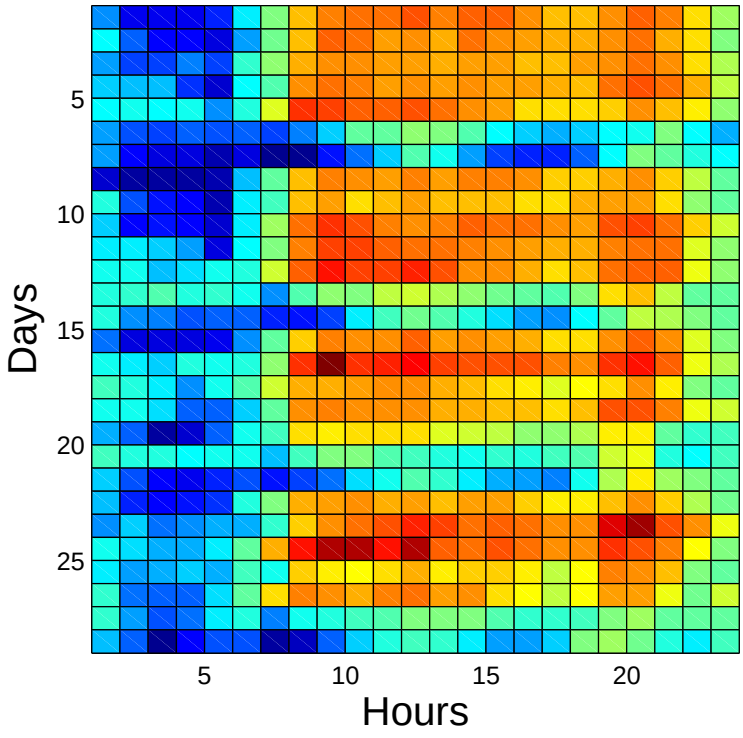
ISSN 2223-3792

Машинное обучение и анализ данных

Декабрь 2011 **Том 1, номер 2**



Машинное обучение и анализ данных

Journal of Machine Learning and Data Analysis

ISSN 2223-3792 Rus

Журнал публикует статьи, содействующие развитию теоретических и прикладных методов машинного обучения и интеллектуального анализа данных. Журнал, прежде всего, предназначен для публикации результатов работ аспирантов и студентов, изучающих курс «Численные методы машинного обучения» и занимающихся теоретическими и эмпирическими исследованиями свойств алгоритмов регрессии и классификации. Приветствуются также обзорные, фундаментальные и методические статьи исследователей, работающих в области машинного обучения.

Тематика журнала:

- регрессионный анализ,
- классификация,
- кластеризация,
- многомерный статистический анализ,
- байесовские методы регрессии и классификации,
- методы прогнозирования временных рядов,
- методы оптимизации в задачах машинного обучения и анализа данных,
- методы визуализации данных,
- обработка и распознавание речи и изображений,
- анализ и понимание текста, информационный поиск,
- прикладные задачи анализа данных.

Научный редактор: В. В. Стрижов (stijov@ccas.ru)
Вёрстка: А. А. Мафусалов, П. А. Сечин

Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

Москва, 2011

От редактора

Второй семестр курса «Численные методы обучения по прецедентам» для студентов кафедры «Интеллектуальные системы» ФУПМ МФТИ посвящен изучению технологии научной работы и называется «Автоматизация и стандартизация научных исследований», см. соответствующие статьи на сайте «MachineLearning.ru». Курс преследует две цели. Общая цель — научиться точно, ясно, красиво излагать свои и чужие идеи. Частная цель — написать научную статью или статью с элементами научной новизны, которая была бы принята другими исследователями, работающими в нашей области и сделать доклад. Предполагаемым результатом работы студента является статья, поданная в рецензируемый журнал из списка ВАК. В течение курса студенты еженедельно делают краткие доклады о ходе работ и их содержании, а также слушают лекции о способах постановки задач машинного обучения, об организации научной работы и о форматах представления ее результатов.

План работы над статьей для студента четвертого курса, осенний семестр 2011 года, выглядит следующим образом.

Дата	Список работ	Представленный результат
5.09	Вводная лекция, представление тем исследований.	Тема выбрана.
12.09	Поиск/получение и описание данных; поиск публикаций, создание bib-базы. Написание аннотации.	Аннотация, описание данных.
19.09	Визуализация данных, презентация графиков, рассказ о собранной литературе.	Графики, рассказ о методах, bib-файл.
26.09	Написание введения: обзор методов решения задачи, описание предлагаемого подхода в целом.	Раздел «Введение».
3.10	Постановка задачи, описание новизны подхода, написание черновика решения задачи.	Раздел «Постановка задачи».
10.10	Постановка вычислительного эксперимента, получение первых результатов.	Результаты вычислительного эксперимента.
17.10	Описание предлагаемого подхода в деталях.	Центральный раздел статьи.
24.10	Вычислительный эксперимент завершен.	Раздел «Вычислительный эксперимент» с графиками и таблицами.
31.10	Описание результатов, последняя часть.	Раздел «Заключение».
7.11	Завершение критической части статьи, анализ ошибок.	Критическое сравнение результатов.
14.11	Корректировка статьи: ее структура и последовательность изложения.	Замечания рецензента.
21.11	Корректировка: теоретическая часть и обозначения.	Проработанная теоретическая часть.
28.11	Корректировка: согласованность терминологии.	Статья, доступная для понимания.
5.11	Контрольная точка представления готового варианта статьи, выбор журнала.	Статья по шаблону журнала.
12.11	Сделан доклад по статье, статья подана в журнал.	Пакет документов в редакции.

Этот номер журнала также содержит работы-эссе студентов четверного курса. Цель их исследований — выявить проблемы известных методов при практическом использовании, в частности, при анализе данных, имеющих сложную структуру и требующих мультимодельного подхода, анализа наличия шумовых и мультикоррелирующих признаков или оценки ковариационных матриц параметров моделей. Работы выполнены в сокращенном варианте, без введения в предметную область.

Систематический подход к исследованиям хорош тем, что за неделю автор уже успевает отдохнуть от текста, но еще держит в памяти состояние своей работы. Это позволяет написать научную статью в течение семестра, потратив разумное количество времени.

Успехов в научных исследованиях!

Вадим Викторович Стрижов

Содержание

Л. Н. Леонтьева
Выбор моделей прогнозирования цен на электроэнергию 129
А. А. Токмакова
Получение устойчивых оценок гиперпараметров линейных регрессионных моделей 140
М. П. Кузнецов
Уточнение ранговых экспертных оценок с использованием монотонной интерполяции 156
А. А. Зайцев
Исследование устойчивости оценок ковариационной матрицы признаков 165
Р. А. Сологуб
Восстановление поверхности волатильности биржевых опционов помощью индуктивно-порождаемых моделей 174
Г. И. Рудой
Индуктивное порождение суперпозиций в задачах нелинейной регрессии 185
М. Е. Панов
Аппроксимация функции ошибки 200
К. С. Скипор
Выбор признаков в задачах логистической регрессии 205
К. В. Павлов
Оценка параметров смеси распределений 222
А. П. Мотренко
Многоклассовый прогноз вероятности наступления инфаркта 227
А. А. Романенко
Событийное моделирование и прогноз финансовых временных рядов 238
Е. А. Будников
Обзор некоторых статистических моделей естественных языков 245

Выбор моделей прогнозирования цен на электроэнергию*

Л. Н. Леонтьева
liubov.sanduleanu@gmail.com
Московский физико-технический институт, ФУПМ, каф. «Интеллектуальные системы»

Исследуется проблема оптимальной сложности модели в связи с ее точностью и устойчивостью. Задача состоит в нахождении наиболее информативного набора признаков в условиях их высокой мультиколлинеарности. Для выбора оптимальной модели используется модифицированный алгоритм шаговой регрессии, являющийся одним из алгоритмов добавления и удаления признаков. В работе предложен метод поиска оптимальной модели прогнозирования цен на электроэнергию. В вычислительном эксперименте приведены результаты работы алгоритмов на временных рядах почасовых цен на электроэнергию.

Ключевые слова: *отбор признаков, мультиколлинеарность, шаговая регрессия, метод Белсли, прогнозирование временных рядов.*

Введение

Решается задача восстановления линейной регрессии при наличии большого числа мультиколлиниарных признаков. Термин «мультиколлинеарность» введен Р. Фишером при рассмотрении линейных зависимостей между признаками [1]. Проблема состоит в том, что количество признаков значительно превосходит число зависимых переменных, то есть мы имеем дело с переопределенной матрицей. Для решения этот задачи необходимо исключить наиболее малоинформативные признаки. Для отбора признаков предлагается использовать модифицированный метод шаговой регрессии.

Ранее для решения подобных задач использовались следующие методы: метод наименьших углов LARS [2], Лассо [3], ступенчатая регрессия [4], последовательное добавление признаков с ортогонализацией FOS [5, 6], шаговая регрессия [4, 7, 8] и другие [15]. Шаговыми методами называются методы, заключающихся в последовательном удалении или добавлении признаков согласно определенному критерию. Существует несколько недостатков метода, например, важная переменная может быть никогда не включена в модель, а второстепенные признаки будут включены.

В работе предложен модифицированный метод шаговой регрессии. Шаговыми методами называются методы, заключающиеся в последовательном удалении или добавлении признаков согласно определенному критерию. Метод включает два основных шага: шаг Add (последовательное добавление признаков) и шаг Del (последовательное удаление признаков). Добавление признаков производится с помощью FOS [5, 6]. Данный метод последовательно добавляет признаки которые максимально коррелируют с вектором регрессионных остатков. Удаление признаков в нашей работе осуществляется методом Белсли [9]. Он позволяет выявить мультиколлинеарность признаков используя сингулярное разложение матрицы признаков. Для нахождения алгоритма, который доставляет одновременно точную и устойчивую, в смысле минимизации числа мультиколлиниарных признаков, модель, предложен новый метод останова этапов Add и Del, а так же останова всего алгоритма.

Научный руководитель В. В. Стрижов

Предложенный метод выбора модели проилюстрирован задачей прогнозирования почасовых цен на элекроэнергию на сутки вперед. Ранее эта задача решалась с помощью гребневой регрессии [10], метода наименьших углов, построения локальных регрессионных моделей [12, 13] и других.

