

Восстановление поверхности волатильности биржевых опционов помощью индуктивно-порождаемых моделей*

Р. А. Сологуб

alucardische@gmail.ru

Вычислительный центр РАН

В работе решается задача отбора признаков при восстановлении линейной регрессии. Принята гипотеза о нормальном распределении вектора зависимой переменной и параметров модели. Для оценки ковариационной матрицы параметров используется аппроксимация Лапласа: логарифм функции ошибки приближается функцией нормального распределения. Исследуется проблема присутствия в выборке шумовых и коррелирующих признаков, так как при их наличии матрица ковариаций параметров модели становится вырожденной. Предлагается алгоритм, производящий отбор информативных признаков. В вычислительном эксперименте приводятся результаты исследования на временном ряде.

Ключевые слова: *нелинейная регрессия, символьная регрессия, индуктивное порождение, полное порождение, биржевой опцион.*

Введение

В работе решается задача порождения модели оптимальной структуры при восстановлении нелинейной регрессии. Регрессионной моделью в контексте работы называется параметрическое семейство функций, а каждая из порождаемых моделей является суперпозицией функций из некоторого экспертно-заданного множества. Эти функции называются примитивами или порождающими функциями. Для создания модели — суперпозиции порождающих функций выбирается набор этих функций. Порождается набор моделей, выборка разбивается на обучающую и тестовую, параметры моделей оцениваются по обучающей выборке и выбирается модель, максимизирующая коэффициент детерминации. В работе исследуются методы порождения моделей различных классов: линейных, нейросетей, существенно-нелинейных. Предполагается единый алгоритм их порождения.

Использование нелинейной регрессии для решения прикладных задач широко описывается в работах Дж. Себера [1, 2]. В них описывается построение и оценка параметров нелинейных моделей. Для оценки моделей используется алгоритм Левенберга-Марквардта [4].

Для индуктивного порождения моделей в работах Дж. Козы [3] и Н. Зелинки [6], связанных с генетическим программированием, используется символьная регрессия — метод построения регрессионных моделей путем перебора различных произвольных суперпозиций функций из некоторого заданного набора. Индуктивное порождение моделей рассматривается в приложении к задаче определения оптимальной формы антенны [7]. При этом авторы ставят ряд нерешенных вопросов: появление моделей с ненастраиваемыми параметрами, деление на ноль, возникновение комплексных аргументов. Часть этих проблем может быть разрешена с использованием алгоритма, рассматриваемого в данной статье. В работах В. В. Стрижова [8, 9] идеи индуктивного порождения регрессионных моделей находят свое развитие в применении методов Байесовского вывода к процессу порождения и настройки моделей.

Научный руководитель В. В. Стрижов

В данной работе развитие идей индуктивного порождения моделей заключается в создании процедуры порождения моделей вышеперечисленных классов при помощи единообразного подхода к их записи и порождению. В работе ставится задача поиска модели и набора параметров, минимизирующих сумму квадратов невязок, доставляемых построенной моделью на тестовой выборке. Для выбора оптимальной модели, принадлежащей к определенному классу, необходимо осуществлять поиск модели среди всего множества моделей, принадлежащих к данному классу. В связи с этим, требуется показать возможность построения всех моделей этого класса. Для построения бесконечного множества потенциальных моделей требуется бесконечное количество операций, однако заметим, что каждая модель может быть представлена в виде суперпозиции конечного числа функций, являющихся элементами некоторого счетного множества. Отсюда, множество моделей оказывается счетным, поэтому достаточно показать конструктивный способ порождения модели, занимающую заранее известное место при нумерации всех моделей класса.

Для недопущения эффекта переобученности сложность моделей должна ограничиваться. Сложность моделей в рамках данной работы оценивается по методу, предложенному К. Владиславлевой [19]. Для модели, представленной в виде дерева, её сложность является количеству элементов во всех под-деревьях данного дерева.

