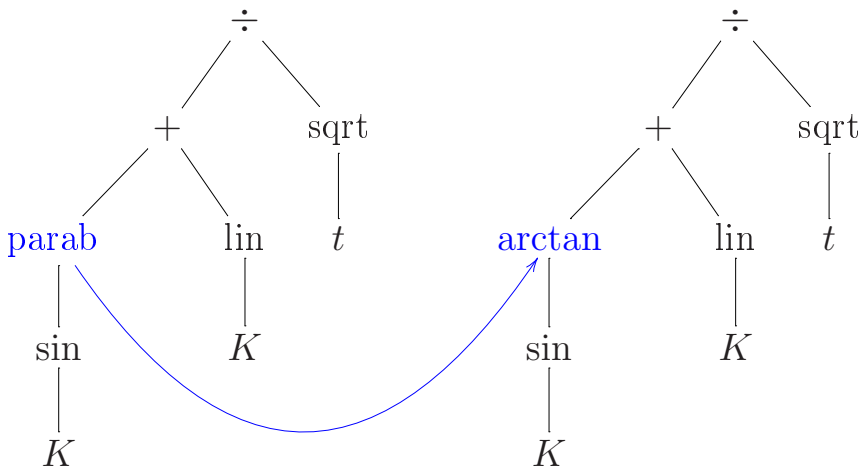
вершина $v_k$. Выбирается произвольная модель $f_m$ из множества $F$ и произвольная вершина $v_l$ её дерева $\Gamma_m$. Модель $f'_j$ модифицируется путем замещения в дереве $\Gamma'_j$ поддерева, корнем которого является вершина функции $v_k$, на поддерево дерева $\Gamma_m$ с корнем в вершине $v_l$. Измененная модель $f'_j$ добавляется в множество $F$:




(2.3)

3. С заданной вероятностью $p_0$ каждая модель $f_j \in F$ подвергается изменениям. В дереве $\Gamma'_j$, соответствующем модели $f'_j$, в соответствии с некоторой заданной функцией распределения выбирается вершина $v_k$. Соответствующая ей порождающая функция $g_k$ заменяется случайным образом выбранной функцией $g_m$ той же арности из множества порождающих функций

$G$:




4. Модели из множества $F$ сортируются в соответствии со значениями функции ошибки $S_i$. Заданная доля наилучших моделей используется в дальнейших итерациях.

## 2.4. Теорема схем для деревьев

Алгоритм $\mathfrak{G}$ последовательного порождения моделей является рандомизированным — таким образом, достижение некоторой оптимальной модели не может быть гарантированно. Однако возможно доказать сходимость по вероятности в некотором пространстве. Для множества моделей может быть рассмотрена структурная сходимость — наличие некоторых элементов во всем множестве рассматриваемых моделей, или в моделях наивысшего качества. Данный подход возникает интуитивно при представлении моделей в виде строк — для них элементами будут подстроки с фиксированными элементами на выбранных позициях. Такие подстроки называются схемами [26]. Для схем формулируется теорема об экспоненциальном увеличении количества моделей, их содержащих.

При представлении моделей в виде деревьев данный подход необходимо изменить, так как при выборе некоторой части модели по-

мимо элементов, в ней содержащихся, оказывается также важна их топология, связанная с взаимным расположением вершин в дереве. Предлагается описать совокупность вершин и их расположения как единый объект и вычислить вероятность сохранения данного объекта при проведении операций на деревьях по аналогии с теоремой схем на строках.

В прямоугольной системе координат на графах определим функции на них. Простейшими подобными функциями может быть функция *имени*, значением которой является порождающая функция, соответствующая вершине, координаты которой являются аргументами функции. Другой полезной функции является функция *размера поддерева*, значением которой является количество вершин в поддереве с корнем в вершине, координаты которой являются аргументами функции.

Другие используемые в работе функции:

- Арность вершины $A(d, i, \Gamma)$, значением которой является количество потомков вершины с координатами $(d, i)$ дерева $\Gamma$ в координатной сетке (1.6).
- Тип вершины $T(d, i, \Gamma)$, значением которой является тип вершины (дополнительная классификация порождающих функций на функции из $G$ и свободные переменные).
- Размер поддерева $S(d, i, \Gamma)$, значением которой является количество элементов в поддереве дерева $\Gamma$ с корнем в вершине $(d, i)$.

Аналогичным образом могут быть введены функции на нескольких деревьях. Примером подобной функции может быть функция размера дерева после замены поддеревьев $S_0(d_1, i_1, d_2, i_2, h_1, h_2)$, значением которой является размер дерева, полученного после замены

поддерева с координатами корня $(d_1, i_1)$ в дереве $h_1$ на поддерево с корнем в $(d_2, i_2)$ дерева $h_2$. Данная функция принимает значения:

$$S_0(d_1, i_1, d_2, i_2, h_1, h_2) = S(0, 0, h_1) - S(d_1, i_1, h_1) + S(d_2, i_2, h_2).$$

Одной из используемых операций в алгоритме $\mathfrak{G}$ последовательного порождения моделей является операция замены поддерева. Вершина, являющаяся корнем заменяемого поддерева, выбирается в соответствии с каким либо распределением с плотностью $p(\Gamma)$. При использовании прямоугольной системы координат плотность вероятности $p(d, i|\Gamma)$ такого распределения можно выразить функцией:

$$\begin{gathered} p(d, i|\Gamma) = \Pr\{\text{Вершина в слое } d \\ \text{с порядковым номером } i \text{ выбрана в дереве } \Gamma\}, \end{gathered} \tag{2.4}$$

при этом предполагается, что $p(d, i|\Gamma)$ равна нулю для всех координат $(d, i)$, не присутствующих в дереве $\Gamma$.

С помощью обозначения (2.4) можно удобно представить, например, вероятность выбора вершины уровня $d$. Для этого надо просуммировать вероятности (2.4) по второй координате:

$$p(d|\Gamma) = \sum_{i \in \mathbb{N}} p(d, i|\Gamma).$$

Через вероятность (2.4) оценивается математическое ожидание номера слоя, из которого выбирается вершина

$$\mathsf{E}(d|\Gamma) = \sum_{d \in \mathbb{N}} d p(d|\Gamma).$$

Также плотность вероятности $p(d, i|\Gamma)$ используется для оценки математического ожидания значения функций на вершинах, к примеру средний размер поддерева при выборе вершины в дереве $\Gamma$:

$$\mathsf{E}(S(d, i, \Gamma)|\Gamma) = \sum_{d \in \mathbb{N}} \sum_{i \in \mathbb{N}} S(d, i, \Gamma) p(d, i|\Gamma).$$

Через эту величину удобно оценить сложность суперпозиции, соответствующей дереву, как произведение среднего размера поддерева на общее количество вершин в дереве.

Для исследования процедуры замены поддеревьев следует ввести некоторые понятия.

**Определение 11.** Схемой называется дерево, содержащее функции из множества $G \bigcup \{=\}$ и свободные переменные из множества $T \bigcup \{=\}$, где $G$ и $T$ — множества порождающийх функций и свободных переменных соответственно. Порождющая функция $\{=\}$ означает произвольный символ, который может быть любой функцией или свободной переменной. Порядком схемы $O(H)$ называется количество вершин в ней, не являющихся $\{=\}$.

**Определение 12.** Гиперсхемой называется дерево, содержащее функции из множества $G \bigcup \{=\}$ и свободные переменные из множества $T \bigcup \{=, \#\}$. Порождющая функция $\{=\}$ определяется также, как и для схемы, а свободная переменная $\{\#\}$ означает любое допустимое дерево.

Рассмотрим операцию замены поддерева со следующим ограничением: топологическая структура родительских вершин заменяемого и заменяющего поддерева равны, то есть равны арности соответствующих вершин.

**Определение 13.** Операция замены поддерева $\mathfrak{r}$, для которой топологическая структура родительских вершин заменяемого и заменяющего поддерева равны, будет называться операцией замены поддерева с сохранением структуры. Вводится функция $C(d, i, \Gamma_1, \Gamma_2)$, значение которой равно 1 если условие арности вершин выполняется, и 0 если нет. Обозначим данную операцию $\mathfrak{r}_c$.

Таким образом, $C(d,i,\Gamma_1,\Gamma_2)=1$, если $(d,i)=(1,1)$ или

$$\begin{gathered} A(\mathrm{parent}(d,i),\Gamma_1)=A(\mathrm{parent}(d,i),\Gamma_2)\neq 0, \\ A(d,i,\Gamma_1)\geq 0, A(d,i,\Gamma_2)\geq 0, \\ C(\mathrm{parent}(d,i),\Gamma_1,\Gamma_2)=1. \end{gathered} \tag{2.5}$$

Проверяется арность $A$ всех вершин на пути от каждого листа до корня поддерева.

Для операции $\mathfrak{r}_c$ замены поддерева с сохранением структуры оценивается вероятность сохранения схемы $H$:

$$\alpha(H,t)=(1-p_{x0})p(H,t)+p_{x0}\alpha_{x0}(H,t)$$

где

$$\alpha_{x0}(H,t)=\sum_{h_1}\sum_{h_2}\frac{p(h_1,t)p(h_2,t)}{NC(h_1,h_2)}\sum_{i\in C(h_1,h_2)}\delta(h_1\in U(H,i))\delta(h_2\in L(H,i))$$

при этом:

$p_{x0}$ — вероятность проведения операции замены поддерева,

$p(H,t)$ — вероятность выбора вершины из схемы $H$

суммы проходят по всем деревьям из набора,

$NC(h_1,h_2)$ — количество вершин с равной структурой родительских вершин,

$L(H,i)$ — гиперсхема, получаемая из $H$ заменой всех вершин от корня до вершины $i$ вершинами типа $=$, а всех поддеревьев, выходящих из этих вершин — $\#$,

$U(H,i)$ — гиперсхема, получаемая из $H$ заменой поддеревьев ниже точки $i$ вершинами типа $\#$.

Если заменяемая вершина $i$ находится в области равной структуры двух деревьев но не входит в схему $H$, гиперсхемы $L(H,i)$ и $U(H,i)$ — пусты.

Данное утверждение является формулировкой теоремы схем для случая замены поддеревьев с сохранением структуры. Формулировки для теорем, посвященных случаям произвольной замены поддеревьев, следует смотреть в [42].

# Глава 3
# Трансформация моделей

При порождении моделей в общем случае одному и тому же отображению соответствуют суперпозиции различной сложности, например одно и то же отображение соответствует моделям $x$ и $\sqrt[3]{x^3}$. Также возможны случаи порождения деревьев, некоторые ветви которых не оказывают влияния на значение функции (например, умножаются на 0). Данная проблема оказывается важной для многих классов задач, например, для построения логических функций или для задачи угадывания функции [18]. Для понимания, как упрощать подобные суперпозиции, следует ввести понятие эквивалентности моделей.

**Определение 14.** Модель $f_2$ с вектором параметров $\mathbf{w}_2$ называется обобщающей для модели $f_1$ с вектором параметров $\mathbf{w}_1$, если для любого вектора $\mathbf{w}_1$ найдется такой вектор $\mathbf{w}_2$, что для любого $\mathbf{x} \in D$ значения функций $f_1(\mathbf{w}_1, \mathbf{x})$ и $f_2(\mathbf{w}_2, \mathbf{x})$ равны:

$$\mathbf{x} \in D \Rightarrow f_1(\mathbf{w}_1, \mathbf{x}) = f_2(\mathbf{w}_2, \mathbf{x}).$$

**Определение 15.** Модели $f_1$ и $f_2$ с векторами параметров $\mathbf{w}_1$ и $\mathbf{w}_2$ называются эквивалентными, если каждая из них является обобщающей для другой.

Для построения оптимальной модели $f$ ограниченной сложности $C(f) < C_0$ необходимо найти способ трансформации модели $f$ большей структурной сложности в модель меньшей сложности $f'$ с помощью специального алгоритма упрощения. Алгоритм упрощения модели $f(\mathbf{w}, \mathbf{x})$ минимизирует сложность суперпозиции, соответствующей её дереву, при условии, что результирующая модель $f'(\mathbf{w}', \mathbf{x})$

является обобщающей моделью для исходной модели $f(\mathbf{w}, \mathbf{x})$. При проведении данной операции какие-либо вершины и ребра из дерева, соответствующего трансформируемой модели $f$ будут удалены, и будут построены другие вершины и ребра вместо них. Обобщим алгоритм упрощения на орграфы любого вида, а не только на деревья. Далее для каждого графа подразумевается, что это орграф.

**Определение 16.** Подграф $L$, удаляемый из графа $G$ в алгоритме упрощения, будет называться заменяемым подграфом.

**Определение 17.** Создаваемый подграф $R$, помещаемый в граф $G$ в алгоритме упрощения, называется замещающим подграфом.

Существуют, по меньшей мере, два широко используемых метода упрощения моделей: «алгебраическое упрощение», являющееся частным случаем алгебраической трансформации графов и «упрощение эквивалентным решением» [39].

## 3.1. Алгебраический подход к трансформации графов

Определение трансформации графа как замены одного подграфа на другой является интуитивно понятным, однако нестрогим. Для использования математического аппарата теории категория следует строго определить трансформацию графа.

**Определение 18.** Трансформацией $\mathfrak{f}$ на множестве графов $\Gamma$ является пара гиперсхем $H_1$ и $H_2$, функция поиска $\mathfrak{m}$, ставящая в соответствие гиперсхеме $H_1$ подграф $\Gamma$, соответствующий этой гиперсхеме, и взаимно-однозначное отображение $f$, ставящее в соответствие корню и листьям $H_1$ корень и листья $H_2$. При этом порождающие функции, соответствующие этим вершинам, должны совпадать.