Помимо поиска оптимального набора признаков, необходимо выбрать подходящий метод прогнозирования. Особенность нашей прикладной задачи заключается в наличии мультипериодичности данных, то есть наличии нескольких периодов: день, неделя, год.

Для построения прогноза в работе предлагается использовать авторегрессионный алгоритм. В основе этого алгоритма лежит построение авторегрессионной матрицы, в которую построчно укладывается временной ряд, причем длина строки (ширина матрицы авторегрессии) равна периодике — 24-м часам. Таким образом, каждый столбец содержит цену в некоторый час по всем суткам, и рассматривается как признак в задаче регрессии, а каждая строка является элементом выборки. Строятся 24 регрессионные модели — для прогнозирования цен на каждый час следующих суток. При использовании линейных моделей задача может быть решена методом наименьших квадратов.

В вычислительном эксперименте проведено сравнение предлагаемого алгоритма с базовым методом SSA [14].

Работа состоит из трех основных частей. Первая часть посвящена прогнозированию с помощью авторегрессионной матрицы. Во второй части описан выбор признаков при прогнозировании, здесь же можно найти описание метода Белсли. В последнем разделе приведены результаты вычислительного эскперимента, проведенного на основе данных почасовых цен на электроэнергию в Германии.

Задача прогнозирования с помощью авторегрессионной матрицы

Даны временной ряд $\mathbf{s}_1 = \{x_i\}_{i=1}^{T}$, будем называть его целевым рядом, и матрица признаков, столбцами которой являются временные ряды $\mathbf{s}_2, \mathbf{s}_3 \dots \mathbf{s}_p$. Необходимо спрогнозировать следующие τ значений ряда $\mathbf{s}_1$. Предполагается, что

— отсчеты x_i сделаны через равные промежутки времени,
— ряд s имеет периодическую составляющую τ,
— ряд s не имеет пропущенных значений,
— длина ряда s кратна периоду τ.

Для нахождения оптимальной модели, предлогается построить алгоритм прогноза, позволяющий решать задачи прогнозирования периодических рядов. С помощью этого алгоритма строится прогноз по выбранному набору признаков. В нашей работе предлагается использовать метод авторегрессии.

Сначала опишем как строится прогноз методом авторегрессии без учета вспомогательных рядов $\mathbf{s}_2, \mathbf{s}_3 \dots \mathbf{s}_p$, а затем обобщим метод на случай многомерного ряда (метод многомерной авторегрессии).

Пусть длина временного ряда $T = m\tau$. Построим авторегрессионную матрицу $\mathbf{X}^*$:

$$\mathbf{X}^* = \left(\begin{array}{cccc|c} x_1 & x_2 & \dots & x_{\tau-1} & x_\tau \\ \dots & \dots & \dots & \dots & \dots \\ x_{j\tau+1} & x_{j\tau+2} & \dots & x_{(j+1)\tau-1} & x_{(j+1)\tau} \\ \dots & \dots & \dots & \dots & \dots \\ x_{(m-2)\tau+1} & x_{(m-2)\tau+2} & \dots & x_{(m-1)\tau-1} & x_{(m-1)\tau} \\ \hline x_{T-\tau+1} & x_{T-\tau+2} & \dots & x_{T-1} & x_T \end{array}\right).$$

Введем обозначения:

$$\mathbf{X}^* = \left(\begin{array}{c|c} \mathbf{X} & \mathbf{y} \\ \hline \mathbf{x}_m & x_T \end{array}\right).$$

Необходимо построить линейную регрессию:

$$\mathbf{y} = \mathbf{X}\mathbf{w}, \tag{1}$$

где $\mathbf{w}$ — вектор параметров. Тогда получим

$$x_T = \langle \mathbf{x}_m, \mathbf{w} \rangle .$$

Требуется решить задачу минимизации евклидовой нормы вектора регрессионных остатков

$$||\mathbf{X}\mathbf{w} - \mathbf{y}||^2 \to \mathbf{min}.$$

Вектор параметров $\mathbf{w}$ отыскивается с помощью метода наименьших квадратов

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y}).$$

Однако зависимость $\mathbf{y} = \mathbf{f}(\mathbf{w}, \mathbf{X})$ может быть существенно нелинейной относительно свободных переменных, и для построения линейной модели удовлетворительного качества необходимо расширить множество признаков с помощью функциональных преобразований исходных признаков.

Зададим

$$\mathbf{X} = \begin{pmatrix} g_1(x_1) & \dots & g_1(x_1) & \dots & \dots & g_r(x_\tau) & \dots & g_r(x_\tau) \\ g_1(x_{\tau+1}) & \dots & g_1(x_{\tau+1}) & \dots & \dots & g_r(x_{2\tau}) & \dots & g_r(x_{2\tau}) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_1(x_{T-\tau+1}) & \dots & g_1(x_{T-\tau+1}) & \dots & \dots & g_r(x_T) & \dots & g_r(x_T) \end{pmatrix},$$

где множество функций $G = \{g_k | k = 1, \dots, r\}$ задано экспертом (например функции $g_1 = = 1$, $g_2 = \sqrt{x}$, $g_3 = x$, $g_4 = x\sqrt{x}$).

В случае многомерного ряда при построении авторегрессионной матрицы необходимо учитывать временные ряды $\mathbf{s}_2, \mathbf{s}_3 \dots \mathbf{s}_p$. Сначала строится авторегрессионная матрица $\mathbf{X}_1$ для ряда $\mathbf{s}_1$. Для каждого следующего временного ряда $\mathbf{s}_j$, где $j = 1, \dots, p$, строится авторегрессионная матрица $\mathbf{X}_j$. Но для рядов $\mathbf{s}_2, \mathbf{s}_3 \dots \mathbf{s}_p$ не вводятся вектора $\mathbf{y}$ в отличии от ряда $\mathbf{s}_1$. То есть для этих рядов авторегрессионная матрица будет содержать τ столбцов, а не $\tau - 1$, как для целевого ряда. Присоединив авторегрессионные матрицы всех временных рядов, получим матрицу

$$\mathbf{X} = [\mathbf{X}_1 | \mathbf{X}_2, \dots, \mathbf{X}_p] .$$

Откуда получим

$$\mathbf{X} = \left(\begin{array}{ccccc|ccc} x_1 & x_2 & \dots & x_{\tau-1} & x_\tau & s_2^1 & \dots & s_p^1 \\ x_{\tau+1} & x_{\tau+2} & \dots & x_{2\tau-1} & x_{2\tau} & s_2^2 & \dots & s_p^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{T-\tau+1} & x_{T-\tau+2} & \dots & x_{T-1} & x_T & s_2^m & \dots & s_p^m \end{array}\right),$$

где s_i^j — j-ое значение ряда $\mathbf{s}_i$.

Опишим в чем состоит задача выбора оптимальной модели. Задана выборка $D = = (\{\mathbf{x}_i, y_i\}), i \in \mathcal{I}$, где множество свободных переменных — вектор $\mathbf{x} = [x_1, \dots, x_j, \dots, x_n]$,

проиндексированно $j \in \mathcal{J} = \{1, \ldots, n\}$. Задано разбиение множества индексов элементов выборки $\mathcal{I} = \mathcal{L} \sqcup \mathcal{C}$. Так же задан класс регрессионных моделей $f(\mathbf{w}, \mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$ — параметрических функций, линейных относительно параметров. Функция ошибки задана следующим образом

$$S = \sum_{i \in \mathcal{X}} (y_i - f(\mathbf{w}, \mathbf{x}_i))^2, \tag{2}$$

где $\mathcal{X} \subseteq \mathcal{I}$ —некоторое множество индексов. Требуется найти такое подмножество индексов $\mathcal{A} \subseteq \mathcal{J}$, которое бы доставляло минимум функции

$$\mathcal{A}^* = \arg \min_{\mathcal{A} \subseteq \mathcal{J}} S(f_{\mathcal{A}} | \mathbf{w}^*, \mathcal{D}_{\mathcal{C}}) \tag{3}$$

на множестве индексов $\mathcal{C}$. При этом парамептвы $\mathbf{w}^*$ модели должны доствалять минимум функции

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathcal{W}} S(\mathbf{w} | \mathcal{D}_{\mathcal{L}}, f_{\mathcal{A}}) \tag{4}$$

на множестве индексов $\mathcal{L}$. Здесь $f_{\mathcal{A}}$ обозначает модель f, включающую только столбцы матрицы X с индексами из множества $\mathcal{A}$, а обозначение вида $S(\mathbf{w}|\mathcal{D})$ означает, что переменная $\mathcal{D}$ фиксирована, а переменная $\mathbf{w}$ изменяется.

Выбор признаков при прогнозировании

Предположим, что мы имеем данные о цене на электроэнергию за год. Тогда матрица $\mathbf{X}$ имеет размерность $(\tau - 1) \times (n - 1)$, где $\tau = 24$, а $n = 365$. То есть $\mathbf{X}$ является матрицей 23×364. Как видно, авторегрессионная матрица является переопределенной, поэтому необходим отбор признаков. Для отбора признаков в предлагается модифицировать метод шаговой регрессии.

Процедура выбора оптимального набора признаков

Опишем два этапа алгоритма Add и Del. На первом этап последовательно добавляются признаки согласно (4), доставляющие минимум S на обучающей выборке, заданной множеством индексов $\mathcal{L}$. На втором этапе происходит последовательное удаление признаков согласно методу Белсли. Пусть на k-ом шаге алгоритма имеется активный набор признаков $\mathcal{A}_k \in \mathcal{J}$. На нулевом шаге $\mathcal{A}_0$ пуст. Опишем этапы Add и Del.

Этап Add. Находим признак доставляющий минимум S на обучающей выборке

$$j^* = \arg \min_{j \in \mathcal{J} \setminus \mathcal{A}_{k-1}} S(\mathbf{w} | \mathcal{D}_{\mathcal{L}}, f_{\mathcal{A}_{k-1} \cup \{j\}}).$$

Затем добавляем новый признак j^* к текущему активному набору

$$\mathcal{A}_k = \mathcal{A}_{k-1} \cup \{j^*\}$$

и повторяем эту процедуру до тех пор пока $S(f_{\mathcal{A}_k} | \mathbf{w}^*, \mathcal{D})$ превосходит свое минимальное значение на данном этапе не более чем на некоторое заданное значение ΔS_1.