В качестве иллюстрации предложенного подхода рассматривается задача поиска формулы поверхности волатильности [11] биржевых опционов [10]. Данная задача является важной проблемой биржевой торговли, т.к. позволяет уточнить оценку опционов дальних цен и времен исполнения. Подобные опционы являются одним из основных инструментов страхования биржевых рисков для институциональных инвесторов, а их справедливая оценка необходима для успешной работы маркет-мейкеров — специалистов, отвечающих за наличие небольшой разницы между спросом и предложением на рынке ценных бумаг.

Задача многомерной нелинейной регрессии

Задана выборка — множество пар $D = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, в котором $\mathbf{x} \in \mathbb{R}^P$ — свободная переменная и $y \in \mathbb{R}^1$ — зависимая переменная. Строится отображение $\varphi(\mathbf{x}, \mathbf{w}) \rightarrow \mathbb{R}^1$. Требуется определить 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 . Модель должна соответствовать отображению φ . Для модели определяется набор параметров \mathbf{w}_0 , доставляющие минимум функции квадратичной ошибки

$$S(\mathbf{w}|D, f) = \sum_{n=1}^N (f(\mathbf{x}_n, \mathbf{w}) - y_k)^2.$$

Выражение $S(\mathbf{w}|D, f)$ означает значение S , соответствующее минимизирующему эту ошибку набору параметров \mathbf{w} при заданной выборке D и модели f . Такая модель будет называться оптимальной при условии, что её сложность $C(f)$ не превышает заданной. Сложность определяется как количество элементов во всех под-деревьях, которые можно выделить из дерева, представляющего модель.

Задано множество G порождающих функций $g(\mathbf{w}, \mathbf{x})$. Для каждого элемента данного множества g_i определены области аргументов $\mathbf{w} \in R^m, \mathbf{x} \in R^n$ и значений, при этом область значений принадлежит \mathbb{R}^1 . В множество порождающих функций обязательно входит не имеющая аргументов функция $id(\mathbf{x})$, значение которой тождественно значению свободной переменной. Порождается множество моделей $f \in F$ — допустимых суперпозиций, состоящих из функций $g_i \in G$. Требуется выбрать модель, доставляющую минимум $S(f|\mathbf{w}^*, D)$ при условии, накладываемом на сложность $C(f) < C^*$

Для описания процедуры порождения моделей необходим язык описания моделей, легко интерпретируемый как пользователем — экспертом в предметной области, так и программной системой. Каждой модели поставим в соответствие направленный граф — дерево $\Gamma = \langle V, E \rangle$. Каждой вершине $v_i \in V$ соответствует порождающая функция g_i . Количество ветвей $e_i \in E$, выходящих из каждой вершины v_i будет равно количеству аргументов порождающей функции g_i , соответствующей данной вершине. Листьями дерева Γ являются порождающие функции, не имеющие аргументов — константы $id(C)$ и свободные переменные $id(x)$.

Конструктивное порождение допустимых суперпозиций

В данном разделе будут рассмотрены следующие классы моделей: линейные функции, обобщенно-линейные модели, нейросети, построенные на радиальных базисных функциях. Нейросети общего вида, существенно нелинейные модели. Описание данных классов моделей в терминах порождающих функций и суперпозиций и критерии принадлежности моделей к различным классам разобраны в следующих параграфах. Рассмотрим построение всех моделей для некоторых классов, а также задание ограничений на граф, представляющий модель, так что любой граф описанной структуры будет описывать модель, принадлежащую к определенному классу.