Каждой трансформации, таким образом, может быть поставлена в соответствие обратная трансформация $\mathfrak{f}^{-1}$.

В рамках алгебраического подхода к трансформации графов следует ввести категорию трансформаций графов $\mathfrak{G}$, объектами которой являются графы $\Gamma$, а стрелками - трансформации графов $\mathfrak{f}$. Рассмотрим аксиомы категории:

1. $\mathfrak{G}$-объектами в данной категории являются множества графов $\Gamma$.
2. $\mathfrak{G}$-стрелками в данной категории являются трансформации графов $\mathfrak{f}_i$.
3. Функции $\mathrm{dom}(\mathfrak{f})$ и $\mathrm{cod}(\mathfrak{f})$ для трансформаций графов определяются с помощью функции поиска $\mathfrak{m}$. $\mathrm{cod}(\mathfrak{f})$ может быть найден как $\mathrm{dom}(\mathfrak{f}^{-1})$.
4. Ассоциативность следует из наличия обратной функции.
5. Единицей является трививальная трансформация с гиперсхемами $H_1 = H_2 = \#$.

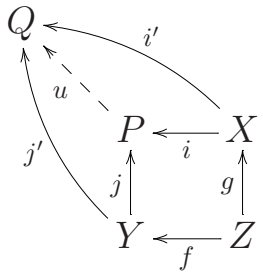
Алгебраический подход к трансформации графов основывается на конструкции кодекартова квадрата морфизмов.

**Определение 19.** Кодекартов квадрат морфизмов $f : Z \to Y$ и $g : Z \to X$ — это объект $P$ и два морфизма $i : X \to P$ и $j : Y \to P$, для которых следующая диаграмма коммутативна:

$$\begin{array}{ccc} P & \xleftarrow[i]{} & X \\ {\scriptstyle j}\uparrow & & \uparrow{\scriptstyle g} \\ Y & \xleftarrow[f]{} & Z \end{array} \tag{3.1}$$

Кодекартов квадрат $(P, i, j)$ является универсальным среди объектов, для которых диаграмма (3.1) коммутативна. То есть, для любой $(Q, i', j')$, такого что предыдущая диаграмма коммутирует, существует единственный морфизм $u : P \to Q$, делающий следующую

диаграмму коммутативной:




(3.2)

Как и любой универсальный оператор, кодекартов квадрат определен с точностью до изоморфизма.

**Определение 20.** Кодекартов квадрат морфизмов $f : Z \to X$ и $g : Z \to Y$ — это копредел диаграммы $X \leftarrow Z \to Y$.

В контексте категории графов, используемой в данной работе, кодекартов квадрат является дизъюнктивной суммой множеств графов $X$ и $Y$, при этом элементы с общим прообразом множестве в $Z$ склеиваются, то есть для каждого графа —элемента множества $Z$, образы его вершин и ребер относительно преобразований $i \cdot g$ и $j \cdot f$ будут совпадать. В рамках данной работы вместо термина «кодекартов квадрат» также будет использоваться термин-синоним «склейка». Трансформация графа может строится сразу как два кодекартовых квадрата. Данный подход называется двойной склейкой в противоположность к однократной склейке. Оба подхода описаны ниже. В процессе трансформации графов каждый граф $\Gamma_1$ — элемент множества $X$ является заменяемым и заменяющим подграфом, каждый граф $\Gamma_2$ — элемент множества $Y$ — неизменной частью этого графа, а элементы $Z$ — общей частью заменяемого и заменяющего подграфов. Естественным образом вводится операция соединения графов $\Gamma_1$ и $\Gamma_2$, результатом которой является объединение множеств, при этом соответствующие вершины и ребра накладываются друг на друга.

### 3.1.1. Трансформация двойной склейкой

Для рассмотрения трансформации графов необходимо ввести понятие правила, построенного в виде кодекартовом квадрате морфизмов. Множества графов $\Lambda$ и $\Phi$ являются в схеме кодекартова квадрата множеством $X$, множество граф $\Psi$ - множеством $Z$, а множеству $Y$ соответствует $\Delta$. Множеству $P$ для двух квадратов соответствуют начальный и конечный графы $\Gamma$ и $\Omega$.

**Определение 21.** Правило — это тройка $p = (\Lambda, \Psi, \Phi)$, где $\Lambda$ и $\Phi$ являются заменяемым и замещающим подграфами и граф $\Psi$ является общей частью подграфов $\Lambda$ и $\Phi$, то есть их пересечением. Заменяемый, или начальный подграф $\Lambda$ называется условием применения правила, замещающий, или конечный подграф $\Phi$ — итогом его применения. Подграф $\Psi$ описывает часть графа, необходимую для применения правила, но неизменную в процессе применения. Множество $\Lambda \setminus \Psi$ является удаляемой частью графа, вместо неё создается множество $\Phi \setminus \Psi$.

**Определение 22.** Процедура поиска $\mathfrak{m}$ — отображение из $\Lambda$ в $\Gamma$, ставящая в соответствие заменяемому графу эквивалентный ему подграф. При этом процедура $\mathfrak{m}$ сохраняет структуру графа $\Gamma$.

**Определение 23.** Трансформация графа — это пара, элементами которой являются правило $p$ и процедура поиска $\mathfrak{m}$. Процедура трансформации графа $\Gamma$ в граф $\Omega$ с помощью правила $p$ и процедуры поиска $\mathfrak{m}$ будет также обозначаться как $\Gamma \overset{p,\mathfrak{m}}{\to} \Omega$.

Процедура трансформации графа правилом $p$ и процедурой поиска $\mathfrak{m}$ состоит из двух шагов. На первом шаге все ребра и вершины, соответствующие множеству $\Lambda \setminus \Psi$ удаляется из графа $\Gamma$. Удаляемая часть может не являться графом, но оставшаяся структура

$\Delta = \{\Gamma \setminus \mathfrak{m}(\Lambda)\} \bigcup \mathfrak{m}(\Psi)$ должна оставаться графом, т.е. в ней не должно быть подвешенных ребер. Таким образом, процедура поиска $\mathfrak{m}$ должна удовлетворять условию соединения графов, то есть результатом соединения $\Lambda \setminus \Psi$ и $\Delta$ является граф $\Gamma$ (см. диаграмму 3.3). На втором шаге трансформации граф $\Delta$ соединяется с графом $\Phi \setminus \Psi$ для образования производного графа $\Omega$ (см. диаграмму 3.3). Так как подграфы $\Lambda$ и $\Phi$ могут иметь пересечение $\Psi$, подграф $\Psi$ существует и в начальном графе $\Gamma$ и не удаляется на первом шаге, т.е. существует и в промежуточном графе $\Delta$. Для присоединения новых ребер и вершин к графу $\Delta$ используется граф $\Psi$. Таким образом определяются присоединенные вершины с помощью которых граф $\Phi$ присоединяется к графу $\Delta$. Для получения графа оптимальной структуры процедура одиночной трансформации графа может быть выполнена несколько раз.

Формально трансформация графа задается следующим образом. Пусть дано правило:

$$p = (\Lambda \leftarrow \Psi \rightarrow \Phi)$$

и промежуточный граф $\Delta$, который включает в себя $\Psi$, тогда исходный граф $\Gamma$ трансформации $\Gamma \rightarrow \Omega$ с помощью правила $p$ это соединение $\Lambda$ и $\Delta$ с помощью $\Psi$:

$$\Gamma = \Lambda +_{\Psi} \Delta,$$

а результирующий граф $\Omega$ определяется как соединение $\Phi$ и $\Delta$ с помощью $\Psi$:

$$\Omega = \Phi +_{\Psi} \Delta.$$

Более точно, используются морфизмы

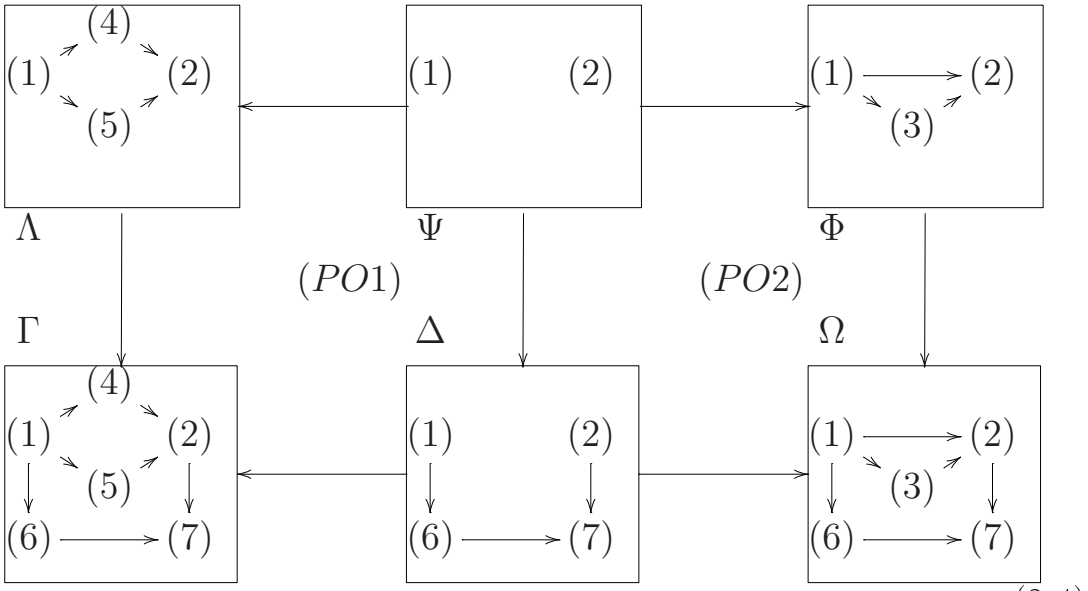
$$r : \Psi \rightarrow \Lambda, \qquad l : \Psi \rightarrow \Phi, \qquad k : \Psi \rightarrow \Delta$$

для того, чтобы показать, каким образом $\Psi$ входит в $\Lambda$,$\Phi$ и $\Delta$ соответственно. Данный способ построения начального графа $\Gamma$ и конечного графа $\Omega$ позволяет определить конструкции соединения $\Gamma = \Lambda +_\Psi \Delta$ и $\Omega = \Phi +_\Psi \Delta$ как конструкции склейки (см. диаграмму 3.3). Таким образом, диаграмма 3.3 является двойным кодекартовым квадратом. Результирующий морфизм $\mathfrak{n} : \Phi \to \Omega$ называется ко-поиском трансформации $\Gamma \to \Omega$. Данная функция является функцией поиска в графе $\Omega$ подграфа, изоморфного заменяющему подграфу $\Phi$. Коммутативная диаграмма для трансфомации графа строится следующим образом:

$$\begin{array}{ccccc} \Lambda & \xleftarrow{r} & \Psi & \xrightarrow{l} & \Phi \\ \downarrow{\scriptstyle \mathfrak{m}} & (1) & \downarrow{\scriptstyle k} & (2) & \downarrow{\scriptstyle \mathfrak{n}} \\ \Gamma & \xleftarrow{g} & \Delta & \xrightarrow{h} & \Omega \end{array} \tag{3.3}$$

Для применени правила $p$ с процедурой поиска $\mathfrak{m}$ подграфа $\Lambda$ в графе $\Gamma$, при заданном морфизме $\mathfrak{m} : \Lambda \to \Gamma$, как показано на коммутативной диаграмме (3.3), в первую очередь необходимо построить промежуточный граф $\Delta$, такой что соединение $\Lambda +_\Psi \Delta$ даст результатом граф $\Gamma$. На следующем шаге мы строим соединение $\Phi +_\Psi \Delta$ графов $\Phi$ и $\Delta$ с помощью графа $\Psi$, получая граф $\Omega$ и, таким образом, получаем процедуру двойной склейки $\Gamma \to \Omega$ с помощью правила $p$ и процедуры поиска $\mathfrak{m}$. Для первого шага необходимо выполнение условия соединения графов, что позволяет нам построить $\Delta$ из условия $\Gamma = \Lambda +_\Psi \Delta$. Для процедуры $\mathfrak{m}$ условие соединения означает, что все подвешенные вершины $\Lambda$, то есть вершины $v \in \Lambda$ такие что $\mathfrak{m}(v)$ является начальной или конечной вершиной некоторого ребра $e$, принадлежащего $\Gamma \setminus \Lambda$, должны быть в $\Psi$.

Рассмотрим пример двойной склейки:




(3.4)

Данная диаграмма соответствует общей схеме 3.3. Следует заметить, что в диаграмме 3.3 граф $\Gamma$ является соединением графов $\Lambda$ и $\Delta$ с помощью $\Psi$, причем обозначения вершин показывают как вершины размечаются при применении морфизмов.

Рассмотрим условие корректности построения структуры графа $\Delta$. Разметка ребер может быть единственным образом выведена из разметки вершин. Условие соединения вершин выполнено на диаграмме 3.4, потому что подвешенные вершины (1) и (2), принадлежащие $\Lambda$, также являются соединительными вершинами. Таким образом, не остается подвешенных ребер, выходящих из вершин (1) и (2). При этом граф $\Omega$ является соединением графов $\Phi$ и $\Delta$ вместе с $\Psi$, что приводит к трансформации $\Gamma \to \Omega$ с помощью правила $p$. Фактически, диаграммы 3.3 и 3.4 являются кодекартовыми квадратами в категории графов, состоящей из графов и морфизмов на них.