Этап Del. Находим индексы обусловленности и долевые коэффициенты для текущего набора признаков $\mathcal{A}_{k-1}$ согласно методу Белсли, описание которого приведено ниже. Далее находим максимальный индекс обусловленности

$$i^* = \arg \max_{i \in \mathcal{A}_{k-1}} \eta_i. \tag{5}$$

Затем ищем максимальный долевой коэффициент соответствующий найденному индекс обусловленности η_{i^*}

$$j^* = \arg \max_{j \in \mathcal{A}_{k-1}} q_{i^* j}. \tag{6}$$

Удаляем j^*-ый признак из текущего набора

$$\mathcal{A}_k = \mathcal{A}_{k-1} \backslash j^*$$

и повторяем эту процедуру до тех пор пока $S(f_{\mathcal{A}_k}|\mathbf{w}^*, \mathcal{D})$ превосходит свое минимальное значение на данном этапе не более чем на некоторое заданное значение ΔS_2.

Повторение этапов Add и Del осуществляется до тех пор пока значение $S(f_{\mathcal{A}_k}|\mathbf{w}^*, \mathcal{D})$ не стабилизируется.

Метод Белсли для удаления признаков

Рассмотрим матрицу признаков $\mathbf{X}$. Она имеет размерность $m \times n$. Выполним ее сингулярное разложение:

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T,$$

где $\mathbf{U}$, $\mathbf{V}$ — ортогональные матрицы размерностью соответственно $m \times m$ и $n \times n$ и $\mathbf{\Lambda}$ — диагональная матрица с элементами (сингулярными числами) на диагонали, такими что

$$\lambda_1 > \lambda_2 > \ldots > \lambda_r,$$

где r — ранг матрицы $\mathbf{X}$. Заметим, что в нашем случае $r = n$. Это связано с тем, что в алгоритме шагового выбора на каждом шаге мы имеем мультиколлиниарный, но невырожденный набор признаков.

Столбцы матрицы $\mathbf{V}$ являются собственными векторами, а квадраты сингулярных чисел — собственными значениями корреляционной матрицы $\mathbf{X}^T\mathbf{X}$.

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^T\mathbf{U}^T\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T = \mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^T,$$

$$\mathbf{X}^T\mathbf{X}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}^2.$$

Отношение максимального сингулярного числа к j-му сингулярному числу назовем индексом обусловленности с номером j

$$\eta_j = \frac{\lambda_{\max}}{\lambda_j}.$$

Если матрица X неполноранговая, то значительная часть индексов обусловленности неопределено. Однако, в нашем случае, как упомяналось выше, матрица признаков X является матрицей полного ранга.

Используя сингулярное разложение, дисперсия параметров, найденных методом наименьших квадратов $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, где $\mathbf{w}$ — вектор параметров модели, может быть записана как

$$\mathbf{var}(\mathbf{w}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} = \sigma^2(\mathbf{V}^T)^{-1}\mathbf{\Lambda}^{-2}\mathbf{V}^{-1} = \sigma^2 V D^{-2} V^T.$$

где σ^2 — это дисперсия регрессионных остатков. Таким образом дисперсия j-го регрессионного коэффициента — это j-й диагональный элемент матрицы $\mathbf{Var}(\mathbf{w})$.

Для обнаружения мультиколлинеарности признаков построим таблицу, в которой каждому индексу обусловленности η_j соответствуют значения q_{ij} — долевые коэффициенты. Сумма долевых коэффициентов по индексу j равна единице.

$$\sigma^{-2}\mathbf{var}(w_i) = \sum_{j=1}^{n} \frac{v_{ij}^2}{\lambda_j^2} = (q_{i1} + q_{i2} + \ldots + q_{in}) \sum_{j=1}^{n} \frac{v_{ij}^2}{\lambda_j^2},$$

где q_{ij} — отношение соответствующего слагаемого в разложении вектора $\sigma^{-2}\mathbf{var}(w_i)$ ко всей сумме, а $\mathbf{V} = (v_{ij})$.

Таблица 1. Разложение $\mathbf{var}(w_i)$

Индекс обусловленности	$\mathbf{var}(w_1)$	$\mathbf{var}(w_2)$	$\ldots$	$\mathbf{var}(w_n)$
η_1	q_{11}	q_{21}	$\cdots$	q_{n1}
η_2	q_{12}	q_{22}	$\cdots$	q_{n2}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
η_n	q_{1n}	q_{2n}	$\cdots$	q_{nn}

Из таблицы (1) определяется мультиколлинеарность: большие величины η_j означают, что возможно есть зависимость между признаками. Большие значения q_{ij} в соответствующих строках относятся к признакам, между которыми эта зависимость существует. Маленькие значения η_j также исследуются: между признаками, соответствующими большим значениям q_{ij}, зависимости не существует. Для нахождения мультиколлиниарных признаков решаются задачи (5) и (6).

Описание базового алгоритма прогноза

В вычислительном эксперименте проведено сравнение предлагаемого алгоритма с базовым методом SSA. Приведем его краткое описание.

Для последующего разложение ряда $\mathbf{s_1} = \{x_i\}_{i=1}^{T}$ по главным компонентам преобразуем ряд в траекторную матрицу (матрицу Ганкеля) $\mathbf{Y}$, которую строим следующим образом:

$$\mathbf{Y} = \begin{pmatrix} x_1 & x_2 & \ldots & x_n \\ x_2 & x_3 & \ldots & x_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & \ldots & x_T \end{pmatrix}. \tag{7}$$

где $n = T - m + 1$ — время жизни гусеницы. Матрицу (7) будем называть нецентрированной траекторной матрицей, порожденной гусеницей длины l. Проводимый в дальнейшем анализ главных компонент может проводиться как по центрированной, так и по нецентрированной выборкам. Для упрощения выкладок рассмотрим простейший нецентрированный вариант.

Построим ковариационную матрицу следующим образом:

$$\mathbf{C} = \frac{1}{n}\mathbf{Y}^T\mathbf{Y}.$$

Так как матрица Ганкеля $\mathbf{Y}$ невырождена, то матрица $\mathbf{C}$ является полноранговой, то есть ее ранг равен n. Выполним её сингулярное разложение:

$$\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T,$$

где $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ — диагональная матрица собственных чисел, $\mathbf{V} = [v^1, \ldots, v^n]$ — ортогональная матрица собственных векторов-столбцов. При этом будем предполагать, что собственные векторы упорядочены по убыванию соответствующих собственных чисел, т. е. $\lambda_1 > \lambda_2 > \ldots > \lambda_n$.

Перейдем к прогнозированию временных рядов методом гусеницы.

Рассмотрим систему уравнений:

$$\begin{cases} \sum\limits_{j=1}^{n} h_j v_1^j = x_{m+1}, \\ \ldots \\ \sum\limits_{j=1}^{n} h_j v_{n-1}^j = x_T. \end{cases} \tag{8}$$

Пусть $\mathbf{h}^* = [h_1^*, \ldots, h_n^*]$ — решение системы (8), тогда для продолжения ряда получим

$$x_{T+1} = \sum_{j=1}^{n} h_j^* v_n^j.$$

Выбераем r главных компонент из матрицы V

$$\mathbf{V}^* = \begin{pmatrix} v_1^{i_1} & v_1^{i_2} & \ldots & v_1^{i_r} \\ v_2^{i_1} & v_2^{i_2} & \ldots & v_2^{i_r} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n-1}^{i_1} & v_{n-1}^{i_2} & \ldots & v_{n-1}^{i_r} \end{pmatrix},$$

где $i_1, \ldots, i_r$ — номера выбранных главных компонент. Введем следующие обозначения $\mathbf{v} = (v_n^{i_1}, v_n^{i_2}, \ldots, v_n^{i_r})$, $\mathbf{q} = (x_{m+1}, \ldots, x_n)^T$ и $\tilde{\mathbf{h}} = (h_{i_1}, \ldots, h_{i_r})^T$. В этих обозначениях система (8) запишется как

$$\mathbf{V}^* \tilde{\mathbf{h}} = \mathbf{q}. \tag{9}$$

Учитывая (9), можно записать для прогнозируемого значения x_{T+1} следующую формулу:

$$x_{T+1} = \mathbf{v}((\mathbf{V}^*)^T \mathbf{V}^*)^{-1} (\mathbf{V}^*)^T \mathbf{q}.$$

Таким образом мы построили прогноз — следующее по времени значение x_{T+1} временного ряда $\mathbf{s}_1 = \{x_i\}$.

Вычислительный эксперимент

С целью сравнить предложенный в настоящей работе подход с базовым алгоритмом SSA была проведена серия экспериментов по краткосрочному прогнозированию временных рядов. Данными послужили почасовые цены на электроэнергию в Германии за период с 1 января 2003 г. по 10 июля 2009 г. [16]. Мы прогнозируем почасовые цены на ближайшие сутки по предидущей истории.

На рис. 1 приведен график отражающий колебание цен на электроэнергию. На нем мы видим суточную и недельную периодики.

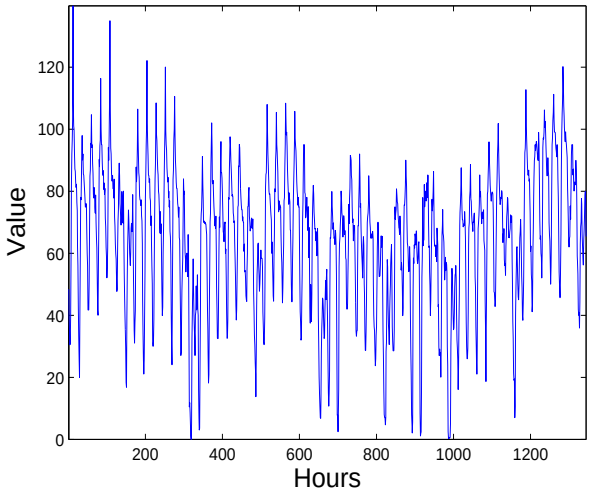


Рис. 1. График зависимости цен от времени за 2 месяца

Более детально поведение цен за неделю изображено на рис. 2. На котором виден спад цен в ночное время и в выходных дни.