Линейные модели. Класс линейных моделей является наиболее просто устроенным среди рассматриваемых в работе классов. Термин «линейная модель» в контексте данной работы означает модель $f(\mathbf{w}, \mathbf{x})$, являющуюся суммой входных переменных с настраиваемыми коэффициентами — параметрами $\mathbf{w} = [c_0, c_1, \dots, c_n]$.

$$f(\mathbf{w}, \mathbf{x}) = \mathbf{w}^T \mathbf{x} = \sum_i^n c_i x_i + c_0 \quad (1)$$

Для порождения множества линейных моделей требуется две порождающие функции: функция умножения на константу (c) и функция сложения ($+$). Также к множеству примитивов добавляются функции аргумента ($id(x_i)$) и константы ($id(C)$): $G = \{+, c, id(x_i), id(C)\}$. Накладываются следующие правила порождения:

1. В случае, если вершине v_i дерева Γ соответствует примитив $+$, родительской вершине может соответствовать только примитив $+$.
2. В случае, если вершине v_i соответствует примитив c , родительской вершине должен соответствовать примитив $+$.
3. В случае, если вершине соответствует примитив C , родительской вершине должен соответствовать примитив $+$.

Алгоритм 1. Модели порождаются рекурсивно (в каждую заново создаваемую вершину v_i подставляются все возможные для неё порождающие функции $g_j \in G$), при этом порождение идет от вершины дерева v_0 , и каждый следующий элемент g_{j+1} выбирается с учетом приведенных выше правил.

ПРОЦЕДУРА ПостроениеДерева(ТекущееДерево, Примитивы, Родитель)

для $j = 1, \dots, size(Примитивы)$

НовыйЭлемент:=Примитивы(j);

ВременноеДерево:=[ТекущееДерево;НовыйЭлемент];

если ПроверкаПравилПорождения(ВременноеДерево)==1 то

ТекущееДерево:=ВременноеДерево;

НовыеЭлементы=Примитивы(НовыйЭлемент).КоличествоДетей;

для $i = 1, \dots, \text{НовыеЭлементы}$

ТекущееДерево:=ПостроениеДерева(ТекущееДерево, Примитивы, НовыйЭлемент);

return ТекущееДерево

ВЫХОД

Теорема 1. Любая линейная модель вида (1) порождается при использовании данных правил порождения моделей. Любая порожденная алгоритмом 1 модель будет линейной.

Доказательство. Линейная модель имеет вид $y = \sum_i^n c_i x_i + c_0$. Данная запись может быть представлена в польской нотации:

$$y = + \times c_1 x_1 + \times c_2 x_2 \dots + \times c_n x_n c_0.$$

При замене $\times c_i$ как c_i запись принимает вид:

$$y = + c_1 x_1 + c_2 x_2 \dots + c_n x_n c_0.$$

Данная замена эквивалентна опусканию знака умножения в традиционной записи формул. Нотацию справа от знака равенства можно рассматривать как запись дерева при проходе в глубину. Построенное таким образом дерево будет удовлетворять всем условиям, наложенным выше, по построению.

Докажем, что любое дерево, удовлетворяющее изложенным выше условиям, будет задавать линейную модель. Структура данного дерева такова, что родительская вершина примитивов x_i или C может быть или примитивом $+$, или c_i (правило 3). Над примитивом c_i могут располагаться только примитивы $+$ (правила 1,2). Над примитивом $+$ располагается только примитивы $+$ (правило 1), поэтому можно заменить все примитивы $+$ суммой n элементов. В этой сумме будут элементы типа $c_i * x_j$, x_j или C . Таким образом, собрав элементы по свободным переменным, получим сумму свободных переменных с коэффициентами, которые состоят из сумм настраиваемых переменных. Это возможно потому, что любой примитив c_i при объединении попадет только внутрь одной скобки, т.к. у примитива c_i может быть только один потомок. Данная структура полностью соответствует описанию линейной модели.

Обобщенно-линейные модели. Класс обобщенно-линейных моделей расширяет класс линейных моделей включением функций связи.

$$f(\mathbf{w}, \mathbf{x}) = \sum_i^n c_i \mu_i(x_i) + c_0 \quad (2)$$

Для порождения множества обобщенно-линейных моделей требуется две порождающие функции класса линейных моделей $(c, +)$, функции связи μ_i и примитив аргумента $(id(x_i))$. Накладываются следующие правила порождения:

1. В случае, если вершине v_i соответствует примитив $+$, родительской вершине v_j может соответствовать только примитив $+$.
2. В случае, если вершине v_i соответствует примитив c , родительской вершине v_j должен соответствовать примитив $+$.
3. В случае, если вершине v_i соответствует примитив функции связи, родительской вершине v_j должен соответствовать примитив $.$
4. В случае, если вершине v_i соответствует примитив C , родительской вершине v_j должен соответствовать примитив $+$.