Сформулируем точное условие соединения графов при трансформации графа. Для этого вводим следующие определния:

**Определение 24.** Точки соединения — вершины и ребра в $\Lambda$, которые не удаляются при применении правила $p$.

**Определение 25.** Точки обнаружения — вершины и ребра в $\Lambda$, образы которых относительно $\mathfrak{m}$ имеют более одного прообраза.

**Определение 26.** Подвешенные вершины — вершины в $\Lambda$, образы которых относительно $\mathfrak{m}$ в $\Gamma$ имеют входящие или выходящие ребра, не содержащиеся в $\Lambda$.

В данных определениях условие соединения графа выглядит следующим образом:

**Теорема 8.** Пусть дано правило $p = (\Lambda \leftarrow \Psi \rightarrow \Phi)$, граф $\Gamma$ и процедура поиска $\mathfrak{m} : \Lambda \rightarrow \Gamma$. Вершины графов обозначаются буквой $V$, ребра — $E$. Тогда правило $p$ с процедурой поиска $\mathfrak{m}$ удовлетворяет условию соединения, если все точки обнаружения и подвешенные вершины также являются точками соединения.

Докажем данную теорему от противного. Пусть существует подвешенная вершина $v_0$, не являющаяся точкой соединения. Данная вершина удаляется из $\Gamma$ при применении правила $p$. Однако в $\Gamma$ существуют ребра, не содержащиеся в $\Lambda$, и присоединенные к $v_0$. Таким образом, полученный граф будет недопустимым, потому что у некоторых ребер не будет начала или конца. Точки обнаружения являются точками соединения, так как иначе правило будет внутренне противоречивым.$\square$

Ограничения, накладываемые естестенным образом на трансформации двойной склейкой, не позволяют удобно производить многие операции с графами, используемые на практике. Так, операция замены вершины поддерева $v_i$ не может быть описана в виде заменяемого и замещающего графов, состоящих из одной вершины,

т.к. если в заменяемом графе всего одна вершина $v_i$ — эта вершина не будет являться подвешенной, только если весь граф состоит из одной вершины. Таким образом, для применения трансформаций предлагается метод, сопоставляющий неудовлетворяющей условиям трансформации набор допустимых трансформаций.

**Теорема 9.** Любой трансформации $t = (\Lambda_t, \Psi_t, \Phi_t)$ графа соответствует набор правил $p_t = (\Lambda_{p_t}, \Psi_{p_t}, \Phi_{p_t})$, удовлетворяющих условию соединения, такой что любое применение трансформации $t$ аналогично применению одного из правил $p_t$.

Данная теорема доказывается конструктивно — рассматриваются все возможные наборы количеств ребер, которые могут иметь точки соединения $v_c$, и для каждого набора создаются заменяемый и замещающий подграфы, в который добавляются вершины типа # на концах всех ребер, выходящих из $v_c$ и не содержавшихся ранее в заменяемом подграфе $\Lambda$.$\square$

Морфизмы $\Psi \to \Lambda$ и $\Psi \to \Phi$ в произведениях могут быть ограничены как инъективные морфизмы — каждому образу в $\Lambda$ и $\Phi$ соответствует только один проообраз из $\Psi$. Тем не менее, возможны неинъективные варианты процедур поиска $\mathfrak{m} : \Lambda \to \Gamma$ и ко-поиска $\mathfrak{n} : \Phi \to \Omega$. Это может быть особенно важным, когда рассматривается парралельное применение правил:

$$p_1 \bigoplus p_2 : \Lambda_1 \bigoplus \Lambda_2 \leftarrow \Psi_1 \bigoplus \Psi_2 \to \Phi_1 \bigoplus \Phi_2,$$

где $\bigoplus$ означает дизъюнктивное объединение. Даже для инъективных вариантов $\mathfrak{m}_1 : \Lambda_1 \to \Gamma$ с помощью $p_1$ и $\mathfrak{m}_2 : \Lambda_2 \to \Gamma$ с помощью $p_2$, итоговая операция $\mathfrak{m} : \Lambda_1 + \Lambda_2 \to \Gamma$ не является инъективной, если образы процедур поиска $\mathfrak{m}_1(\Lambda_1)$ и $\mathfrak{m}_2(\Lambda_2)$ имеют непустое пересечение в $\Gamma$.

**Теорема 10.** Существует набор трансформаций $(p_1, \mathfrak{m}_1)$ и $(p_2, \mathfrak{m}_2)$, такой что их параллельное применение имеет неинъективную функцию поиска $\mathfrak{m} = \mathfrak{m}_1 \bigoplus \mathfrak{m}_2$.

Построим пример таких трансформаций. Пусть трансформация преобразует дерево $\Gamma_0$, соответствующее функции

$$f_0 = (x+1) * (x - 1 + x^2 - x + 1),$$

и есть два правила

$$p_1 = \{(x+1)(x-1), x^2 - 1\} \text{ и } p_2 = \{(x+1)(x^2 - x + 1), x^3 + 1\}.$$

В обоих этих правилах подграф $\Psi$ является пустым. Обе процедуры поиска $\mathfrak{m}_1$ и $\mathfrak{m}_2$ будут находить часть суперпозиции $(x+1)$. Из этого следует, что при объединении $\Lambda_1$ и $\Lambda_2$ у одного образа из $\Gamma$ будет более одного прообраза. То есть объединенное правило $p_{12}$ дважды найдет в графе $\Gamma_0$ подграф, соотвествующий суперпозиции $(x+1)\cdot$.$\square$

Для рассмотрения случаев применения нескольких трансформаций необходимо определить условие, при котором трансформации могут применяться последовательно и параллельно. Введем понятия параллельно и последовательно независимых трансформаций.

**Определение 27.** Две трансформации графов $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Gamma \overset{p_2,m_2}{\to} \Omega_2$ являются параллельно независимыми, если все вершины и ребра, попадающие в образ обоих морфизмов поиска, являются соединительными:

$$\mathfrak{m}_1(\Lambda_1) \bigcap \mathfrak{m}_2(\Lambda_2) \subseteq \mathfrak{m}_1(l_1(\Psi_1)) \bigcap \mathfrak{m}_2(l_1(\Psi_2))$$

Две трансформации графов $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Omega_1 \overset{p_2,m_2}{\to} \Omega_2$ являются последовательно независимыми, если все вершины и ребра, попадающие в пересечение морфизмов $\mathfrak{n}_1$ и $m_2$, являются соединительными:

$$\mathfrak{n}_1(\Phi_1) \bigcap \mathfrak{m}_2(\Lambda_2) \subseteq \mathfrak{n}_1(r_1(\Psi_1)) \bigcap \mathfrak{m}_2(l_2(\Psi_2))$$

Следует заметить, что для графов-деревьев возникает простой достаточный критерий пареллельной и последовательной независимости трансформаций, если их замещаемые графы $\Lambda$ являются односвязными.

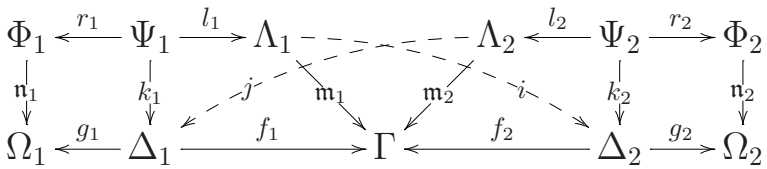
**Теорема 11.** Две трансформации графов-деревьев $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Gamma \overset{p_2,m_2}{\to} \Omega_2$ являются параллельно и последовательно независимыми, если образы корней $v_1$ и $v_2$ деревьев $\mathfrak{m}_1(\Lambda_1)$ и $\mathfrak{m}_2(\Lambda_1)$ не принадлежат друг другу:

$$v_1 \notin \mathfrak{m}_2(\Lambda_2)\ v_2 \notin \mathfrak{m}_1(\Lambda_1) \tag{3.5}$$

Данная теорема простым образом доказывается от противного. Пусть условие 3.5 выполняется и существует вершина $v_0$, принадлежащая пересечению множеств $\mathfrak{m}_1(\Lambda_1)$ и $\mathfrak{m}_2(\Lambda_1)$. Тогда в графе будет цикл, проходящий через вершины $v_1$,$v_0$,$v_2$ и корень дерева. Но в дереве не может быть циклов.$\square$

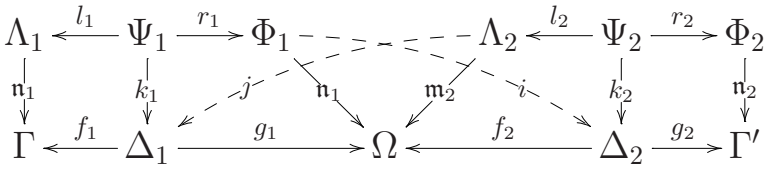
Определение независимости оказывается неудобным для применения, так как оно слабо формализовано. Определим необходимое и достаточное условие, при котором графы являются параллельно или последовательно независимыми через существование соответствующих морфизмов.

**Теорема 12.** Две трансформации графов $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Gamma \overset{p_2,m_2}{\to} \Omega_2$ являются параллельно независимыми, если существуют морфизмы $i : \Lambda_1 \to \Delta_2$ и $j : \Lambda_2 \to \Delta_1$, такие что $f_2 \circ i = \mathfrak{m}_1$ и $f_1 \circ j = \mathfrak{m}_2$:




(3.6)

**Теорема 13.** Две трансформации графов $\Gamma \overset{p_1,m_1}{\to} \Omega$ и $\Omega \overset{p_2,m_2}{\to} \Gamma'$ являются последовательно независимыми, если существуют морфизмы $i : \Phi_1 \to \Delta_2$ и $j : \Lambda_2 \to \Delta_1$, такие что $f_2 \circ i = \mathfrak{n}_1$ и $g_1 \circ j = \mathfrak{m}_2$:




(3.7)

Доказательство. Рассмотрим необходимость и достаточность критерия для параллельной независимости. Для последовательной независимости доказательство будет строиться аналогичным образом. Вершина $v \in \Lambda_1$ или принадлежит множеству $\mathfrak{m}_2(\Lambda_2)$, или лежит вне его. Рассмотрим оба случая:

1. Множество $\mathfrak{m}_1(v) \notin \mathfrak{m}_2(\Lambda_2)$. Все вершины графа $\Gamma$ являются образами при применении отображений $\mathfrak{m}_2$ или $f_2$. Отсюда $\mathfrak{m}_1(v) \in f_2(\Delta_2)$.
2. Множество $\mathfrak{m}_1(v) \in \mathfrak{m}_2(\Lambda_2)$. Тогда $\mathfrak{m}_1(v) \in \mathfrak{m}_1(\Lambda_1) \cap \mathfrak{m}_2(\Lambda_2) \subseteq \mathfrak{m}_1(l_1(\Psi_1)) \cap \mathfrak{m}_2(l_2(\Psi_2))$. При этом из коммутативной диаграммы следует, что $\mathfrak{m}_2(l_2(\Psi_2)) = f_2(k_2(\Psi_2))$. Отсюда $\mathfrak{m}_1(v) \in f_2(\Delta_2)$.

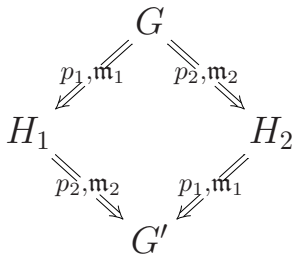
В обоих случаях оказывается, что $\mathfrak{m}_1(x) \in f_2(\Delta_2)$, так что инъективность $f_2$ позволяет нам определить $i(x) = f_2^{-1} \circ \mathfrak{m}_1(x)$. Аналогично, $j$ определяется из условия $f_1 \circ j = \mathfrak{m}_2$.

При данных $i, j$ с $f_2 \circ i = m_1$ и $f_1 \circ j = m_2$, пусть $y \in \mathfrak{m}_1(\Lambda_1) \bigcap \mathfrak{m}_2(\Lambda_2)$. Тогда $y \in \mathfrak{m}_1(L_1) \bigcap f_1(j(\Lambda_2))$. Из условия кодекартова квадрата следует, что существует $z_1 \in \Psi_1$, такое что $y = \mathfrak{m}_1(l_1(z_1)) = f_1(k_1(z_1))$. Значит $y \in \mathfrak{m}_1(l_1(\Psi_1))$, аналогично $y \in \mathfrak{m}_2(l_2(\Psi_2))$, откуда следует условие независимости $\mathfrak{m}_1(\Lambda_1) \bigcap \mathfrak{m}_2(\Lambda_2) \subseteq \mathfrak{m}_1(l_1(\Psi_1)) \bigcap \mathfrak{m}_2(l_1(\Psi_2))$.$\square$

С использованием данных критериев можно определить, как связаны друг с другом независимые параллельно и последовательно трансформации. Данная теорема является частным случаем теоремы Чёрча-Россера

**Теорема 14.** Пусть даны две параллельно независимых трансформации графов $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Gamma \overset{p_2,m_2}{\to} \Omega_2$. Тогда существует граф $\Gamma'$, такой что $\Omega_1 \overset{p_2,m_2}{\to} \Gamma'$ и $\Omega_2 \overset{p_1,m_1}{\to} \Gamma'$, а пары трансформаций $\Gamma \overset{p_1,m_1}{\to} \Omega_1$, $\Omega_1 \overset{p_2,m_2}{\to} \Gamma'$ и $\Gamma \overset{p_2,m_2}{\to} \Omega_2$,$\Omega_2 \overset{p_1,m_1}{\to} \Gamma'$ являются последовательно независимыми.