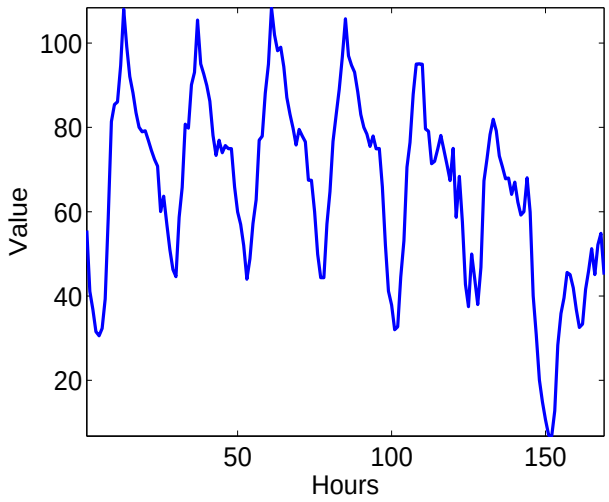


Рис. 2. Диаграмма зависимости цен от времени суток и дня недели

Диаграмма на рис. 3 показывает изменение цен за четыре недели в зависимости от времени суток.

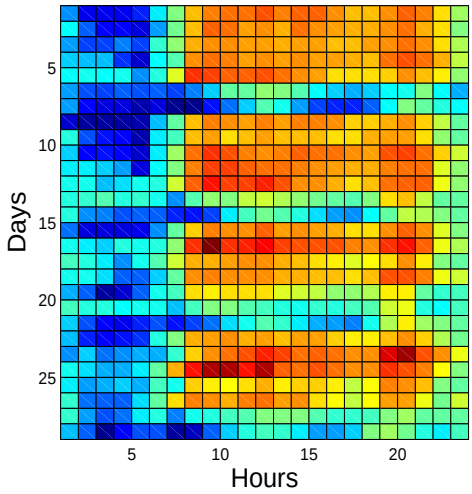


Рис. 3. График зависимости цен от времени за неделю

На рис. 4 приведены два графика. Первый отражает зависимость реальных данных о ценах от времени, а второй — зависимость спрогнозированных, с помощью предложенного в работе алгоритма, на сутки цен от времени.

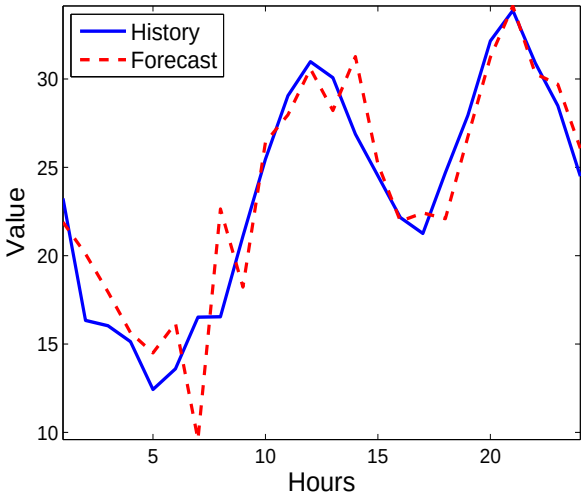


Рис. 4. Прогнозирование почасовых цен на сутки без учета рядов-признаков

На рис. 5 показано, как изеняется квадратичная ошибка S, определенная в (2), на тестовой выборке $\mathcal{C}$, от итерации к итерации.

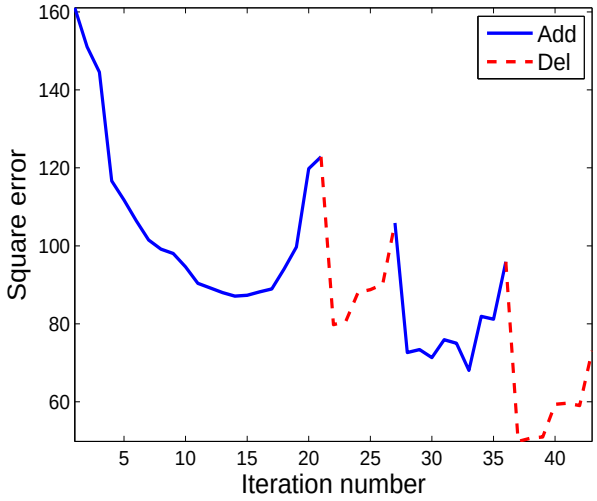


Рис. 5. Зависимость S от номера итерации

Сравним результаты работы предложенного алгоритма с базовым алгоритмом SSA. Результаты экспериментов показаны в таблице (2).

Таблица 2. Результаты работы алгоритмов

	MSE	MAPE в рабочие дни	MAPE в выходные дни	AIC	BIC	Число признков n
Предложенный алгоритм	8.18	6.17	10.33	96.44	123.53	23
Метод SSA	13.25	16.16	29.01	130.74	149.59	16

Для каждого алгоритма вычислялись ошибки MSE (средний квадратичное отклонение) и MAPE (средняя абсолютная процентная ошибка):

$$\text{MSE} = \frac{1}{\tau}\sum_{i=T+1}^{T+\tau}(\tilde{x}_i - x_i)^2, \tag{10}$$

$$\text{MAPE} = \frac{1}{\tau}\sum_{i=T+1}^{T+\tau}100\frac{|\tilde{x}_i - x_i|}{|x_i|}, \tag{11}$$

где $\tilde{x}_i$ — спрогнозированное значение целевого рядя в точке i, x_i — фактическое значение этого ряда в точке i. Так же сравнивались значения информационных критериев Акаике (AIC) и Байеса (BIC)

$$\text{AIC} = \tau(\ln\frac{S}{\tau}) + 2|\mathcal{A}|,$$

$$\text{BIC} = \tau(\ln\frac{S}{\tau}) + |\mathcal{A}|\ln\tau,$$

где S — среднеквадратичная ошибка, вычисленная по набору активных признаков $\mathcal{A}$. В таблице так же указанно число n признаков, входящих в модель.

Заключение

В работе предложен метод поиска оптимальной модели, основанный на комбинации двух стратегий: отбор признаков и выбор модели. Как показал вычислительный эксперимент предложенный подход значительно эффективнее базового алгоритма. Особенно полезен предложенный метод в случае, когда данные содержат большое число мультиколлинеарных признаков.

Вычислительный эксперимент показал, что увеличение числа признаков позволяет добиться улучшения качества модели. Однако, при этом требуется введение дополнительных условий, позволяющих избежать появление мультиколлинеарных признаков. Предлагаемый алгоритм включает процедуру анализа мультиколлинеарности и позволяет получать хорошо обусловленные наборы порожденных признаков.

Литература

[1] Frisch R. *Statistical Confluence Analysis by means of complete regression systems*, Universitetets Okonomiske Institute, 1934.

[2] Efron B., Hastie T., Johnstone I., Tibshirani R. *Least angle regression*, The Annals of Statistics, 2004, Vol. 32, no. 3., Pp. 407-499.

[3] Tibshirani R. *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society, 1996, Vol. 32, no. 1, Pp. 267-288.

[4] Draper N. R., Smith H. *Appied Regression Analysis*,John Wihley and Sons, 1998.

[5] Chen Y. W., Billings C. A., Luo W. *Orthogonal least squares methods and their application to non-linear system identication*, International Journal of Control, 1989, Vol. 2, no. 50, Pp. 873-896.

[6] Chen S., Cowan C. F. N., Grant P. M. *Orthogonal least squares learning algorithm for radial basis function network*, Transaction on neural netwark, 1991, Vol. 2, no. 2, Pp. 302-309.

[7] Efroymson M. A. *Multiple regression analysis*, New York: Ralston, Wiley, 1960.

[8] Rawlings J. O., Pantula S. G., Dickey D. A. *Appied Regression Analysis: A Research Tool*, New York: Springer-Verlag, 1998.

[9] Belsley D. A. *Conditioning Diagnostics: Collinearitu and Weak Data in Regression*, New York: John Wiley and Sons, 1991.

[10] Tarantola A. *Inverse Problem Theory*, SIAM, 2005.

[11] Johnstone I., Tibshirani R., Efron B., Hastie T. *Least Angle Regression*, 2004.

[12] McNames J. *Innovations in Local Modeling for Time Series Prediction*, 1999.

[13] Федорова В. П. *Локальные Методы Прогнозирования Временных Рядов*, 2009.

[14] В. Н. Солнцев, Д. Л. Данилов, А. А. Жиглявский. *Главные Компоненты Временных Рядов: Метод "Гусеница"*, С.-Петербургский государственный университет, 1997.

[15] Е. А. Крымова, В. В. Стрижов *Выбор моделей в линейном регрессионном анализе*, Информационные технологии, 2011.

[16] https://dmba.svn.sourceforge.net/svnroot/dmba/Data/GermanSpotPrice.csv

Получение устойчивых оценок гиперпараметров линейных регрессионных моделей*

А. А. Токмакова
aleksandra-tok@yandex.ru
Московский физико-технический институт

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: *байесовский вывод, ковариационная матрица, гиперпараметры модели, отбор признаков, регрессия.*

Введение

Часто при анализе временных рядов требуется рассмотрение большого количества признаков. В связи с этим возникают проблемы, связанные с наличием в выборке большого количества мультикоррелирующих признаков или с высокой зашумлённостью выборки. В работе выдвинута гипотеза о нормальном распределении вектора зависимой переменной и вектора параметров модели [1, 2]. Необходимо оценить ковариационные матрицы этих распределений и установить связь между пространством данных и пространством параметров, что позволит произвести отбор шумовых и коррелирующих признаков.

Развитие методов отбора признаков имеет богатую историю. Так начиная с 1960г., начали активно развиваться шаговые методы (Stepwise Regression) [3]. Главная идея этих методов состоит в отборе признаков, вносящих наибольший вклад в зависимую переменную. Вводится критерий, на основании которого алгоритм добавляет или удаляет признаки. Широкое применение получили частные случаи шаговой регрессии — алгоритмы LARS(Least Angle Regression) [4] и LASSO(Least Absolute Shrinkage and Selection Operator) [5]. Алгоритм LARS заключается в последовательном добавлении признаков. На каждом шаге веса признаков меняются таким образом, чтобы доставить наибольшую корреляцию с вектором регрессионных остатков. Алгоритм позволяет сократить количество свободных переменных и избежать проблемы неустойчивой оценки весов. Метод LASSO вводит ограничения на норму вектора коэффициентов модели, что приводит к обращению в ноль некоторых коэффициентов модели. Метод приводит к повышению устойчивости модели, позволяет отбирать признаки, оказывающие наибольшее влияние на вектор ответов.