Модели порождаются рекурсивно (в каждую заново создаваемую вершину v_i подставляются все возможные для неё порождающие функции $g_j \in G$), при этом порождение идет от вершины дерева v_0 , и каждый следующий элемент g_{j+1} выбирается с учетом приведенных выше правил. Алгоритм построения моделей аналогичен Алгоритму 1.

Теорема 2. Любая обобщенно-линейная модель (2) может быть порождена при использовании данных правил порождения моделей.

Доказательство теоремы 2. Доказательство теоремы 2 аналогично доказательству теоремы 1. Обобщенно-линейная модель имеет вид $y = \sum_i^n c_i \mu_i(x_i) + c_0$. Данная запись может быть представлена в польской нотации:

$$f(\mathbf{w}, \mathbf{x}) = + \times c_1 \mu_1(x_1) + \times c_2 \mu_2(x_2) \dots + \times c_n \mu_n(x_n) c_0.$$

При переобозначении $\times c_i$ как c_i запись принимает вид:

$$f(\mathbf{w}, \mathbf{x}) = + c_1 \mu_1(x_1) + c_2 \mu_2(x_2) \dots + c_n \mu_n(x_n) c_0.$$

Нотацию справа от знака равенства можно рассматривать как запись дерева при проходе в глубину (при этом следует рассматривать $\mu_i(x_i)$ как пару вершин). Построенное таким образом дерево будет удовлетворять всем условиям, наложенным выше, по построению.

Радиальные базисные функции. Класс моделей, построенных на радиально-базисных функциях, является подклассом класса нейронных сетей. Однако, в силу его простого устройства, он также является подклассом класса обобщенно-линейных моделей, в связи с чем доказательство возможности построения всех моделей данного класса не требуется.

$$f(x) = \mathbf{w}^T \boldsymbol{\varphi} \left(\frac{x^2}{\sigma^2} \right) = \sum_{i=1}^U w_i r(x_i) \quad (3)$$

Для порождения множества моделей, построенных на радиально-базисных функциях требуется две порождающие функции: функция суммы (+) и радиально базисные функции (r_i). Также, аналогично классу линейных моделей, к множеству примитивов добавляются функции аргумента (x_i) и константы (C). Накладываются следующие правила порождения:

1. Вершине дерева v_0 ставится в соответствие примитив +.
2. В случае, если вершине v_i соответствует примитив +, родительской вершине v_j может соответствовать только примитив +.
3. В случае, если вершине v_i соответствует примитив радиальной базисной функции k , родительской вершине v_j должен соответствовать примитив +.

Модели порождаются рекурсивно (в каждую заново создаваемую вершину v_i подставляются все возможные для неё порождающие функции $g_j \in G$), при этом порождение идет от вершины дерева v_0 , и каждый следующий элемент g_{j+1} выбирается с учетом приведенных выше правил. Алгоритм построения моделей аналогичен Алгоритму 1.

Теорема 3. Любая модель, построенная на радиально базисных-функциях (3) может быть порождена при использовании данных правил порождения моделей.

Доказательство теоремы 3. Доказательство теоремы 3 полностью аналогично доказательству теоремы 2.

Нейронные сети

Основное отличие нейронных сетей от класса радиальных базисных функций состоит в существовании в нейронных сетях общего вида скрытых слоев нейронов. Формула нейронной сети может быть записана следующим образом:

$$f(\mathbf{w}, \mathbf{x}) = \nu \sum_{i=1}^N u(S_i(\mathbf{x}, \mathbf{w})), \text{ где } S_i$$

$$S_i(x, \mathbf{w}) = \sum_j w_j u(A_j(\mathbf{x}, \mathbf{w})) \quad (4)$$

$$A_j(x, \mathbf{w}) = \mathbf{x}^T \mathbf{w}$$

Данная модель не может быть представлена деревом, так как у одного элемента A_j может быть несколько «родителей». Однако данная проблема может быть обойдена с помощью добавления правила построения, согласно которому всем вершинам, соответствующих одному A_j , будут присвоены одни и те же веса.