Пусть даны последовательно независимые трансформации $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Omega_1 \overset{p_2,m_2}{\to} \Gamma'$. Тогда существует граф $\Omega_2$ такой что $\Gamma \overset{p_2,m_2}{\to} \Omega_2$ и $\Omega_2 \overset{p_1,m_1}{\to} \Gamma'$, а трансформации $\Gamma \overset{p_1,m_1}{\to} \Omega_1$ и $\Gamma \overset{p_2,m_2}{\to} \Omega_2$ являются параллельно независимыми.




(3.8)

Доказательство данной теоремы следует смотреть в [21].□

Для рассмотрения свойств трансформаций следует рассмотреть операцию трансформации графа в алгебраическом подходе.

**Теорема 15.** Множество независимых трансформаций $(p, \mathfrak{m})$ на графах $\Gamma$ относительно операции произведения является абелевой группой.

Рассмотрим выполнение аксиом групп для трансформаций.

1. Ассоциативность и коммутативность применения правил следует из ассоциативности операции дизъюниктивной суммы $\bigoplus$.

2. Нейтральным элементом является трансформация с пустыми множествами $\Lambda$ и $\Phi$. Такая трансформация, очевидно, независима с любой другой.
3. Обратным элементом для трансформации $(p = (\Lambda \leftarrow \Psi \rightarrow \Phi), \mathfrak{m})$ с ко-поиском $\mathfrak{n}$ будет трансформация $(p = (\Phi \leftarrow \Psi \rightarrow \Lambda), \mathfrak{n})$.$\square$

Трансформации графов также можно рассматривать как операции с двумя базовыми множествами $V$ (вершины) и $E$ (ребра) и дополнительными операциями $s : E \rightarrow V$(источник) и $t : E \rightarrow V$ (цель). Морфизмы $f$:

$$f : \Gamma_1 \rightarrow \Gamma_2$$

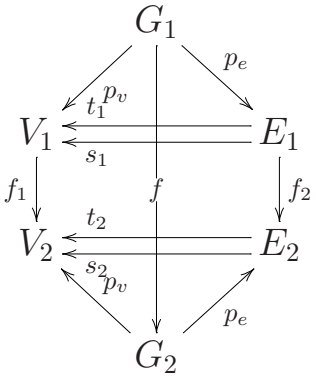
являются частными случаями гомоморфизмов общего вида:

$$f = (f_V : V_1 \rightarrow V_2, f_E : E_1 \rightarrow E_2).$$

Это значит, что $f_V$ и $f_E$ должны быть совместимы с операциями $s$ и $t$, т.е.

$$f_V \circ s_1 = s_2 \circ f_E \text{ и } f_V \circ t_1 = t_2 \circ f_E$$

В виде коммутативной диаграммы данное условие выглядит следующим образом:




(3.9)

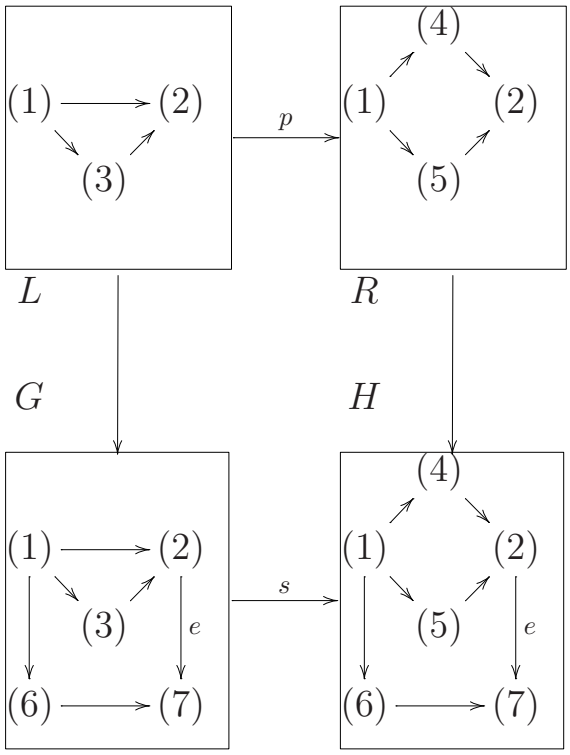
Морфизм $f$ может быть разложен как произведение отображений $f_1$ и $f_2$. Данные отображения работают с множествами $V_1$ и

$E_1$, которые являются проекциями множества $G$:$V_1 = p_v(G_1), E_1 = p_e(G_1)$. Условие на операции $s$ и $t$ наглядно на коммутативной диаграмме, по сути оно обозначает следующее: для любого ребра $e$ образы его конца $f_v(t(e))$ и начала $f_v(s(e)$ совпадают с концом и началом образа ребра — $t(f_e(e)$ и $s(f_e(e))$.

Таким образом, в диаграмме 3.3 все стрелки между прямоугольниками являются морфизмами на графах. Более того, конструкция соединения графов может быть рассмотрена как алгебра. Этот алгебраический взгляд на графы и трансформации графов более подробно рассмотрен в [20, 23].

### 3.1.2. Трансформация одиночной склейкой

Как было отмечено, конструкции соединения в алгебраическом подходе являются кодекартовыми квадратами в смысле морфизмов категории графов. С другой стороны, правило $p = (\Lambda \leftarrow \Psi \rightarrow \Phi)$ на рисунке может быть также рассмотрено как частичный морфизм графов $p : \Lambda \rightarrow \Phi$, доменом которого является множество $\mathrm{dom}(p) = \Psi$. Более того, диаграмма $(\Gamma \leftarrow \Delta \rightarrow \Omega)$ может быть рассмотрена как частичный морфизм графов $s : \Gamma \rightarrow \Omega$ с доменом $\mathrm{dom}(s) = \Delta$. Таким образом, получается следующая диаграмма




(3.10)

На данной диаграмме горизонтальные морфизмы являются частичными, а вертикальные — полными морфизмами графов. По сути, диаграмма 3.11 является кодекартовым квадратом в расширенной категории графов, которая состоит из графов и частичных морфизмах на графах и показывает, что трансформации графов могут быть выражены как одиночные кодекартовы квадраты в расширенноей категории графов. Данный подход развивался Раулем [45] и был полностью разработан Лёве [34], итогом их работы является подход однократного вытеснения.

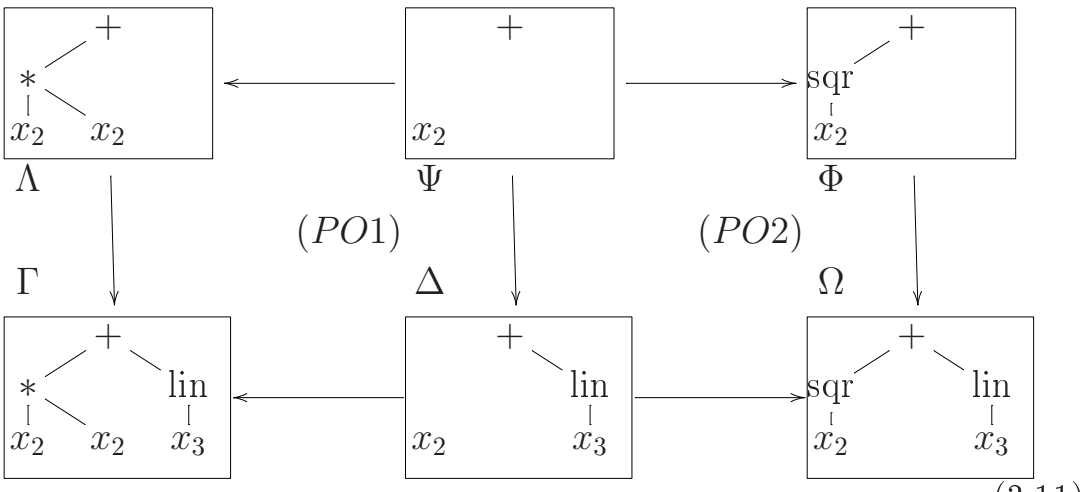
С точки зрения прикладного использования, подход с одиночным вытеснением отличается от подхода с двойным вытеснением в одном главном отношении, которое касается удаления вспомогательных элементов графа в процессе трансформации графа. Если процедура поиска $m : \Lambda \to \Gamma$ не удовлетворяет условию соединения по отношению к правилу $p = (\Lambda \leftarrow \Psi \to \Phi)$. Данное правило не

применимо в подходе с двойным вытеснением. Но оно может быть применимо в подходе с однократным вытеснением, которые позволяет появляться подвешенным ребрам после удаления подграфа $\Lambda \setminus \Psi$ из $\Gamma$. Следует заметить, что подвешенные ребра из $\Gamma$ также удаляются для создания допустимого графа $\Omega$.

Если на диаграмме 3.4 вершина (2) была бы удалена из $\Psi$, то конструкция соединения не удовлетворяла бы подходу с двойным вытеснением. В подходе с однократным вытеснением это значило бы, что вершина (2) не находится в домене $p$, что ведет нас к подвешенному ребру в $\Gamma$ после удаления $\Lambda \setminus \mathrm{dom}(p)$ на диаграмме 3.11. В результате ребро $e$ удялется из $\Omega$.

Более подобное описание и сравнение данных подходов разобрано в [22].

### 3.1.3. Прикладная задача упрощения суперпозиций




(3.11)

При последовательном порождении моделей зачастую оказывается так, что некоторые части модели становятся рудиментарными. Упрощение Соула [50] является вариантом алгебраического упроще-

ния, в котором объектами упрощения являются элементы моделей, параметры которых не влияют на значение функции. Область применения подобных методов ограничена [50], однако они показывают хороший результат на некоторых задачах, например при обнаружении функции. В данном типе задачи восстановления регрессии дисперсия случайной ошибки равна нулю, и выборка генерируется в соответствии с какой либо эталонной функцией $f_0$, которая должна быть обнаружена алгоритмом.

Упрощение эквивалентным решением заключается в сравнении значений моделей, а не структур. Эквивалентность моделей проверяется не по структуре деревьев, соответствующих им, а численно. В таком случае, два выражения, дающие равные значения на области определения независимых переменных модели, считаются равными.

**Определение 28.** Шаблон $\theta$ — гиперсхема, обладающая наименьшей сложностью среди всех гиперсхем, таких что при их взаимном замещении получаемые модели оказываются эквивалентными. Сложность гиперсхемы определяется как сложность суперпозиции при замещении всех символов $\{=\}$ и $\{\#\}$, означающих соответственно произвольную независимую переменную и произвольное поддерево, на константы.

Экспертно выбирается некоторый набор шаблонов $\Theta$. Процедура упрощения состоит из двух шагов:

1. Все поддеревья $\Gamma_j$ в выбранном дереве $\Gamma$ проверяются на эквивалентность шаблонам из $\Theta$ согласно заданным правилам.
2. Если какое-либо поддерево $\Gamma_j$ в дереве эквивалентно дереву из $\Theta$, данное поддерево заменяется соответствующим элементом из $\Theta$.

Процедура повторяется до тех пор, пока после вышеперечисленных итераций дерево Г не останется неизменным. При наличии в множестве порождающих функций коммутативных функций вводится алфавитное упорядочение для ветвей, выходящих из вершины $\gamma_i$ дерева Г, соответствующей коммутативной порождающей функции $g_i$.

Эквивалентное упрощение является альтернативой алгебраическому упрощению, позволяя упрощать некоторые модели за меньшее количество операций.

Рассмотрим сложность алгоритма, упрощающего поддерево высоты $l$ с вершинами арности не более $m$. Количество вершин в таком дереве ограничивается сверху как $m^l$. Рассмотрим дерево с максимальным количеством вершин — для такого дерева все вершины, кроме листьев, будут иметь арность $m$. Для сравнения всех возможных поддеревьев с шаблонами из $\Theta$ необходимо рассмотреть поддеревья любой высоты с корнем в каждой из вершин дерева. Подсчитаем количество поддеревьев всех возможных высот в таком дереве. Обозначим высоту данного дерева $l = \log_m k + 1$. Тогда для вершины, находящейся на расстоянии $x$ от корня количество поддеревьев с корнем в этой вершине составляет не менее, чем $l - x$. Тогда искомое количество поддеревьев:

$$\sum_{x=0}^{l-1}(l-x)m^x = \frac{m(m^l-1)-lm+l}{(m-1)^2}$$

Данное выражение пропорционально $m^l$, то есть количеству элементов в дереве, все вершины которого (кроме листьев) имеют максимальное число потомков. Сложность алгоритма, упрощающего дерево, состоящее из $k$ вершин, оказывается порядка не менее чем $k$. В случае, если алгоритм проверки правил эквивалентности имеет

значительную сложность, подсчет значений оценок зависимых переменных $\hat{y}$ на множестве независимых переменных $x \in D$ и сравнение этих значений с получаемыми при использовании шаблонов $\Theta$ имеет значительно меньшую сложность. Данный метод может применяться в случае, если независимые и зависимые переменные принимают ограниченное число значений. Для такого поддерева вне зависимости от количества элементов $k$ в нем область определения соответствующей функции содержит $2^t$ точек, где $t$ — количество независимых переменных, являющихся листьями данного поддерева. При небольших $t$ число $2^t$ не превосходит $k$ и в таком случае алгоритм сравнения по значениям оказывается менее сложным, чем алгоритм сравнения структур поддеревьев с шаблонами.

Важным частным случаем использования алгоритма упрощения по значениям является случай равенства функций на области определения независимых переменных при необязательном равенстве вне этой области. Для решения прикладной задачи функции, дающие равные значения на области определения, будут равны, и подобная замена будет правомощна.