Одной из причин возникновения задачи отбора признаков является их мультиколлинеарность. Первые шаги по решению этой проблемы были сделаны А. И. Тихоновым в 1963г., который ввел понятие регуляризации — дополнительного ограничения на задачу [6]. В работе [7] введено понятие регуляризации и описан общий метод решения задач. Так как работы А. И. Тихонова были опубликованы на западе только лишь в 1977г., в 1970г. Hoerl и Kennard предложили метод гребневой регрессии [8]. В минимизируемую

Научный руководитель В. В. Стрижов

функцию вводилось дополнительное слагаемое, что повышало устойчивость решения [9], однако не позволяло производить отбор признаков. Позднее стали появляться методы, использующие качественно иной подход для решения проблемы мультиколлинеарности. Например, Belsley предложил метод для удаления признаков [10], использующий сингулярное разложение матрицы плана. Алгоритм находит коэффициент, характеризующий степень зависимости признаков друг от друга. Позднее появился метод фактора инфляции дисперсии(Variance Inflation Factor) [11], оценивающий увеличение дисперсии заданного коэффициента регрессии, что свидетельствует о высокой корреляции данных.

При анализе временных рядов данные аппроксимируют какой-либо функцией (например, линейной), которую называют моделью. Вектор коэффициентов этой функции называется вектором параметров модели. Рассмотрим набор конкурирующих моделей, определяемых своим набором параметров. Априорная вероятность модели определена как вероятность появления модели, а апостериорная — вероятность появления модели при условии наличия конкретных данных. Таким образом, с помощью формулы Байеса [12, 13] получим связь между пространством данных и пространством параметров. Основываясь на гипотезе о нормальном распределении параметров модели [1], оценивается ковариационная матрица распределения параметров [2, 14]. На её главной диагонали стоят дисперсии случайных величин, что позволяет установить степень значимости данного конкретного параметра в модели. При таком подходе к отбору признаков не возникает необходимости разбиения выборки на обучение и контроль.

Постановка задачи

Дана регрессионная выборка: $D = \{\mathbf{x}_i, y_i\}_{i=1}^m = (X, \mathbf{y})$, где $\mathbf{x}_i \in \mathbb{R}^n, i = 1, \ldots, m$ — векторы независимой переменной, а $y_i \in \mathbb{R}, i = 1, \ldots, m$ — значения зависимой переменной. Решается задача восстановления регрессии

$$\mathbf{y} = \mathbf{f}(\mathbf{w}, X), \tag{1}$$

где $\mathbf{f}(\mathbf{w}, X)$ — некоторая параметрическая вектор-функция. Пусть многомерная случайная величина $\mathbf{y}$ имеет нормальное распределение:

$$\mathbf{y} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I_m),$$

где $\mathbf{f}$ — вектор-функция, σ^2 — дисперсия распределения, I_m — единичная матрица размерности m. Обозначим $\beta^{-1} = \sigma^2$.

Требуется приблизить функцию $\mathbf{f}(\mathbf{w}, X)$ параметрической функцией $\widehat{\mathbf{f}}(X, \mathbf{w})$ из заданного класса $\mathcal{F}$ (например, линейные функции), причем $|\mathcal{F}|$ конечно. Отображение $\mathbf{f} : \mathbb{R}^m \times \times \mathbb{W}^n \to \mathbb{R}^m$ будем называть моделью. Здесь $\mathbb{R}^m$ — пространство данных, а $\mathbb{W}^n \subseteq \mathbb{R}^n$ — пространство параметров. В задаче линейной регрессии задача приближения функции $\mathbf{f}(\mathbf{w}, X)$ эквивалентна задаче отбора признаков. В данном случае модель определяется параметрами, которые соответствуют множеству индексов активных признаков $\mathcal{A} \subseteq \mathcal{J}$, $\mathcal{J} = \{1, 2, ..., n\}$. Таким образом, при выборе модели требуется найти такое множество индексов $\mathcal{A}^*$, которое бы доставляло минимум функции:

$$\mathcal{A}^* = \arg\min_{\mathcal{A} \subseteq \mathcal{J}} S(\mathbf{f}_\mathcal{A} | \mathbf{w}^*, D),$$

где $S(\mathbf{f}|\mathbf{w}, D)$ — функция ошибки, заданная выражением 9, $\mathbf{f}_\mathcal{A}$ — параметрическая вектор-функция, вычисляемая только на множестве активных признаков, заданном индексами из множества $\mathcal{A}$. При этом параметры $\mathbf{w}^*$ модели должны доставлять минимум функции:

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbb{W}} S(\mathbf{w} | \mathbf{f}_\mathcal{A}, D).$$

Вид функции ошибки

Пользуясь предположением о том, что вектор зависимой переменной — многомерная случайная величина с нормальным распределением, запишем конкретный вид функции ошибки $S(\mathbf{w})$ для поставленной задачи.

Пусть многомерная случайная величина $\mathbf{y}$ имеет нормальное распределение $\mathbf{y} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I_m)$, где $\mathbf{f}$ — вектор-функция, σ^2 — дисперсия распределения, I_m — единичная матрица размерности m. Обозначим $\beta^{-1} = \sigma^2$. Тогда распределение зависимой переменной $\mathbf{y}$ можно представить в следующем виде:

$$p(\mathbf{y}) = (2\pi\beta^{-1})^{-\frac{m}{2}} \exp(-\frac{1}{2}(\mathbf{y}-\mathbf{f})^T \beta I(\mathbf{y}-\mathbf{f})). \tag{2}$$

Рассмотрим функцию правдоподобия данных, которая имеет вид

$$p(\mathbf{y}|X, \mathbf{w}, \beta, \mathbf{f}) \stackrel{\text{def}}{=} p(D|\mathbf{w}, \beta, \mathbf{f}) = \frac{\exp(-\beta E_D)}{Z_D(\beta)}. \tag{3}$$

Здесь E_D — функция ошибки. Из выражений (2) и (3), определим её как:

$$E_D = \frac{1}{2}(\mathbf{y}-\mathbf{f})^T(\mathbf{y}-\mathbf{f}).$$

Коэффициент Z_D нормирует функцию плотности нормального распределения и равен:

$$Z_D(\beta) = (2\pi\beta^{-1})^{\frac{m}{2}}. \tag{4}$$

Рассмотрим равенство (1). Слева стоит многомерная случайная величина $\mathbf{y}$, имеющая нормальное распределение. Матрица X не является случайной величиной, поэтому предположим, что $\mathbf{w} \in \mathbb{W}^n$ также является многомерной случайной величиной с нормальным распределением. Параметрами этого распределения будут математическое ожидание $\mathbf{w}_0$ и матрица ковариаций A^{-1}:

$$p(\mathbf{w}|A, \mathbf{f}) = \frac{\exp(-E_\mathbf{w})}{Z_\mathbf{w}(A)}. \tag{5}$$

Определим функцию-штраф за большое значение параметров модели для принятого распределения как $E_\mathbf{w} = \frac{1}{2}(\mathbf{w}-\mathbf{w}_0)^T A(\mathbf{w}-\mathbf{w}_0)$. Нормирующая константа $Z_\mathbf{w}$ в этом случае равна:

$$Z_\mathbf{w}(A) = (2\pi)^{\frac{n}{2}} |A^{-1}|^{\frac{1}{2}}. \tag{6}$$

Апостериорное распределение параметров модели для заданных A и β имеет вид:

$$p(\mathbf{w}|D, A, \beta, \mathbf{f}) = \frac{p(D|\mathbf{w}, \beta, \mathbf{f}) p(\mathbf{w}|A, \mathbf{f})}{p(D|A, \beta, \mathbf{f})}, \tag{7}$$

$$\frac{p(D|\mathbf{w}, \beta, \mathbf{f}) p(\mathbf{w}|A, \mathbf{f})}{p(D|A, \beta, \mathbf{f})} = \frac{\exp(-\beta E_D)\exp(-E_\mathbf{w})}{Z_D(\beta) Z_\mathbf{w}(A)} = \frac{\exp(-(\beta E_D + E_w))}{Z_D(\beta) Z_\mathbf{w}(A)}, \tag{8}$$

где

$p(\mathbf{w}|D, A, \beta, \mathbf{f})$ — апостериорное распределение параметров;
$p(D|\mathbf{w}, \beta, \mathbf{f})$ — функция правдоподобия данных;
$p(\mathbf{w}|A, \mathbf{f})$ — априорное распределение параметров;
$p(D|A, \beta, \mathbf{f})$ — функция правдоподобия модели.

Записывая функцию ошибки как

$$S = E_{\mathbf{w}} + \beta E_D = \frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T A(\mathbf{w} - \mathbf{w}_0) + \frac{1}{2}(\mathbf{y} - \mathbf{f})^T \beta I(\mathbf{y} - \mathbf{f}), \tag{9}$$

получим следующее выражение для апостериорного распределения параметров:

$$p(\mathbf{w}|D, A, \beta, \mathbf{f}) = \frac{\exp(-S(\mathbf{w}))}{Z_S(A, \beta)},$$

где $Z_S = Z_S(A, \beta)$ — нормирующий коэффициент. Оценка нормировочного коэффициента производится с помощью аппроксимации Лапласа.