Модели порождаются рекурсивно (в каждую заново создаваемую вершину v_i подставляются все возможные для неё порождающие функции $g_j \in G$), при этом порождение идет от вершины дерева v_0 , и каждый следующий элемент g_{j+1} выбирается с учетом приведенных выше правил. Алгоритм построения моделей аналогичен Алгоритму 1, однако добавляется нестандартное правило, что поддерева, являющиеся потомками различных S -вершин, полностью идентичны.

Теорема 3. Любая модель, построенная на радиально базисных-функциях (3) может быть порождена при использовании данных правил порождения моделей.

Доказательство теоремы 3. Доказательство теоремы 3 полностью аналогично доказательству теоремы 2.

Существенно-нелинейные модели. Класс существенно-нелинейных моделей строится как множество моделей, которые могут быть записаны формулой, состоящей из заранее известных элементов. Данные элементы следует отнести к множеству примитивов, и переписать формулу в префиксном виде. Таким образом, оказывается, что любая существенно-нелинейная модель может быть построена в виде дерева, и что любое дерево описывает существенно-нелинейную модель.

Множество всех существенно-нелинейных моделей, являющихся суперпозициями некоторого набора примитивов G может быть построено с помощью алгоритма 1, при этом множество правил будет пустым.

Задача восстановления поверхности волатильности

Для иллюстрации алгоритма порождения моделей рассматривается задача восстановления регрессии поверхности волатильности. 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 созданными ранее [20] 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 настройка 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 .

Поверхность волатильности

Чтобы дать определение подразумеваемой волатильности, сначала необходимо дать описание некоторых терминов из области финансовой математики. Опционом европейского типа называется производная ценная бумага (контракт), дающая ее обладателю *право* купить или продать актив по указанной в опционе цене (цене исполнения опциона) в указанное в опционе время (момент исполнения опциона). Опцион, дающий право купить активы, называется опционом колл. Опцион, дающий право продать активы, называется опционом пут. Таким образом, выигрыш владельца европейского опциона равен

$$d = \max(P(T) - K; 0)$$

где T — момент исполнения опциона, $P(T)$ — цена базового актива, K — цена исполнения опциона.

Сами опционы, будучи ценными бумагами, также являются объектами торговли. Согласно известной формуле Блэка-Шоулза [15]), текущая справедливая цена $c = c(S)$ европейского опциона колл в момент времени t_0 имеет вид

$$c = P(t_0)\Phi(d_1(\sigma)) - K \exp(-r(T - t_0))\Phi(d_2(\sigma)),$$

$$d_1(\sigma) = \frac{\log P(t_0) - \log K + (T - t_0)(r + \sigma^2/2)}{\sigma\sqrt{T - t_0}}$$

$$d_2(\sigma) = d_1(\sigma) - \sigma\sqrt{T - t_0}$$

$\Phi(\cdot)$ стандартная нормальная функция распределения, r — безрисковая ставка доходности, σ — волатильность, то есть стандартное отклонение цены базового актива за календарный год. Подразумеваемой волатильностью σ_{implied} принято называть такое значение волатильности исходных активов, при котором теоретически справедливая цена $c = c(S)$ европейского опциона, вычисленная по формуле Блэка-Шоулза, совпадает с его рыночной ценой c_m . Другими словами, σ_{implied} — это решение уравнения Блэка-Шоулза относительно σ [12].