# Глава 4
# Вычислительный эксперимент

## 4.1. Задача построения моделей ценообразования

В 1973 году была открыта CBOE — Chicago Board Options Exchange — биржа по заключению стандартных контрактов с опционами. В этом же году были опубликованы работы Блэка, Шоулза и Мертона, определившие методологию оценки опционов [29] в финансовых расчетах.

К 2009 году опционы [27] и другие производные инструменты оказывается более привлекательными для инвесторов, нежели акции традиционные бумаги рынка. Ежегодно заключается более миллиарда контрактов, CBOE занимает более 30% торгов в США.

Среди производных бумаг как инструментов финансовой инженерии наиболее заметное место занимают опционы и фьючерсные контракты. Данные инструменты имеют очень высокий риск, но в то же время они и их различные комбинации могут быть использованы не только с целью получения спекулятивного дохода, но и как средство защиты от резкого значительного изменения цен.

**Фьючерс** или **Фьючерсный контракт** — контракт, предполагающий *обязательство* купить или продать определенную ценность (к примеру акцию, облигацию, валюту) в установленный период или момент времени на заранее оговариваемых условиях.

Основным преимуществом использования фьючерса является отсутствие необходимости обладания достаточной суммой для покупки акции/валюты на момент заключения сделки. Таким образом, становится возможной страховка рисков роста или падения це-

ны важной макроэкономической величины. Однако, во фьючерсном контракте обе стороны несут одинаковую ответственность за исполнение контракта. В связи с этим распостранение получили контракты другого типа, где ответственность лежит только на одном участнике.

**Опцион** — ценная бумага, выпускаемая фирмами, корпорациями, банками и другими финансовыми институтами и дающая покупателю *право* купить или продать определенную ценность (к примеру акцию, облигацию, валюту) в установленный период или момент времени на заранее оговариваемых условиях.

Существует два основных вида опционов. *Опцион покупателя, опцион на покупку* или *опцион "колл"* дает его владельцу право купить базовый акктив в определенный день по определенной цене. *Опцион продавца, опцион на продажу* или *опцион "пут"* дает его владельцу право продать базовый акктив в определенный день по определенной цене. Дата, оговоренная в контракте, называется датой исполненеия контракта, время, оставшееся до даты исполнения обозначается $T$. Цена $K$, оговоренная в контракте, называется *ценой исполнения* опциона, или *страйк*.

Также важной величиной, определяющим цену опциона, является волатильность цены акции. Волатильность цены акции $\sigma$ представляет собой меру неопределенности её доходности. Волатильность акции можно также определить как стандартное отклонение доходности акции за один год, когда доходность рассчитывается непрерывно.

Наиболее распостраненной моделью оценки цены опционов, использующейся на рынке, стала модель Блэка-Шоулза [28]. Формула, по которой рассчитывается цена опциона в данной формуле, яв-

ляется результатом решения стохастического диффиренциального уравнения. При его постановке используются следующие предположения:

1. Цена акции $S$ подчиняется стохастическому процессу

$$dS = \mu S dt + \sigma S dz,$$

   где $\mu$ и $\sigma$ — константы, а $z$ - винеровский процесс.
2. Разрешается продавать ценные бумаги без покрытия и использовать вырученные суммы в полном объеме.
3. Отсутствуют комиссии и налоги. Объем лота не учитывается.
4. На протяжении срока действия контракта отсутствуют дивиденды.
5. Арбитражные возможнсоти, свободные от риска, отсутствуют.
6. Торговля происходит непрерывно.
7. Безрисковая процентная ставка $r$ является постоянной для всех сроков погашения.

Решением данного уравнения является формула оценки цена опциона по Блэку-Шоулзу:

$$c = S_0 N(d_1) - K e^{-rT} N(d_2),$$

где

$$d_1 = \frac{ln(S_0/K) + (r^2 + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$d_2 = \frac{ln(S_0/K) + (r^2 - \sigma^2/2)T}{\sigma\sqrt{T}}$$

Здесь $N(\cdot)$ — интегральная функция нормального распределения.

Следует заметить, что один из параметров, определяющих цену опциона — волатильность, не является измеряемой напрямую величиной. На практике используется термин *подразумеваемая волатильность*, т.е. волатильность, величина которой обусловлена ценами акций, существующих на рынке.

Практика использования модели Блэка-Шоулза показала, что не все предположения, заложенные в модель, являются истинными, так как подразумеваемая волатильность, заложенная в цены опционов, может зависеть от цены исполнения опциона и времени, оставшегося до истечения срока опциона.

График подразумеваемой волатильности опциона, зависящей от цены акции, называется *улыбкой волатильности* [19]. Поверхность, образованная волатильностями опционов, зависящих от времени до исполнения опциона и цены акции, называется *поверхностью волатильности* [46].

### 4.1.1. Правила построения поверхностей волатильности

В [17] Халлом, Дэглишем и Суо предложено математическое описание моделей волатильности. Предложены правила построения функции $\sigma(K,t)$. Правила используются для моделирования волатильности в случае относительно малого количества опционов.

Функция $\sigma(K,t)$ волатильности от цены исполнения представляется в виде суммы четной и нечетной функции, каждая от одного аргумента. Четная функция экспертами называется «smile», нечетная — «skew».

Правило «Sticky strike» предполагает, что $\sigma(K,t)$ не зависит цены исполнения $K$. Это предположение требуется для того, чтобы

чувствительность цены опциона к $K$ была равна

$$\Delta = \frac{\partial C}{\partial K},$$

где $\Delta$ является чувствительностью по отношению к цене базового инструмента. $\Delta$ для инструмента — это частная производная цены опциона $C$ по отношению к цене базового инструмента $S$, $\Delta = \frac{\partial}{\partial S}$.

Здесь для целей вычисления частная производная цены опциона, $c$, полагается функцией цены базового инструмента $S$, волатильности $\sigma(K,t)$ и времени $t$. Это предположение позволяет пользоваться формулой Блэка–Шоулза для вычисления $\Delta$ с волатильностью, равной предполагаемой волатильности опциона.

В модели, основанной на правиле «sticky strike», $\sigma$ является функцией только от цены исполнения $K$ и времени до исполнения $t$ и не зависит от цены базового инструмента $S$.

Правило «square root of time» предполагает связь между волатильностями опционов различных цен исполнения и времен до исполнения. В работе оно будет использоваться в формулировке

$$\sigma(K,t) = \Phi\frac{\ln(K)}{\sqrt{t}}, \tag{4.1}$$

таким образом волатильность является функцией одного параметра, являющегося функцией от переменных $K$ и $T$.

### 4.1.2. Проверка отсутствия арбитража в моделях

Важным принципом построения моделей финансового рынка является отсутствие возможности извлечения арбитражной прибыли в случае, если рынок соответствует модельным ожиданиям. На практике это означает, что предсказанные моделью цены не должны

давать возможность извлекать мгновенную прибыль для игроков. При моделировании поверхности волатильности возникает большое число сложным образом связанных друг с другом цен различных опционов. Для данного набора цен необходимо найти достаточный критерий отсутствия арбитража [14].

Для смоделированной поверхности волатильности можно с помощью формулы Блэка-Шоулза получить цены соответствующих опционов $C_{ij}$, которым соответствуют цены исполнения $K_i, i = 1, \ldots, N$ и время до исполнения $t_j, j = 1m \ldots, M$. Предполагается, что цены исполнения упорядочены по возрастанию. При этом цены опционов стремятся к нулю с увеличением цены исполнения. Также предполагается, что безрисковые ставки и дивиденды равны нулю для всех периодов до исполнения опциона. Дополнительно вводится цена исполнения $K_0 \equiv 0$ — цена опцион с такой ценой исполнения равна текущей цене базового актива $S_0$. Также вводится дополнительное значение срока до исполнения $T_0 = 0$, для которого цены опционов равны $(S_0 - K_i)^+, i = 1, \ldots, N$. Таким образом строится расширенная матрица значений цен опционов $C_{ij}$.

Для любого $j > 0$ вводится величина $Q$:

$$Q_{i,j} = \frac{C_{i-1,j} - C_{i,j}}{K_i - K_{i-1}},\ i > 0,\ \ Q_{0,j} \equiv 1. \tag{4.2}$$

Для $i > 0, Q_{i,j}$ является ценой вертикального спреда, состоящего из покупки $1/(K_i - K_{i-1})$ опционов колл цены исполнения $K_{i-1}$ и продажи $1/(K_i - K_{i-1})$ опционов пут с ценой исполнения $K_i$. График, показывающий финансовый результат данной позиции для различных цен базового актива, должен находиться от нуля до единицы для всех значений $S_0$ [14]. Таким образом, одним из условий отсутствия арбитража является $Q_{i,j} \in [0, 1]$ для всех $i, j$.

Далее, для каждого $j > 0$ определяется следующее соотношение:

$$BS_{i,j} \equiv C_{i-1,j} - \frac{K_{i+1} - K_{i-1}}{K_{i+1} - K_i} C_{i,j} + \frac{K_i - K_{i-1}}{K_{i+1} - K_i} C_{i+1,j},\ i > 0. \quad (4.3)$$

Для каждого $i > 0$, $BS_{i,j}$ является ценой спреда-бабочки, состоящего из одного купленного опциона колл цены исполнения $K_{i-1}$, $\frac{K_{i+1}-K_{i-1}}{K_{i+1}-K_i}$ проданных опционов колл с ценой исполнения $K_i$ и $\frac{K_i-K_{i-1}}{K_{i+1}-K_i}$ купленных опционов цены исполнения $K_{i+1}$. Такой комплект активов не должен стоить меньше нуля и таким образом, второе условие отсутствия арбитража

$$C_{i-1,j} - \frac{K_{i+1} - K_{i-1}}{K_{i+1} - K_i} C_{i,j} + \frac{K_i - K_{i-1}}{K_{i+1} - K_i} C_{i+1,j} \geq 0.$$

Данное условие можно переформулировать в виде

$$C_{i-1,j} - C_{i,j} \geq \frac{K_i - K_{i-1}}{K_{i+1} - K_i}(C_{i,j} - C_{i+1,j}). \quad (4.4)$$

Из условий 4.2 и 4.4 следует дополнительное условие на $Q_{i,j}$:

$$Q_{i,j} \geq Q_{i+1,j},\ i, j \geq 0. \quad (4.5)$$

Отсюда следует, что выполнение условия 4.5 гарантирует отсутствие арбитражной прибыли.

## 4.2. Исходные данные

Цель вычислительного эксперимента — сравнение моделей, полученных с помощью алгоритмов, предложенных в данной работе и моделей начального приближения, предложенных экспертами с учетом упомянутых правил.

Для анализа выбран исторические данные торгов опционом Brent Crude Oil. Срок действия опциона — полгода, с 02.01.2001

по 26.06.2001. Тип опциона — put (право на продажу базового инструмента), символ CLG01. Базовый инструмент — нефть, символ NYM. Использовались ежедневные цены закрытия опциона и базового инструмента. Сетка цен исполнения опциона $\mathcal{K} = \{18.0, 19.0, 19.5, 20.0, 20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 24.0, 24.5, 25.0, 25.5, 26.0, 26.5, 27.0, 27.5, 28.0, 28.5\}$.

Данный выбор обусловлен тем, что объемы торговли инструментом велики. Инструмент имеет низкую волатильность, вследствие чего среди точек $\sigma^{\text{imp}}(K,t)$ нет выбросов. В исходных данных имеются пропуски, так как опционы, с ценами, далекими от цен базового инструмента не торговались непосредственно после выпуска опционов. Эти данные исключены из регрессионной выборки. Пропуски заполнялись в предположении, что цена изменяется в соответствии с безрисковой ставкой доходности: $C(t^*) = C(t_0)e^{B(t^*-t_0)}$, где $t^*$ обозначает время, соответствующее пропущенным данным.

Регрессионная выборка $\{(\mathbf{x}_n, y_n)\} = \{(\langle K_n, t_n\rangle, \sigma_n)\}$ была построена с помощью исходных данных — исторических цен опциона $C_{K,t}$ и базового инструмента $P_t$, где $K \in \mathcal{K}, t \in T$, следующим образом. Для каждого значения $K \in \mathcal{K}$ и $t \in T$ вычисляется значение предполагаемой волатильности как аргумент минимума

$$\sigma^{\text{imp}}_{K,t} = \arg\min_{\sigma\in[0,1.5]}(C_{K,t} - C(\sigma, P_t, B, K, t)), \tag{4.6}$$

где справедливая цена опциона $C$ вычислена по формуле Блэка-Шоулза. Время $t$ выраженное в годах до момента исполнения опциона рассчитывается по формуле $t = \tau/365$, где $\tau$ — число дней, оставшихся до исполнения опциона. Длина истории составляет 112 отсчетов времени. Для индексации выборки задана произвольная биекция $t, K \mapsto n$. Безрисковая ставка доходности $B = 0.075$, что со-

ответствует исторической безрисковой ставке за рассматриваемый период. При отыскании минимума введено ограничение на значение волатильности, $\sigma \leqslant 1.5$; ни в одной точке исходных данных она не принимает большее значение.