Аппроксимация Лапласа

Аппроксимация Лапласа позволяет оценить нормировочный коэффициент для ненормированной плотности вероятности. Пусть задано ненормированное распределение $p^*(\mathbf{w})$. Требуется найти нормировочную константу:

$$Z = \int p^*(\mathbf{w}) d\mathbf{w},$$

при которой распределение $p(\mathbf{w}) = Z^{-1} p^*(\mathbf{w})$. Предположим, что $p^*(\mathbf{w})$ имеет максимум в точке $\mathbf{w}_0$, то есть

$$\left.\frac{dp(\mathbf{w})}{dw}\right|_{\mathbf{w}=\mathbf{w}_0} = 0.$$

Прологарифмируем и разложим $p^*(\mathbf{w})$ по Тейлору в окрестности $\mathbf{w}_0$:

$$\ln p^*(\mathbf{w}) = \ln p^*(\mathbf{w}_0) + 0 - \frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T A(\mathbf{w} - \mathbf{w}_0) + \ldots, \tag{10}$$

где матрица Гессе $A = [\alpha_{ij}]$ определена как:

$$\alpha_{ij} = -\left.\frac{\partial \ln p^*(\mathbf{w})}{\partial w_i \partial w_j}\right|_{\mathbf{w}=\mathbf{w}_0},$$

то есть $A = -\nabla^2 \ln p^*(\mathbf{w})|_{\mathbf{w}=\mathbf{w}_0}$, где ∇ — градиент функции.

Отбрасывая все члены выше квадратичного в разложении и беря экспоненту обеих частей выражения (10), получим:

$$p^*(\mathbf{w}) \approx p^*(\mathbf{w}_0) \exp\left(-\frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T A(\mathbf{w} - \mathbf{w}_0)\right).$$

Тогда нормальное распределение $\widehat{p}(\mathbf{w})$, приближающее нормированное распределение $p(\mathbf{w})$ имеет вид:

$$\widehat{p}(\mathbf{w}) = \mathcal{N}(\mathbf{w}_0, A^{-1}) = \frac{1}{(2\pi)^{\frac{n}{2}} |A^{-1}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T A(\mathbf{w} - \mathbf{w}_0)\right).$$

Следовательно, нормировочная константа имеет вид:

$$Z = p^*(\mathbf{w}_0) \frac{(2\pi)^{\frac{n}{2}}}{|A|^{\frac{1}{2}}}. \tag{11}$$

Оценка ковариационных матриц

Анализируя функцию ошибки $S(\mathbf{w})$, построим алгоритм, позволяющий выявлять шумовые и коррелирующие признаки.

Пусть нам известен локальный минимум $S(\mathbf{w})$, и он находится в точке $\mathbf{w}_0$. Рассмотрим матрицу Гессе функции ошибок $H = -\nabla\nabla S(\mathbf{w})|_{\mathbf{w}=\mathbf{w}_0}$. При появлении в выборке шумовых или коррелирующих признаков, происходит резкое возрастание некоторых элементов матрицы H и увеличивается неустойчивость задачи. Необходимо установить связь между компонентами матрицы Гессе и ковариационной матрицей параметров, для того чтобы произвести отбор активных параметров $\mathcal{A}$ и повысить устойчивость решения.

Рассмотрим ряд Тейлора второго порядка логарифма числителя (7):

$$-S(\mathbf{w}) \approx -S(\mathbf{w}_0) - \frac{1}{2}\Delta\mathbf{w}^T H \Delta\mathbf{w}, \tag{12}$$

где $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}_0$. В выражении (12) нет слагаемого первого порядка, так как предполагается, что точка $\mathbf{w}_0$ доставляет локальный минимум функции $S(\mathbf{w})$. Следовательно:

$$\left.\frac{\partial S(\mathbf{w})}{\partial w}\right|_{\mathbf{w}=\mathbf{w}_0} = 0.$$

Применяя экспоненту к обеим частям выражения (12) получим необходимое приближение:

$$\exp(-S(\mathbf{w})) \approx \exp(-S(\mathbf{w}_0))\exp(-\frac{1}{2}\Delta\mathbf{w}^T H \Delta\mathbf{w}). \tag{13}$$

При полученном приближении выражение (13) будет выглядеть следующим образом:

$$p(\mathbf{w}|D, A, \beta) \approx \frac{\exp(-S(\mathbf{w}_0))\exp(-\frac{1}{2}\Delta\mathbf{w}^T H \Delta\mathbf{w})}{Z_S(A, \beta)}, \tag{14}$$

где $Z_S(A, \beta)$ выступает в роли нормировочного коэффициента плотности вероятностного распределения. Оценка для коэффициента Z_S получена с помощью аппроксимации Лапласа (пояснения см. в главе 5):

$$Z_S = \frac{\exp(-S(\mathbf{w}_0))(2\pi)^{\frac{n}{2}}}{|H|^{\frac{1}{2}}}. \tag{15}$$

Подставив (15) в (14), получим оценку правдоподобия модели, на основании которой будем производить отбор оптимальных гиперпараметров модели

$$p(\mathbf{w}|D, A, \beta) = \frac{|H|^{\frac{1}{2}}\exp(-\frac{1}{2}\Delta\mathbf{w}^T H \Delta\mathbf{w})}{(2\pi)^{\frac{n}{2}}}. \tag{16}$$

Выражение (14) определяет выбор наиболее правдоподобной модели. Для нахождения гиперпараметров воспользуемся принципом максимума правдоподобия $p(D|A, \beta)$ относительно A и β. Запишем $p(D|A, \beta)$ в следующем виде:

$$p(D|A, \beta) = \int p(D|\mathbf{w}, A, \beta)p(\mathbf{w}|A)d\mathbf{w}. \tag{17}$$

Используя выражения (5) и (3) перепишем функцию правдоподобия в виде:

$$p(D|A,\beta) = \frac{1}{Z_{\mathbf{w}}(A)}\frac{1}{Z_D(\beta)}\int \exp(-S(\mathbf{w}))d\mathbf{w}. \tag{18}$$

Из соображений нормировки интеграл выражения (7) равен единице, то есть:

$$\int p(\mathbf{w}|D,\beta)d\mathbf{w} = \int \frac{\exp(-S(\mathbf{w}))}{Z_S(A,\beta)}d\mathbf{w} = 1.$$

Следовательно интеграл в правой части (18) в точности равен Z_S. Поэтому:

$$p(D|A,\beta) = \frac{1}{Z_{\mathbf{w}}(A)}\frac{1}{Z_D(\beta)}\exp(-S(\mathbf{w}_0))(2\pi)^{\frac{n}{2}}|H|^{-\frac{1}{2}}. \tag{19}$$

Подставив значение Z_w из (6) и Z_D из (4) в (19), получим:

$$p(D|A,\beta) = (2\pi)^{-\frac{n}{2}}|A^{-1}|^{-\frac{1}{2}}(2\pi)^{-\frac{m}{2}}(\beta^{-1})^{\frac{m}{2}}\exp(-S(\mathbf{w}_0))(2\pi)^{\frac{n}{2}}|H|^{-\frac{1}{2}}. \tag{20}$$

Получим оценку логарифма правдоподобия:

$$\ln p(D|A,\beta,\mathbf{f}) = -\frac{1}{2}\ln|A^{-1}| - \frac{m}{2}\ln 2\pi + \frac{m}{2}\ln\beta^{-1} - S(\mathbf{w}_0) - \frac{1}{2}\ln|H|. \tag{21}$$

Поочерёдно приравнивая частные производные по A и β выражения (21) к нулю, найдём максимум (21) по гиперпараметрам.

Пусть матрица A диагональна. Введем обозначение $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_n]^T$ для вектора, состоящего из элементов диагонали матрицы A. Представим гессиан в виде:

$$H = -\nabla\nabla S(\mathbf{w}) = -\nabla\nabla(\beta E_D + E_{\mathbf{w}}) = -\beta\nabla\nabla E_D - \nabla\nabla E_{\mathbf{w}} = H_D + H_{\mathbf{w}},$$

где H_D зависит от β, а $H_{\mathbf{w}}$ зависит от A. Так как $\nabla\nabla E_{w_i} = \nabla\nabla(\frac{1}{2}\alpha_i(w_i - w_{0i})^2) = \alpha_i$, то часть гессиана $H_{\mathbf{w}}$ диагональна. Докажем, что H_D также диагональная матрица. Для этого рассмотрим два случая:

1. если все признаки независимы, то матрица H_D будет диагональной, так как недиагональные элементы матрицы Гессе отражают степень зависимости измеряемых величин;
2. при наличии в выборке шумовых или коррелирующих признаков будет наблюдаться возрастание диагональных элементов матрицы (дисперсий признаков), в сравнении с которыми недиагональными элементами можно пренебречь. Таким образом получим, что и в этом случае матрицу H_D можно считать диагональной (на диагонали собственные числа).

Таким образом представим H_D в следующем виде: $H_D = \mathrm{diag}(h_1, ..., h_n)$. Для выявления связи между параметрами и гиперпараметрами модели рассмотрим выражение (21). Воспользуемся необходимым условием минимума и приравняем к нулю первые производные выражения (21) по α_i:

$$\frac{1}{\alpha_i} - (w_i - w_0)^2 - \frac{1}{\beta h_i + \alpha_i} = 0. \tag{22}$$

Данное уравнение имеет два корня. Однако один из них не имеет смысла, так как A^{-1} — диагональная ковариационная(положительно определённая) матрица, следовательно по

критерию Сильвестра (симметричная квадратная матрица является положительно определенной тогда и только тогда, когда все её главные миноры положительны) не имеет отрицательных компонент:

$$\alpha_i = \frac{1}{2}\lambda_i\left(\sqrt{1+\frac{4}{(w_i-w_0)^2\lambda_i}}-1\right), \tag{23}$$

где $\lambda_i = \beta h_i$.

Приравняв производную по β выражения (21), найдём оптимальное значение β:

$$\frac{m}{2\beta} - E_D - \frac{1}{2\beta}\gamma = 0,$$

где

$$\gamma = \sum_{j=1}^{W}\frac{\lambda_j}{\lambda_j+\alpha_j}.$$

Таким образом

$$\beta = \frac{m-\gamma}{2E_D}. \tag{24}$$

Выражения (23) и (24) не позволяют явно вычислить значения $\boldsymbol{\alpha}$ и β. Поэтому итерационный процесс организуется следующим образом. На каждом шаге вычисляем $\mathbf{w}$ (минимизируя функцию ошибки из выражения (9)), далее, используя полученное приближение, находим вектор гиперпараметров $\boldsymbol{\alpha}$, затем значение гиперпараметра β. Процедура продолжается до сходимости как параметров, так и гиперпараметров, то есть до сходимости функции правдоподобия модели $p(D|A,\beta,\mathbf{w})$.