В связи с понятием неявной волатильности необходимо упомянуть два термина: улыбка волатильности (volatility smile, volatility skew, smile-effect [13]) и поверхность волатильности (volatility surface [14]). Улыбка волатильности получается при построении графика неявной волатильности как функции относительной цены исполнения $M = P/K$ при фиксированном T времени исполнения опциона. При этом функция $\sigma_{\text{implied}} = \sigma_{\text{implied}}(M, T)$ оказывается выпуклой вниз, а ее график напоминает улыбку.

Поверхность волатильности получается при построении графика функции $\sigma_{\text{implied}} = \sigma_{\text{implied}}(M, T)$ как функции двух переменных M и T .

Класс моделей	Число параметров	$C(f)$	MSE_{learn}	MSE_{test}	R_{adj}^2	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

Таблица 1. Результаты вычислительного эксперимента

В данной работе рассматривается прикладная задача восстановления зависимости значения σ_{implied} от значений S/K , T . Для этого производится построение моделей различных классов и отбор среди данных моделей лучших по критериям среднеквадратичной ошибки, коэффициента детерминации и информационного критерия Акаике. Полученная модель сравнивается с рассмотренными ранее, см. [20]. Также данные модели могут быть использованы в реальной торговле для определения возможностей построения календарных позиций. На основе значения волатильности, определенного с помощью восстановленной поверхности, может быть зафиксирован момент неэффективности рынка. В такой момент может быть произведена «продажа волатильности» или «покупка волатильности» [16].

Вычислительный эксперимент

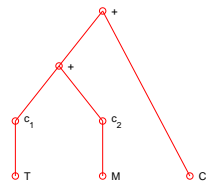
Для анализа выбраны данные торгов опционами Brent Crude Oil — опционы Chicago Mercantile Exchange на сырую нефть марки Brent. Фактически использованы все ликвидные по состоянию на 11 ноября 2011 года опционы — с датами исполнения от 15 ноября 2011 года до 15 декабря 2018 года. Волатильность строилась по данным опционов колл. Следует заметить, что разница оценок волатильности по опционам колл и пут для данного инструмента не превышает 1% в силу высокой ликвидности опционов на нефть. Значения относительной цены исполнения брались в диапазоне от 40% до 200% по наиболее ликвидным опционам.

Выбор данного инструмента обусловлен тем, что опционы на сырьевые товары не являются маржинальными — по ним должна быть обеспечена поставка сырой нефти. В связи с этим данные инструменты не являются объектом массовой спекулятивной торговли из-за возникающих в связи с поставкой рисков.

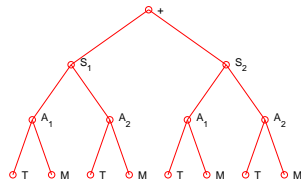
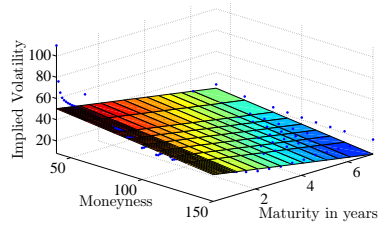
Регрессионная выборка $\{(\mathbf{x}_n, y_n)\} = \{(\langle M_n, t_n \rangle, \sigma_n)\}$ была построена на основе данных системы Bloomberg [17], [18] и дополнена с помощью исходных данных — исторических цен опциона $S_{K,t}$ и базового инструмента P_t , где $K \in \mathcal{K}$, $t \in T$, следующим образом. Для каждого желаемого значения M и $t \in T$ вычисляется значение предполагаемой волатильности как аргумент минимума

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

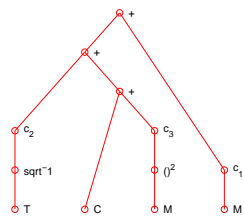
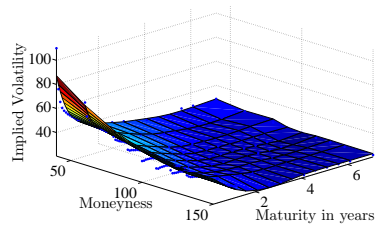
по значению волатильности. Здесь справедливая цена опциона C вычислена по формуле Блэка-Шоулза. Время t выраженное в годах до момента исполнения опциона рассчитывается по формуле $t = \tau/365$, где τ — число дней, оставшихся до исполнения опциона. Значение в искомой точке по шкале относительной цены исполнения берется как линейная комбинация значений в соседних точках. Для индексации выборки задана биекция $(t, M) \mapsto n$. Безрисковая ставка доходности $B = 0.025$, что соответствует ставке доходности по облигациям казначейства США.