Исходное множество $D$, является результатом отображения декартова произведения сетки цен исполнения $K$ и времен $t$ в пространство значений цен опционов $C_{Kt}$. Полученная матрица $C(K,t)$ имеет размерность $\tilde{T} \times \tilde{K}$. При вычислительном эксперименте берется исходное множество цен исполнения $K_j(j = 1..6), |K_j| = \tilde{K}$ и времен $t_i, |t_i| = \tilde{T}$ в днях до исполнения. Для каждой пары $\{K,t\}$ имеется величина $C_{Kt}$, соответствующая цене на опцион колл в момент $t$ и с ценой исполнения $K$. Также имеется набор цен на подлежащий инструмент $S_t$, соответсвующий временам $t_i$. Таким образом, исходное множество задачи имеет вид: $D = \{(P_{Kt}, S_t)|K = 1, .., 6; t = (-91, .., 0)\}$. Из времени получаем величину maturity$=\frac{-t}{365}$, которая выражает время в годах, остающееся до исполнения опцона.

Далее из этой матрицы применением поэлементно функции (4.6), в которой справедливая цена опциона рассчитывается по формуле Блэка-Шоулза, получается матрица $\sigma_{ij}(\tilde{T} \times \tilde{K})$, в которой находятся значения вычисленной волатильности для соответствующих времен до исполнения и цен исполнения. Таким образом, регрессионная модель имеет вид $\sigma = f_i(K,t) + \nu$. При этом безрисковая ставка доходности была принята за $7,5\%$, что соответствует реальной безрисковой ставке на тот период. Сама волатильность ограничивается 1,5, так как ни на одной точке исходных данных она не оказывается больше.

На вход алгоритм получает исходное множество $D$, заданное в виде матрицы $91 \times 8$, что соответствует всем временным точкам по

строкам, столбцы со второго по 7й соответствуют ценам на опционы, первый столбец соответсвует времени до исполнения, и последний — цене на подлежащий инструмент,

Далее из нее применением функции, обратной к формуле Блэка-Шоулза, получается матрица V(91×6), в которой находятся значения вычисленной волатильности для соответствующих времен до исполнения и цен исполнения.

## 4.3. Модели начального приближения

В качестве моделей начального приближения был взят набор из четырех моделей: $f_1$ — базовая модель, предлагаемая биржей ММВБ-РТС, предложенная экспертами, и ее вариации

a) сумма гауссиана и линейной функции от цены исполнения, умноженная на обратный корень времени до исполнения — модель, соответствующая экспертной.

$$\sigma = \sigma(\mathbf{w}) = w_1 + w_2(1 - \exp(-w_3 x^2)) + \frac{w_4 \arctan(w_5 x)}{w_5}, \quad (4.7)$$

где $x = \frac{\ln K - \ln C(t)}{\sqrt{t}}$

b) сумма параболической функции от цены исполнения и гауссиана от цены исполнения, умноженного на обратный корень времени до исполнения,

$$f_2 = (w_1 + w_2 K + w_3 K^2 + w_4 \exp(-w_5 K^2))\sqrt{w_6 t};$$

b) квадратичная поверхность по времени и цене исполнения,

$$f_3 = w_1 + w_2 K + w_3 t + w_4 K t + w_5 K^2 + w_6 t^2;$$

c) сумма гауссиана и линейной функции от цены исполнения, умноженная на кубическую функцию времени до исполнения,

$$f_4 = (w_1 + w_2 K + w_3 \exp(-w_4 K^2))(w_5 t^3 + w_6 t^2 + w_7 t + w_8).$$

## 4.4. Иллюстративный вычислительный эксперимент

Модели, используемые в практике работы на фондовом рынке, часто обладают высокой сложностью и большим числом экспертных поправок, например в них может учитываться влияние дивидендных выплат или риск дефолта по облигациям. В связи с этим наравне с экспериментом по построению адекватной рыночной модели, напрмер аналога 4.7, также полезен эксперимент по сравнению намеренно упрощенных моделей различных классов для оценки качества моделей, порождаемых при применении алгорима последовательного порождения, с моделями, построенными стандартными способами.

Для решения задачи был организован поиск модели среди классов линейных, обобщенно-линейных моделей, нейронных сетей и существенно-нелинейных моделей. Сложность моделей ограничивалась числом 80 (кроме нейронных сетей). Полученные модели при этом сравнивались с созаднными ранее [37] для решения схожей задачи. Для улучшения работы алгоритма для каждого класса моделей использовались следующие спецификации.

1. Для класса линейных моделей было запрещено использование в качестве элемента суперпозиции одной входной переменной более одного раза для получения корректной оценки параметров моделей.
2. Для нейронных сетей количество $S$-вершин было ограничено

числом 10, использовалась оценка параметров нейронной сети с помощью метода обратного распространения ошибки.

3. Для обобщенно-линейных моделей использовались полиномиальные функции, функции $\frac{1}{x}$, $\frac{1}{\sqrt{x}}$, $e^x$ и $\ln x$ как наиболее часто встречающиеся в работах, посвященных финансовой математике.
4. Тот же набор функций использовался для существенно нелинейных моделей, при этом для упрощения алгоритма поиска модель имела вид суммы произведения двух других моделей и константы.

Порождаемые модели настраивались с помощью алгоритма Левенберга-Марквардта, после чего для каждого класса моделей была выбрана модель с наилучшим значением $SSE$.

## 4.5. Результаты иллюстративного эксперимента

Результаты вычислительного эксперимента и графики, отображающие наилучшие модели, представлены в таблице 4.1. Столбцы означают следующее:

1. $C(f)$ — сложность дерева, соответствующего модели
2. $MSE_{\text{learn}}$ — средняя ошибка на обучающей выборке
3. $MSE_{\text{test}}$ — средняя ошибка на контрольной выборке
4. $R^2_{\text{adj}}$ — модифицированный коэффициент корелляции модели
5. $AIC$ — значение критерия Акаике [13] для модели

Из 4.1 можно видеть, что линейная модель выглядит на фоне остальных недостаточно хорошо приближающей данные (плохие значения AIC, SSE) и дает низкий коэффициент детерминации $R^2_{\text{adj}}$.

Таблица 4.1

Результаты вычислительного эксперимента

| Класс моделей | Число параметров | $C(f)$ | $MSE_{\text{learn}}$ | $MSE_{\text{test}}$ | $R^2_{\text{adj}}$ | AIC |
|---|---|---|---|---|---|---|
| Линейная | 3 | 19 | 46.98 | 51.53 | 63% | 192.09 |
| Нейронная сеть | 10 | 81 | 20.43 | 25.21 | 89% | 178.45 |
| Обобщенно-линейная | 6 | 48 | 27.06 | 30.11 | 78% | 133.43 |
| Нелинейная | 4 | 50 | 11.28 | 13.76 | 90% | 69.27 |
| Экспертная модель | 5 | 66 | 27.78 | 30.85 | 77% | 137.95 |

Следует заметить, что при более низкой среднеквадратичной ошибке нейронная сеть оказывается хуже обобщенно линейной модели из-за большого количества параметров, содержащихся в ней. Наилучшим образом показывает себя нелинейная модель. При этом количество настраиваемых параметров в ней меньше, чем в обобщенно-линейной модели. Модели, полученные в ходе вычислительного эксперимента, оказываются предпочтительнее моделей, которые были предложены экспертами ранее [37] — нелинейные модели дают лучшую оценку волатильности при меньшем количестве настраиваемых параметров, чем полиномиальные модели значительно большей сложности.

На графиках, представленных на рис. 4.1 справа, по горизонтальным осям отложены значения относительной цены исполнения $M$ и

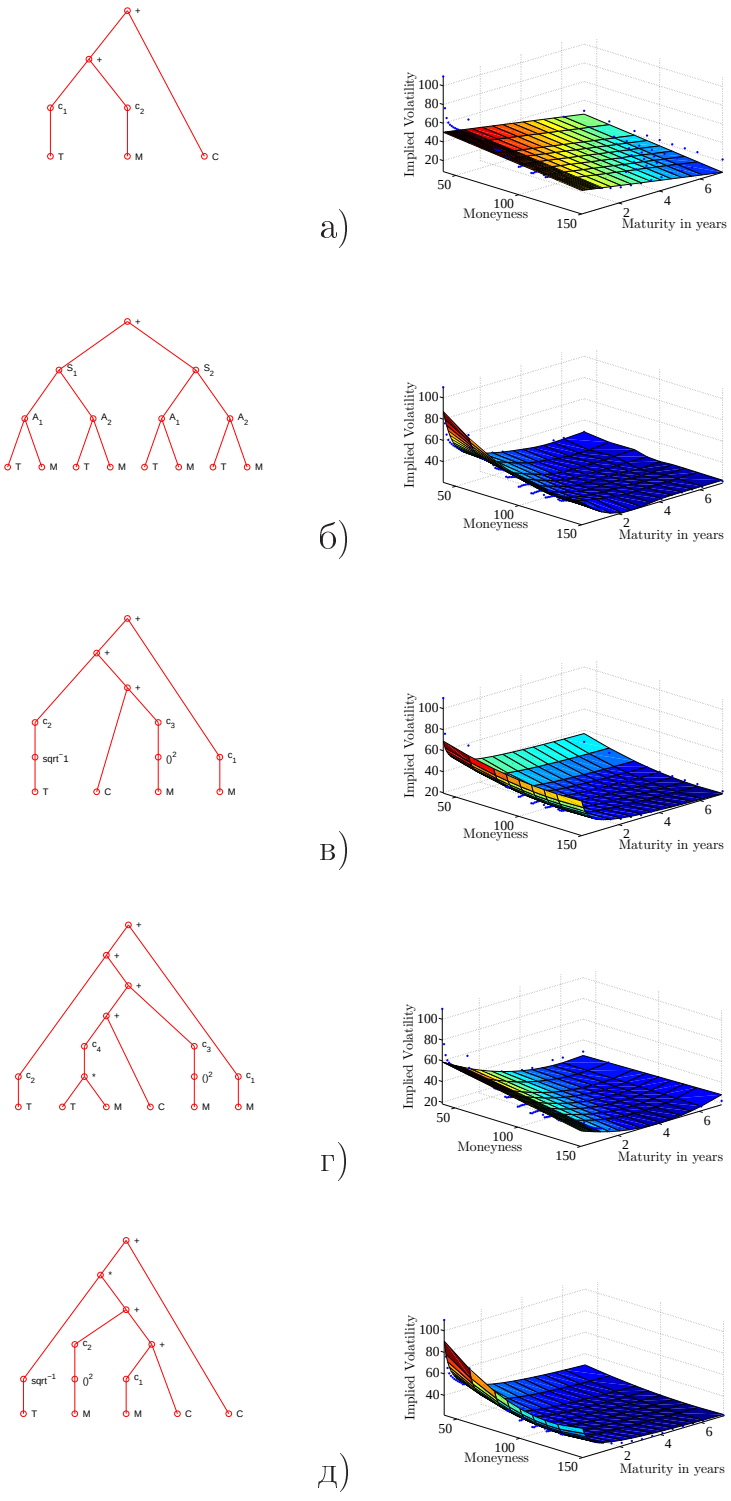



Рис. 4.1. Примеры структур различных классов моделей и восстановленных поверхностей волатильности.

времени до исполнения опциона $t$ в годах. По вертикальным осям отложены предполагаемые волатильности $\sigma_{\text{implied}}$, соответствующие реально торгуемому опциону с параметрами $M$ и $t$. Соответствие рисунков моделям:

а) линейная модель: $\sigma_{\text{imp}} = c_1 M + c_2 T + C$,

б) нейронная сеть: $\sigma_{\text{imp}} = \sum_{i=1}^{10} S_i(\sum_j A_j(M,T)$,

в) обобщенно-линейная модель: $\sigma_{\text{imp}} = c_1 M + c_2 M^2 + c_3 \frac{1}{\sqrt{T}} + C$,

г) экспертная модель: $\sigma_{\text{imp}} = c_1 M + c_2 M^2 + c_3 T + c_4 MT + C$,

д) существенно-нелинейная модель: $\sigma_{\text{imp}} = \frac{c_1 M + c_2 M^2 + C_1}{\sqrt{T}} + C_2$,

На 4.1 слева изображены деревья $\Gamma_i$, отображающие данные модели при представлении моделей в виде деревьев.

## 4.6. Параметры алгоритма модификации суперпозиций

Для построения рабочей модели был использован набор порождающих функций, описанный в работе [6]. При каждой итерации на основе множества моделей-претендентов порождалось 40 новых моделей, из них 20 модифицировалось. Для следующей итерации выбиралось 20 лучших моделей. Алгоритм останавливался при достижении функции ошибки $E_D(\mathbf{w}|f, D) < 0.01$ или после 5000 генераций нового набора моделей-претендентов.

## 4.7. Результаты вычислительного эксперимента

На рис. 1 показаны a) исходные данные, b) базовая модель PTC, c) модель, имеющая наименьшее значение функции ошибки $E_D(|f_i, D)$ среди всех порожденных моделей.