При появлении шумовых или коррелирующих признаков происходит вырождение матрицы Гессе из-за возрастания диагональных элементов(большое значение дисперсии свидетельствует о неинформативности признака). Недиагональные элементы становятся настолько малы, что можно считать матрицу H_D диагональной. Поэтому необходимо принудительно занижать возрастающие диагональные элементы, тем самым производя отсеивание шумовых и коллерирующих признаков.

Псевдокод алгоритма оценки гиперпараметров регрессионной модели

Вход: вектор зависимой переменной $\mathbf{y}$, модель $\mathrm{mdl}(\mathbf{w}, X)$

$\mathbf{w}_0 = 0$;
$\mathbf{w} = 0$;
$A = \mathrm{diag}(n, 1)$;
$\beta = 1$;
для $k = 2, \ldots, MaxIterations$
  вычислить $A, \beta, \mathbf{w}$:
  $\mathbf{w} = \mathbf{FindParameters}(S(\mathbf{w}), A, \beta, \mathbf{w}, \mathbf{w}_0, \mathbf{y})$;
  для $j = 2, \ldots, MaxIterations$
    добиться сходимости A и β при данном векторе $\mathbf{w}$:
    $H = \mathbf{CalcHessian}(S(\mathbf{w}), A, \beta, \mathbf{w}, \mathbf{w}_0, \mathbf{y})$;
    если $\frac{\max(H)}{\min(H)} > 10^6$ **то**
      $idx = \mathrm{find}(\max(H))$; *// индекс строки/столбца(диагональный элемент) с max элементом*

занулить строку и столбец Гессиана, содержащие максимальный элемент;
выход;
$\lambda = \beta * \mathrm{diag}(H)$;
$A = \frac{1}{2}\lambda(\sqrt{1 + \frac{4}{(\mathbf{w}-\mathbf{w}_0)^2\lambda}} - 1)$;
если $idx \neq 0$ **то**
занулить соответствующие диагональные элементы матрицы A(необходимо для сходимости гиперпараметра α);
выход;
$\gamma = \sum \frac{\lambda_j}{\lambda_j+\alpha_j}$;
$\mathbf{f} = \mathrm{mdl}(\mathbf{w}, X)$;
$E_D = \frac{1}{2}(\mathbf{y} - \mathbf{f})^T(\mathbf{y} - \mathbf{f})$;
$\beta = \frac{(m-\gamma)}{2E_D}$;
если $\sum(\alpha_k - \alpha_{k-1})^2 < Convergency$ and $(\beta_k - \beta_{k-1})^2 < Convergency$; **то**
закончить выполнение цикла на текущей итерации;
выход;
если $j = MaxIterations$ **то**
вывести сообщение о величине ошибки и закончить выполнение программы;
выход;
если $\sum(w_k - w_{k-1})^2 < Convergency$ **то**
закончить выполнение программы;

ПРОЦЕДУРА `FindParameters`$(S(\mathbf{w}), A, \beta, \mathbf{w}, \mathbf{w}_0, \mathbf{y})$
пока не найден минимум функции $S(\mathbf{w})$ по $\mathbf{w}$
$\mathbf{f} = \mathrm{mdl}(\mathbf{w}, X)$;
$S(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T A(\mathbf{w} - \mathbf{w}_0) + \frac{1}{2}(\mathbf{y} - \mathbf{f})^T \beta I(\mathbf{y} - \mathbf{f})$;
выход;

return $\mathbf{w}$;

ПРОЦЕДУРА `CalcHessian`$(S(\mathbf{w}), A, \beta, \mathbf{w}, \mathbf{w}_0, \mathbf{y})$
$h = 10^{-6}$; *// шаг разностной схемы*
для $i = 1, ..., l$
для $j = 1, ..., l$
посчитать элемент матрицы Гессе:
$\mathbf{e}_i = 0$; *// вектор приращения*
$e_i(i) = 1$;
$\mathbf{e}_j = 0$;
$e_j(j) = 1$;
$H(i,j) = \frac{S(\mathbf{w}+(\mathbf{e}_i+\mathbf{e}_j)h) - S(\mathbf{w}+\mathbf{e}_i h) - S(\mathbf{w}+\mathbf{e}_j h) + S(\mathbf{w})}{h^2}$;
выход;
выход;

return H;

Алгоритмы отбора признаков

Приведем примеры ранее предложенных методов регуляризации, приводящих к повышению устойчивости решения и отбору признаков в задаче линейной регрессии [8, 5].

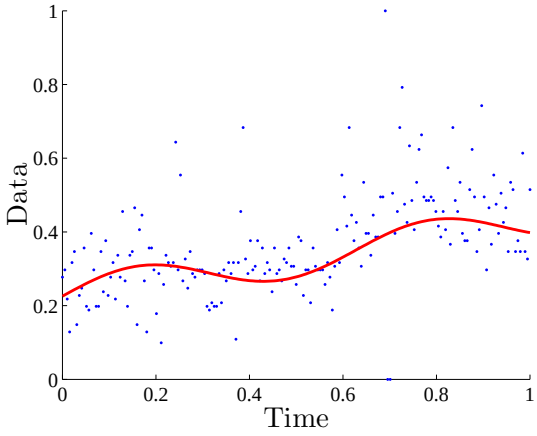


Рис. 1. Данные и аппроксимирующая модель

Гребневая регрессия

Запишем функцию ошибки для линейной модели вида (1):

$$Q(\mathbf{w}) = ||X\mathbf{w} - \mathbf{y}||^2.$$

Для минимизации функции воспользуемся необходимым условием минимума:

$$\frac{\partial Q}{\partial \mathbf{w}} = 2X^T(X\mathbf{w} - \mathbf{y}) = 0,$$

откуда следует, что $X^T X\mathbf{w} = X^T\mathbf{y}$. Если матрица $X^T X$ невырождена, то решением системы является вектор:

$$\mathbf{w}^* = (X^T X)^{-1} X^T\mathbf{y}.$$

Если ковариационная матрица $X^T X$ имеет неполный ранг, то её обращение невозможно. Также выделяют случай мультиколлинеарности: матрица $X^T X$ имеет полный ранг, но близка к некоторой матрице неполного ранга. В этом случае увеличивается разброс коэффициентов $\mathbf{w}^*$, появляются большие по абсолютной величине коэффициенты. Решение становится неустойчивым (небольшие изменения матрицы X ведут к большим изменениям величины $\mathbf{w}^*$).

Для решения проблемы мультиколлинеарности к функционалу Q добавляют регуляризатор, штрафующий большие значения нормы вектора $\mathbf{w}$: $Q_\tau = ||X\mathbf{w} - \mathbf{y}||^2 + \tau||\mathbf{w}||^2$. Решением полученной задачи является вектор:

$$\mathbf{w}^* = (X^T X + \tau I_m)^{-1} X^T\mathbf{y}.$$

Увеличение τ приводит к уменьшению нормы вектора $\mathbf{w}$, однако при этом ни один из параметров не обращается в ноль. То есть повышая устойчивость модели, гребневая регрессия не производит отбор признаков.

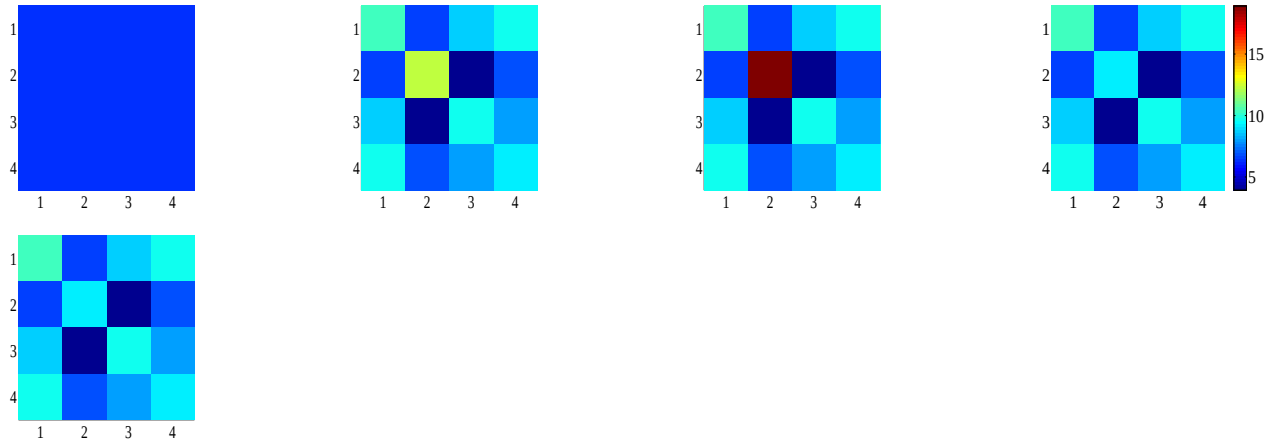


Рис. 2. Итерационный процесс для матрицы Гессе (случай шумового параметра)

Лассо Тибширани

В данном методе вместо добавления штрафного слагаемого к функционалу качества вводится ограничение-неравенство, запрещающее большие абсолютные значения коэффициентов:

$$\begin{cases} Q(\mathbf{w}) = ||X\mathbf{w} - \mathbf{y}||^2 \to \min\limits_{\mathbf{w}\in\mathbb{W}}, \\ \sum\limits_{j=1}^{W} |w_j| < \theta. \end{cases}$$

Чем меньше значение θ, тем больше коэффициентов w_j обнуляется, таким образом происходит исключение j-го признака. Недостатком этого метода относительно алгоритма, представленного в работе, является необходимость в разделении выборки на две части: для обучения и контроля. Также при использовании методов регуляризации возникает проблема выбора константы регуляризации. Для её вычисления обычно используют скользящий контроль, что значительно повышает трудоёмкость всей задачи в целом.