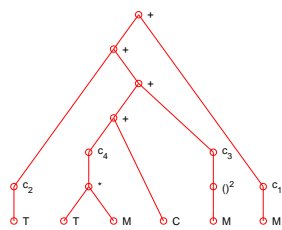
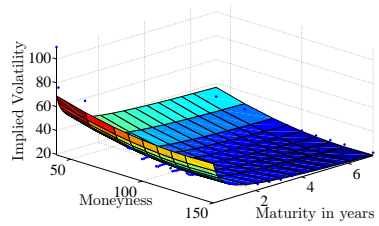
а)



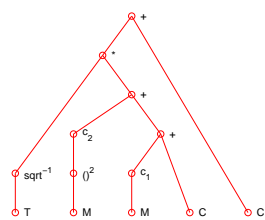
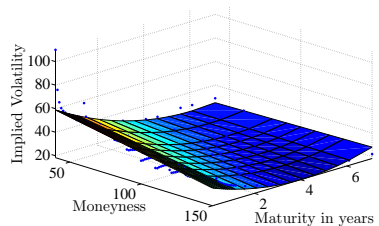
б)



в)



г)



д)

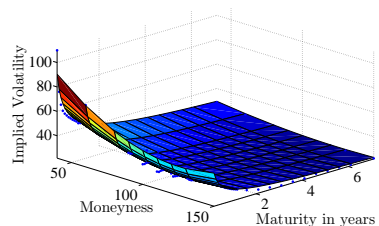


Рис. 1. Примеры структур различных классов моделей и восстановленных поверхностей волатильности

Результаты вычислительного эксперимента

Результаты вычислительного эксперимента и графики, отображающие наилучшие модели, представлены в табл. 1.

Из таблицы 1 можно видеть, что линейная модель выглядит на фоне остальных недостаточно хорошо приближающей данные (плохие значения AIC, SSE) и дает низкий коэффициент детерминации R_{adj}^2 . 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 [20] — нелинейные модели дают лучшую оценку волатильности при меньшем количестве настраиваемых параметров, чем полиномиальные модели значительно большей сложности.

На графиках, представленных на рис. 1 справа, по горизонтальным осям отложены значения относительной цены исполнения M и времени до исполнения опциона t в годах. По вертикальным осям отложены предполагаемые волатильности σ_{implied} , соответствующие реально торгуемому опциону с параметрами M и t . Соответствие рисунков моделям:

- а) линейная модель: $\sigma_{\text{imp}} = c_1M + c_2T + C$,
- б) нейронная сеть: $\sigma_{\text{imp}} = \sum_i = 1^{10} S_i(\sum_j A_j(M, T))$,
- в) обобщенно-линейная модель: $\sigma_{\text{imp}} = c_1M + c_2M^2 + c_3(\sqrt{(T)})^{-1} + C$,
- г) экспертная модель: $\sigma_{\text{imp}} = c_1M + c_2M^2 + c_3T + c_4MT + C$,
- д) существенно-нелинейная модель: $\sigma_{\text{imp}} = \frac{c_1M + c_2M^2 + C_1}{\sqrt{T}} + C_2$,

На рис.1 слева изображены деревья Γ_i , отображающие данные модели при представлении моделей в виде деревьев.