Полученная модель имеет вид

$$f_{\text{best}} = (w_1 + w_2 K + w_3 \exp(-w_4 K^2))(w_5 t^3 + w_6 t^2 + w_7 t + w_8).$$

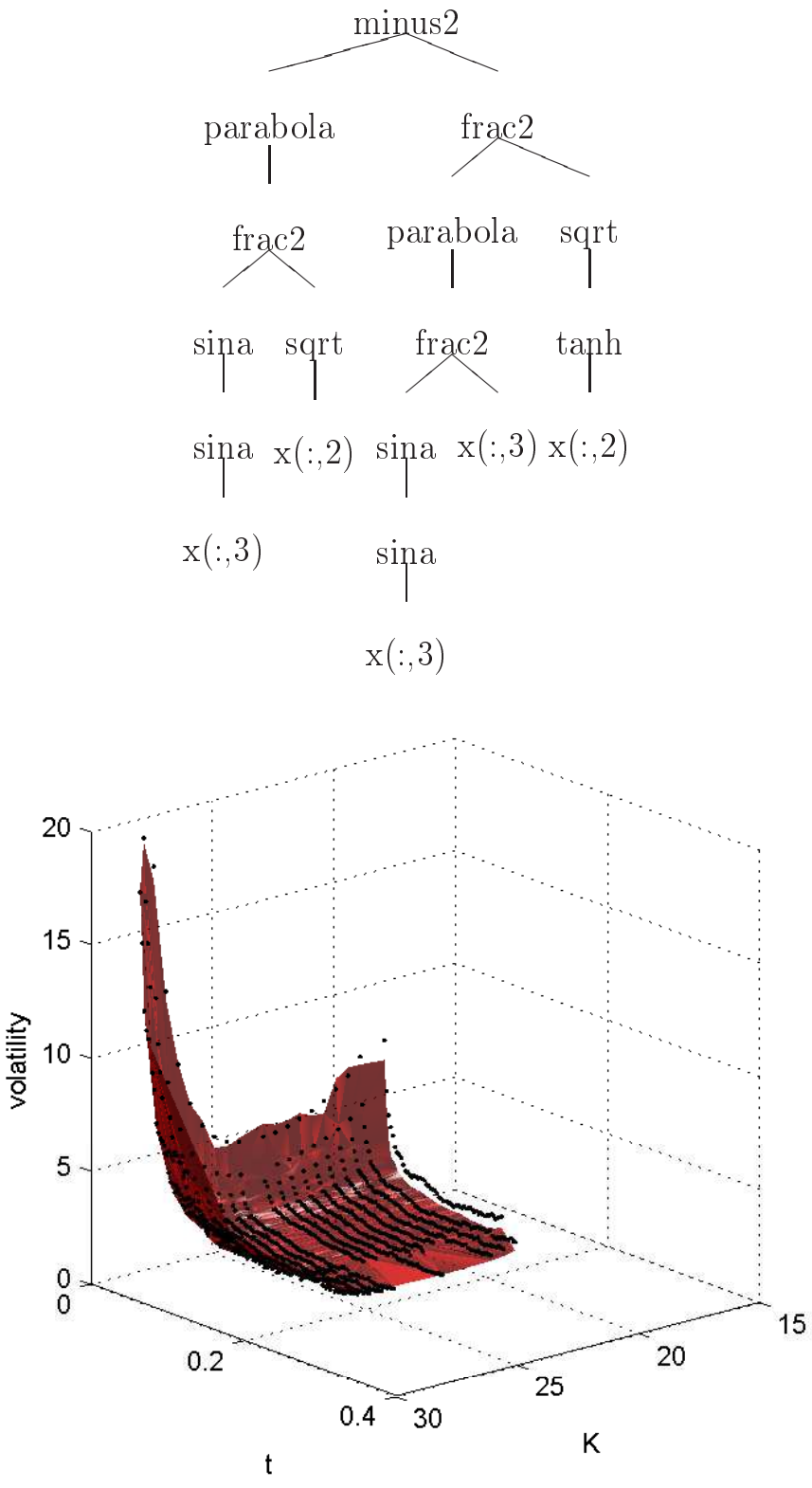
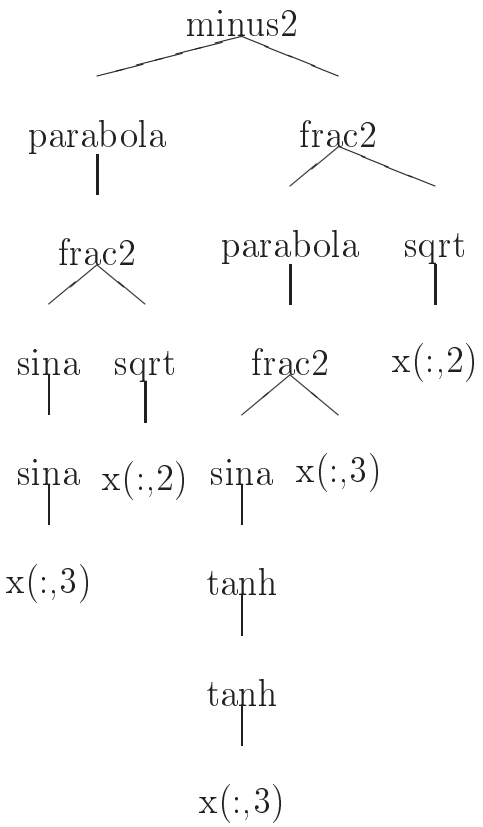



Рис. 4.2. Модель $f_1$.



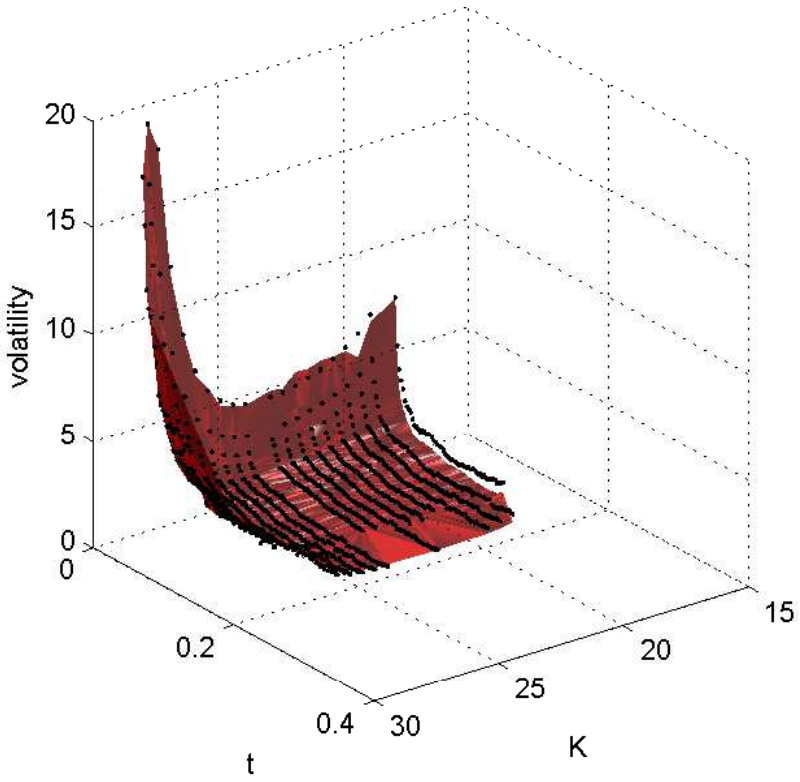
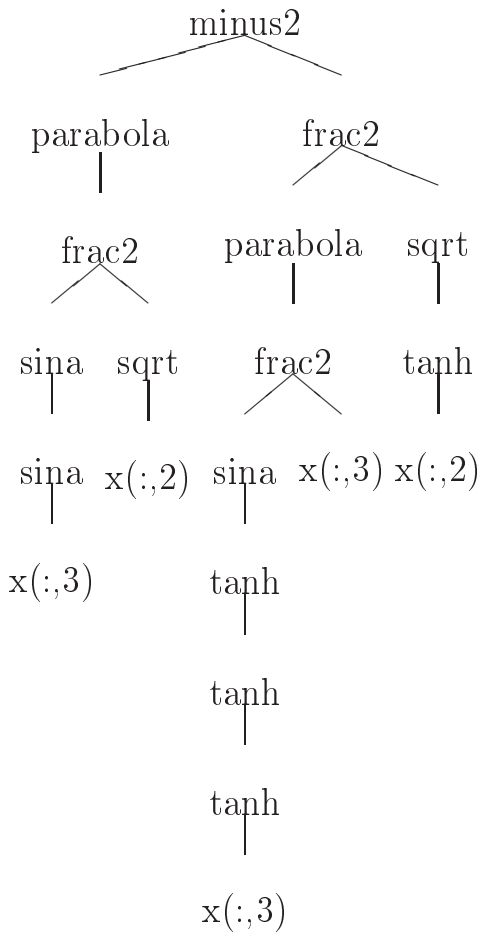




Рис. 4.3. Модель $f_2$.



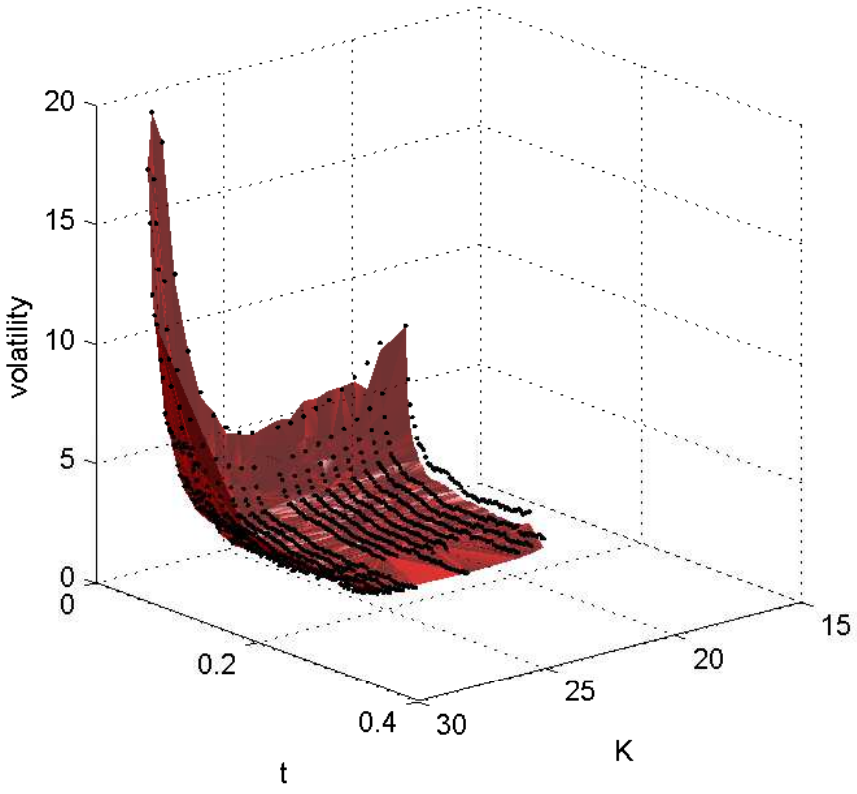




Рис. 4.4. Модель $f_3$.

Таблица 4.2

Порожденные модели с наименьшей ошибкой SSE.

| Модель | Ошибка SSE | Число параметров | max ошибка на выборке | число порождающих ф-ий в суперпозици |
|---|---|---|---|---|
| $f_1$ | 0.0590 | 15 | 1.5989 | 15 |
| $f_2$ | 0.0593 | 12 | 1.8325 | 14 |
| $f_3$ | 0.0596 | 15 | 1.6055 | 15 |
| $f_4$ | 0.0601 | 15 | 1.5754 | 15 |
| $f_5$ | 0.0602 | 14 | 1.5532 | 15 |
| $f_6$ | 0.0603 | 14 | 1.4684 | 15 |
| $f_7$ | 0.0603 | 15 | 1.4324 | 14 |
| $f_8$ | 0.0603 | 15 | 1.4409 | 15 |
| $f_9$ | 0.0603 | 14 | 1.5203 | 15 |
| $f_{10}$ | 0.0604 | 14 | 1.7636 | 15 |

Даже при малом числе итераций в множестве моделей-претендентов остаются в основном те модели, в которых волатильность зависит от времени как обратный квадратный корень (4.1).

Структура выбираемых моделей имеет вид произведения функции от времени на функцию от цены исполнения, что говорит о независимости профиля волатильности от времени.

Полученные модели не обладают высокой сложностью, что позволяет экспертам легко их интерпретировать. Ниже представлены доставляющие наименьшую ошибку модели, созданные программой.

## 4.8. Обсуждение результатов

Для создания адекватной модели измеряемых данных используются экспертно-заданные порождающие функции и набор правил порождения. Модель задается в виде суперпозиции порождающих функций. Правила порождения определяют допустимость суперпозиции и исключают порождение изоморфных моделей.

Развиты существующие методы автоматического порождения моделей. В частности, при порождении моделей учитываются экспертные требования к виду моделей, результирующие модели соответствуют экспертным предпочтениям. Предлагаются новые методы поиска изоморфных суперпозиций, основанные на быстром поиске изоморфных подграфов и подстановке подграфов по правилам.

Вычислительные эксперименты по порождению моделей подтверждают гипотезу о малости изменчивости самой волатильности, высказанную в [9], и гипотезу о обратной зависимости времени и волатильности, высказанную в [17].

Следующие методы и походы, предлагаемые автором, являются новыми: метод порождения моделей по экспертно-заданному набору порождающий функций; метод порождения допустимых суперпозиций с учетом областей определения и областей допустимых значений; метод порождения суперпозиций ограниченной сложности.

## Заключение

В работе предложена процедура направленного порождения модели. Для постановки задачи и для описания процедуры направленного порождения выбрана адекватная алгебраическая структура. Описаны условия существования решений, получаемых в резуль-

тате процедуры порождения, доказаны необходимые теоремы.

Разработан модфицированный алгоритм направленного порождения моделей. Разработаны новые алгоритмы вычисления структурной сложности порождаемых суперпозиций и алгоритмы вычисления расстояния между порождаемыми суперпозициями.

Разработан метод последовательного направленного порождения суперпозиций, исследованы свойства порождаемых суперпозиций.

Введено понятие изоморфных суперпозиций, разработан метод их обнаружения. Разработан алгоритм поиска изоморфных подграфов, соответствующих порожденным суперпозициям.

Разработан новый метод порождения экспертно-интерпретируемых моделей. Создана базовая библиотека правил порождения экспертно-интерпретируемых моделей.

## Список иллюстраций

# Список таблиц

## Список обозначений

Список обозначений

Матрицы обозначены заглавными буквами, векторы — полужирными прописными буквами, множества (как правило) — каллиграфическими буквами.

- $\mathbb{R}$ — множество действительных чисел.
- $\mathbb{N}$ — множество натуральных чисел.
- $\mathfrak{D}$ — выборка, набор исходных данных $\mathfrak{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, $\mathbf{x} \in \mathbb{R}^m$.
- $\mathbf{w}$ — множество векторов параметров моделей, $\mathbf{w} \in \mathbb{R}^m$.
- $X$ — множество свободных переменных.
- $Y$ — множество зависимых переменных.
- $f(\mathbf{w}, \mathbf{x})$ — модель, параметрическое семейство отображений.
- $S$ — внешний критерий качества, квадратичная ошибка.
- $G$ — множество порождающих функций $g(\mathbf{w}, \mathbf{x})$.
- $\mathrm{id}(\mathbf{x})$ — порождающая функция id, значение которой равно её аргументу.
- const — порождающая функция, значение которой равно константе.
- $\mathfrak{F}$ — категория суперпозиций над множество исходных данных.
- $\Gamma_f$ — дерево, эквивалентное модели $f$.
- $r_{ij}$ — расстояние между деревьями $\Gamma_i$ и $\Gamma_j$.
- $C(f)$ — сложность суперпозиции.
- $\mathrm{nl}(f)$ — нелинейность суперпозиции.
- $\mathfrak{F}$ — алгоритм построения всех допустимый моделей.
- $\mathfrak{G}$ — алгоритм последовательного порождения моделей.

- $\mathfrak{r}$ — операция замены поддерева.
- $A(d, i, \Gamma)$ — арность вершины.
- $T(d, i, \Gamma)$ — тип вершины, классификация порождающих функций на функции из $G$ и свободные переменные.
- $S(d, i, \Gamma)$ — количество элементов в поддереве дерева $\Gamma$ с корнем в вершине $(d, i)$.
- $p(d, i|\Gamma)$ — плотность вероятности распределения выбора вершины в дереве.
- $\{=\}$ — произвольный символ, который может быть любой функцией или свободной переменной.
- $\{\#\}$ — любое допустимое поддерево.
- $C(d, i, \Gamma_1, \Gamma_2)$ — функция проверки равенства структуры поддеревьев.
- $p_{x0}$ — вероятность проведения операции замены поддерева.
- $p(H, t)$ — вероятность выбора вершины из схемы $H$, суммы проходят по всем деревьям из набора.
- $NC(h_1, h_2)$ — количество вершин с равной структурой родительских вершин.
- $L(H, i)$ — гиперсхема, получаемая из $H$ заменой всех вершин от корня до вершины $i$ вершинами типа $=$, а всех поддеревьев, выходящих из этих вершин — $\#$.
- $U(H, i)$ — гиперсхема, получаемая из $H$ заменой поддеревьев ниже точки $i$ вершинами типа $\#$.
- $L$ — заменяемый подграф, удаляется из графа $G$ в процессе упрощения.
- $R$ — замещающий подграф, подставляется в граф $G$ в процессе упрощения.

- $p = (\Lambda, \Psi, \Phi)$ — правило трансформации графа. Здесь $\Lambda$ и $\Phi$ являются заменяемым и замещающим подграфами и граф $\Psi$ является их общей частью.
- $\mathfrak{m}$ — отображение из $\Lambda$ в $\Gamma$, ставящая в соответствие заменяемому графу эквивалентный ему подграф.
- $\Gamma \overset{p,\mathfrak{m}}{\rightarrow} \Omega$ — трансформация графа $\Gamma$ в граф $\Omega$ с помощью правила $p$ и процедуры поиска $\mathfrak{m}$.
- $\theta \in \Theta$ — шаблон, гиперсхема, обладающая наименьшей сложностью среди всех гиперсхем, таких что при их взаимном замещении получаемые модели оказываются эквивалентными.
- $C(K, t)$ — цена опциона с ценой исполнения $K$ и сроком до исполнения $t$.
- $\sigma(K, t)$ — волатильность опциона с ценой исполнения $K$ и сроком до исполнения $t$.

## Литература


1. Рудой Г. И., Стрижов В. В. Алгоритмы индуктивного порождения суперпозиций для аппроксимации измеряемых данных // Информатика и ее применения. — 2013. — Т. 1. — С. 17–26.
2. Краснощеков П.С., Петров А.А. Принципы построения моделей. — Изд-во Фазис, 2000.
3. Колмогоров А. Н. Интерполирование и экстраполирование стационарных случайных последовательностей. — Изв. АН СССР., 1941.
4. Стрижов В. В. Поиск регрессионных моделей в индуктивно заданном множестве // Искусственный интеллект. — 2006. — Т. 2. — С. 234–237.
5. Макаров Л. И. Метрические свойства функций расстояний между молекулярными графами // Журнал Структурной Химии. — 2007. — Т. 48. — С. 223–229.
6. Стрижов В. В. Методы индуктивного порождения регрессионных моделей. — М.: ВЦ РАН, 2008. — 54 с.
7. Стрижов В. В. Методы выбора регрессионных моделей. — Москва: ВЦ РАН, 2010.
8. Ивахненко А.Г. Индуктивный метод самоорганизации моделей сложных систем. — Наук. думка, 1982.
9. Ширяев А.Н. Основы стохастической финансовой математики: Факты, модели. No. 1. — Фазис, 2004.
10. Genetic programming: an introduction: on the automatic evolution of computer programs and its applications / Wolfgang Banzhaf,

Frank D. Francone, Robert E. Keller, Peter Nordin. — San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998.

11. Bengio Y. Learning deep architectures for AI // Foundations and Trends in Machine Learning. — 2009. — V. 2, no. 1. — P. 1–127.
12. Bishop C. M. Pattern Recognition and Machine Learning (Information Science and Statistics). — Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
13. Burnham K. P., Anderson D. R. Model selection and multimodel inference: a practical information-theoretic approach. — Springer, 2002.
14. Cawley G. C., Talbot N. L. C. Preventing over-fitting during model selection using bayesian regularisation // JMLR. — 2007. — V. 8. — P. 841–861.
15. Comisky W., Yu J., Koza J. R. Automatic synthesis of a wire antenna using genetic programming // Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference, Las Vegas, Nevada. — Plenum Press, 2000. — P. 18–26.
16. Cun Y. Le, Denker J. S., Solla S. A. Optimal brain damage // Advances in Neural Information Processing Systems. — Morgan Kaufmann, 1990. — P. 598–605.
17. Daglish T., Hull J., Suo W. Volatility surfaces: theory, rules of thumb, and empirical evidence // Quantitative Finance. — 2007. — V. 7, no. 5. — P. 507–524.
18. D'haeseleer P. Context preserving crossover in genetic programming // Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference on Computational Intelligence vol.1. — 1994. — P. 256–261.

19. Dupire B. Pricing with a smile // Risk. — 1994. — V. 7, no. 1. — P. 1–10.
20. Ehrig H. Introduction to the algebraic theory of graph grammars (a survey) // Proceedings of the International Workshop on Graph-Grammars and Their Application to Computer Science and Biology. — UK: Springer-Verlag, 1979. — P. 1–69.
21. Fundamentals of Algebraic Graph Transformation / H. Ehrig, K. Ehrig, U. Prange, G. Taentzer. — illustrated edition edition. — Springer, Berlin, 2006.
22. Handbook of Graph Grammars and Computing by Graph Transformation: Vol. 3: Concurrency, Parallelism, and Distribution / Ed. by H. Ehrig, H.-J. Kreowski, U. Montanari, G. Rozenberg. — River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1999.
23. Ehrig H., Pfender M., Schneider H. J. Graph-grammars: An algebraic approach // Switching and Automata Theory, 1973. SWAT '08. IEEE Conference Record of 14th Annual Symposium on. — 1973. — Oct. — P. 167–180.
24. Hassibi B., Stork D. G., Com S. C. R. Second order derivatives for network pruning: Optimal brain surgeon // Advances in Neural Information Processing Systems 5. — Morgan Kaufmann, 1993. — P. 164–171.
25. Hinton G. E. Learning multiple layers of representation // Trends in Cognitive Sciences. — 2007. — V. 11. — P. 428–434.
26. Holland J. H. Adaptation in Natural and Artificial Systems. — Ann Arbor, MI: University of Michigan Press, 1975.
27. Hull J. C. Options, Futures, and Other Derivatives with Derivagem CD. — Prentice Hall, 2008. — .

28. Jackwerth J. C., Rubinstein M. Recovering probability distributions from option prices // Journal of Finance. — 1996. — V. 51, no. 5. — P. 1611–32.

29. Kendall M.G., of Economics London School, Division Political Science. Research Techniques. The Analysis of Economic Time-series: Prices. — London School of Economics and Political Science, 1953.

30. Analytic programming in the task of evolutionary synthesis of a controller for high order oscillations stabilization of discrete chaotic systems / Z. Kominkova Oplatkova, R. Senkerik, I. Zelinka, M. Pluhacek // Comput. Math. Appl. — 2013. — . — V. 66, no. 2. — P. 177–189.

31. Koza John R. Genetic Programming IV: Routine Human-Competitive Machine Intelligence. — Norwell, MA, USA: Kluwer Academic Publishers, 2003.

32. Koza J.R. Genetic Programming: vol. 1 , On the programming of computers by means of natural selection. Complex Adaptive Systems Series. — Bradford, 1992.

33. Levenberg K. A method for the solution of certain non-linear problems in least squares // Quart. J. Appl. Maths. — 1944. — V. II, no. 2. — P. 164–168.

34. Lowe M., Ehrig H. Algebraic approach to graph transformation based on single pushout derivations / Ed. by Rolf. Muhring. Lecture Notes in Computer Science. — Springer Berlin Heidelberg, 1991. — P. 338–353.

35. MacKay D.J.C. Information Theory, Inference and Learning Algorithms. — Cambridge University Press, 2003.

36. Madala H.R., Ivakhnenko A.G. Inductive learning algorithms for complex systems modeling. — CRC Press, 1994.

37. Forecasting extreme volatility of ftse-100 with model free vftse, carr-wu and generalized extreme value (gev) option implied volatility indices: Economics Discussion Papers / University of Essex, Department of Economics. — Executor: S. M. Markose, Y. Peng, A. Alentorn: 2012.

38. Michell M. An Introduction to Genetic Algorithms. Complex Adaptive Systems Series. — MIT Press, 1998.

39. A new method for simplifying algebraic expressions in genetic programming called equivalent decision simplification / N. Mori, B. McKay, N. X. Hoai et al. // Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living / Ed. by S. Omatu, M. P. Rocha, J. Bravo et al. — V. 5518 of Lecture Notes in Computer Science. — Salamanca, Spain: Springer, 2009. — P. 171–178.

40. Mueller J.-A., Lemke F. Self-organising data mining: An intelligent approach to extract knowledge from data. — Berlin: ScriptSoftware International, 1999.

41. Nordin P., Banzhaf W. Complexity compression and evolution // Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95). — Morgan Kaufmann, 1995. — P. 310–317.

42. Poli R., Langdon W. B. Schema theory for genetic programming with one-point crossover and point mutation // Evolutionary Computation. — 1998. — V. 6, no. 3. — P. 231–252.

43. Poli R., McPhee N. F. General schema theory for genetic programming with subtree-swapping crossover: Part i.

// Evolutionary Computation. — 2003. — V. 11, no. 1. — P. 53–66.

44. Poli R., McPhee N. F. General schema theory for genetic programming with subtree-swapping crossover: Part ii. // Evolutionary Computation. — 2003. — V. 11, no. 2. — P. 169–206.
45. Raoult J. C. On graph rewritings // Theoretical Computer Science. — 1984. — V. 32, no. 1. — P. 1 – 24.
46. Ross S. A. Information and Volatility: The No-Arbitrage Martingale Approach to Timing and Resolution Irrelevancy // The Journal of Finance. — 1989. — V. 44, no. 1. — P. 1–17.
47. Rumelhart D.E., McClelland J.L., University of California San Diego. PDP Research Group. Parallel Distributed Processing: Foundations. A Bradford book. — MIT Press, 1986.
48. Seber G.A.F. The Collected Works of George A.F. Seber. Wiley Series in Probability and Statistics. — Wiley, 2009.
49. Seber G.A.F., Wild C.J. Nonlinear Regression. Wiley Series in Probability and Statistics. — John Wiley & Sons, 2005.
50. Soule T., Heckendorn R. B. An analysis of the causes of code growth in genetic programming // Genetic Programming and Evolvable Machines. — 2002. — V. 3, no. 3. — P. 283–309.
51. Strijov V., Sologub R. Generation of the implied volatility models // Mathematics. Computer. Education. Conference Proceedings. — Moscow: Regular and Chaotic Dynamics, 2009. — P. 270.
52. Model-based problem solving through symbolic regression via pareto genetic programming: Rep. / Tilburg University. — Executor: E. Vladislavleva: 2008.

53. Vladislavleva E.J., Smits G.F., den Hertog D. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming // Evolutionary Computation, IEEE Transactions on. — April. — V. 13, no. 2. — P. 333–349.