Заключение

В работе предложен алгоритм описания и конструктивного порождения регрессионных моделей. Был проведен анализ неразрешенных проблем в работах, посвященных порождению нелинейных моделей регрессии. Описан способ представления моделей в виде суперпозиций заданных параметрических функций. Доказана корректность данного описания для моделей известных классов. Описан способ порождения всех моделей заданных классов с помощью алгоритма. Для иллюстрации работы алгоритма рассмотрена модель зависимости волатильности биржевого опциона от цены исполнения и времени до исполнения. Модели, полученные в ходе вычислительного эксперимента, оказываются предпочтительнее моделей, которые были предложены экспертами ранее - нелинейные модели дают лучшую оценку волатильности при меньшем количестве настраиваемых параметров, чем полиномиальные модели значительно большей сложности. Результаты экспериментов могут быть использованы для получения справедливой оценки цены опциона.

Литература

- [1] *Seber G. A. F., Wild C. J.* Nonlinear Regression. Wiley-IEEE, 2003.
- [2] *Seber G. A. F., Schwarz C. J.* Estimating Animal Abundance: Review III. Stat Sci. Vol. 14 Num. 4: P. 427–456
- [3] *John R. Koza, Martin A. Keane, James P. Rice* Performance improvement of machine learning via automatic discovery of facilitating functions as applied to a problem of symbolic system

- identification // 1993 IEEE International Conference on Neural Networks I:191–198, San Francisco, USA, 1993.
- [4] *Levenberg K.* A Method for the Solution of Certain Non-Linear Problems in Least Squares // The Quarterly of Applied Mathematics. Vol. 2. P. 164–168.
- [5] *Madala H. R., Ivakhnenko A. G.* Inductive Learning Algorithms for Complex Systems Modeling. CRC Press. 1994.
- [6] *Zelinka, I., Nolle, L., Oplatkova, Z.* Analytic Programming – Symbolic Regression by Means of Arbitrary Evolutionary Algorithms // Journal of Simulation. Vol. 6. No 9. P. 44–56.
- [7] *William Comisky, Jessen Yu, John R. Koza* Automatic synthesis of a wire antenna using genetic programming // Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference, Las Vegas, Nevada. Pages 179–186.
- [8] *Стрижов В. В.* Поиск параметрической регрессионной модели в индуктивно заданном множестве // Журнал вычислительных технологий. 2007. No 1. С. 93–102.
- [9] *Стрижов В. В., Сологуб П. А.* Индуктивное построение регрессионных моделей волатильности опционов // Журнал вычислительных технологий. 2009. No 5. С. 102–113.
- [10] *Hull J. C.* Options, Futures and Other Derivatives. Prentice Hall, 2000.
- [11] *Daglish T., Hull J., Suo W.* Volatility Surfaces: Theory, Rules of Thumb, and Empirical Evidence // Quantitative Finance. Vol. 7, No. 5. 2007. P. 507–524.
- [12] *Jackwerth J., Rubenstein M.* Recovering Probability Distributions from Option Prices // Journal of finance, Vol. 51 No. 5, December 1996.
- [13] *Ross S.* Information and volatility: the no-arbitrage martingale approach to timing and resolution irrelevancy // Journal of Finance, 1989, vol. 44, No. 1, p. 1–17.
- [14] *Kendall M.* The analysis of economic time series: Part I, Prices // J. Royal Statist. Soc., 1953, vol. 96, p. 11–25.
- [15] *Black F., Scholes M.* The Pricing of Options and Corporate Liabilities // Journal of Political Economy, 81, 637–654
- [16] *Dupire B.* Pricing with a smile // Risk Vol. 7, P. 18–20.
- [17] *Brigo D., Mercurio F.* Dynamics and Calibration to Market Volatility Smiles // International Journal of Theoretical & Applied Finance. Vol. 5(4). P. 427–446.
- [18] *Fouque J-P., Papanicolaou G., Sircar K. R.* Derivatives in Financial Markets with Stochastic Volatility, Cambridge University Press. 2000.
- [19] *Vladislavleva E., Smith G., Hertog D.* Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming // IEEE Transactions on Evolutionary Computation. Vol. 13(2). P. 333–349.
- [20] *Alentorn A.* Modelling the implied volatility surface: an empirical study for FTSE options. May 2004.