Вычислительный эксперимент

Результатом вычислительного эксперимента является отбор шумовых и коррелирующих признаков. Тестирование алгоритма производится на временном ряде продаж нарезного хлеба в зависимости от времени. Ряд содержит 195 записей. Модель, аппроксимирующая ряд: $\mathbf{y} = 0.2256 + 0.1996\boldsymbol{\xi} + 0.0496\sin(10\boldsymbol{\xi})$, где $\boldsymbol{\xi} \in \mathbb{R}^n$ — регрессионная выборка. Введем следующие обозначения: $\boldsymbol{\xi}^0$, $\boldsymbol{\xi}^1$ — значение каждого элемента выборки в нулевой и первой степени соответственно, $\sin(10\boldsymbol{\xi})$ — поэлементное применение элементарной функции к вектору $\boldsymbol{\xi}$. На рис. 1 представлена выборка и аппроксимирующая её модель. Пусть матрица плана X представлена в следующем виде $X = [\boldsymbol{\chi}_1, ..., \boldsymbol{\chi}_n]$, где $\boldsymbol{\chi} \in \mathbb{R}^m$. В данном случае она состоит из трёх столбцов: $\boldsymbol{\xi}^0$, $\boldsymbol{\xi}^1$, $\sin(10\boldsymbol{\xi})$.

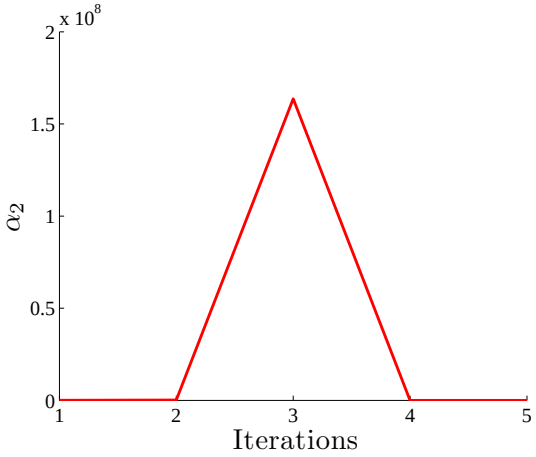


Рис. 3. Элемент матрицы A, соответствующий шумовому параметру модели

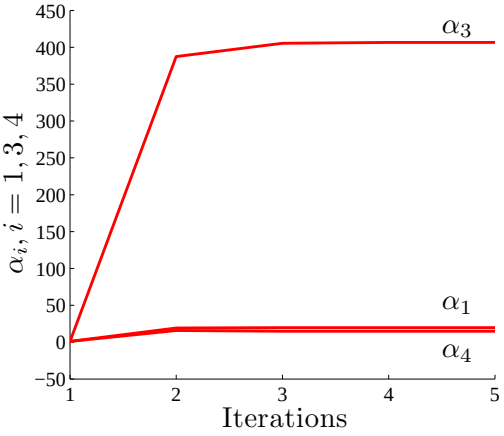


Рис. 4. Элементы матрицы A, соответствующие нешумовым параметрам модели

Отбор шумовых признаков

Шумовая выборка сформирована при помощи добавления столбца случайных чисел с нормальным распределением. Модель, аппроксимирующая данные в эксперименте: $\mathbf{y} = w_1\boldsymbol{\chi}_1 + w_2\boldsymbol{\chi}_2 + w_3\boldsymbol{\chi}_3 + w_4\boldsymbol{\chi}_4$, где $\boldsymbol{\chi}_1 = \boldsymbol{\xi}^0$, $\boldsymbol{\chi}_2 \sim \mathcal{N}(0, 2)$, $\boldsymbol{\chi}_3 = \boldsymbol{\xi}^1$, $\boldsymbol{\chi}_4 = \sin(10\boldsymbol{\xi})$. При наличии в выборке шумового элемента процедура сходится за восемь итераций. Ниже на рис. 2 проиллюстрированы изменения матрицы Гессе H на каждом шаге процедуры.

На 2-ой итерации наблюдается резкое отличие диагонального элемента $(2, 2)$. В течение итераций 2 и 3 он продолжает возрастать, пока не достигает критической относительной величины (принята эмпирическая оценка отношения максимального элемента матрицы к минимальному 10^6). Далее на 4-ой итерации выполняется его зануление. Таким образом происходит выявление шумового признака.

На рис. 3 и 4 представлены диагональные элементы матрицы A. Первый график иллюстрирует изменения второго диагонального элемента α_2, который соответствует шумовому параметру модели. Резкий скачок объясняется тем, что на данной итерации алгоритм находится вблизи локального минимума $\mathbf{w}_0$, и, несмотря на возрастание диагональных

элементов матрицы H, знаменатель формулы (23) мал. Далее происходит зануление элементов матрице Гессе, и соответствующий гиперпараметр α становится равным нулю.

На графиках 5 и 6 представлен скалярный гиперпараметр β и процесс изменения параметров модели w_i соответственно.

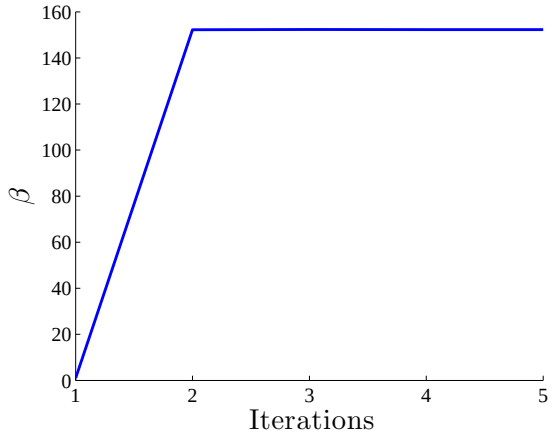


Рис. 5. Скалярный гиперпараметр β (случай шумового параметра)

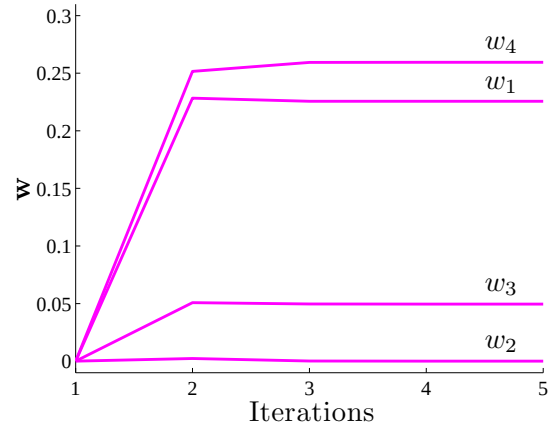


Рис. 6. Параметры модели **w** (случай шумового параметра)

Отбор коррелирующих признаков

Выборка с коррелирующими признаками сформирована при помощи добавления в матрицу плана столбца $1.3\boldsymbol{\chi}_2$. Таким образом, модель, аппроксимирующая данные в эксперименте: $y = w_1\boldsymbol{\chi}_1 + w_2\boldsymbol{\chi}_2 + w_3\boldsymbol{\chi}_3 + w_4\boldsymbol{\chi}_4$, где $\boldsymbol{\chi}_1 = \boldsymbol{\xi}^0$, $\boldsymbol{\chi}_2 = \boldsymbol{\xi}^1$, $\boldsymbol{\chi}_3 = 1.3\boldsymbol{\xi}^1$, $\boldsymbol{\chi}_4 = \sin(10\boldsymbol{\xi})$. Ниже на рис. 7 поэлементно проиллюстрирована матрица Гессе H.

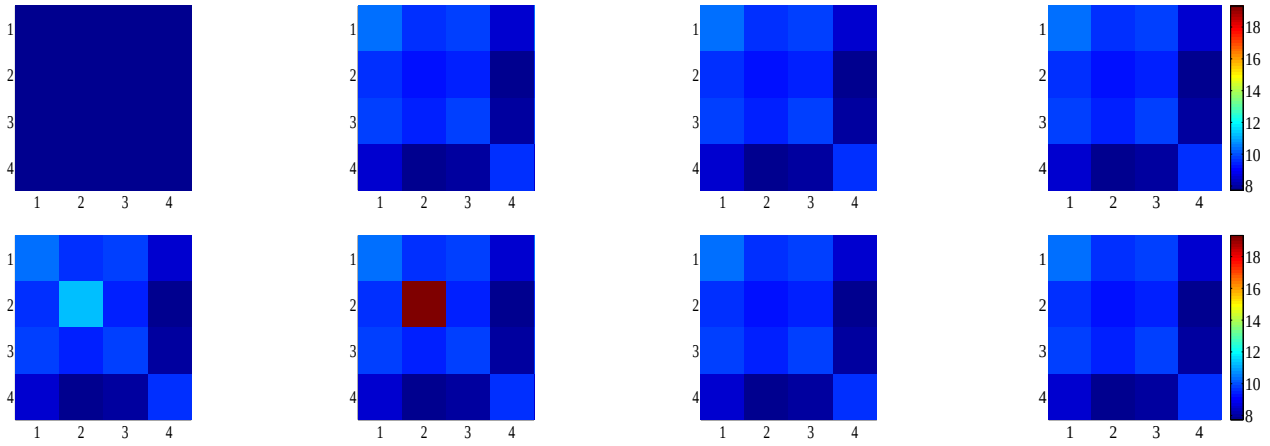


Рис. 7. Итерационный процесс для матрицы Гессе (случай коррелирующих параметров модели)

При наличии коррелирующих признаков также наблюдается возрастание диагональных элементов. Это происходит из-за того, что алгоритм выбирает ближайший вектор $\boldsymbol{\chi}$ к вектору $\mathbf{y}$ (в пространстве векторов матрицы X), а коррелирующий с ним считает шумовым.

На графиках 8 и 9 представлены диагональные элементы матрицы A.

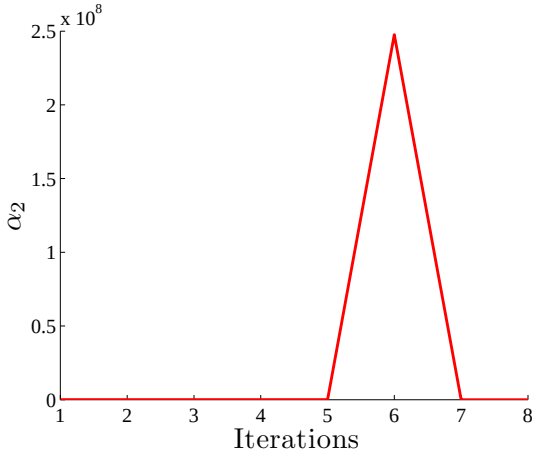


Рис. 8. Элементы матрицы A, соответствующие независимым параметрам модели

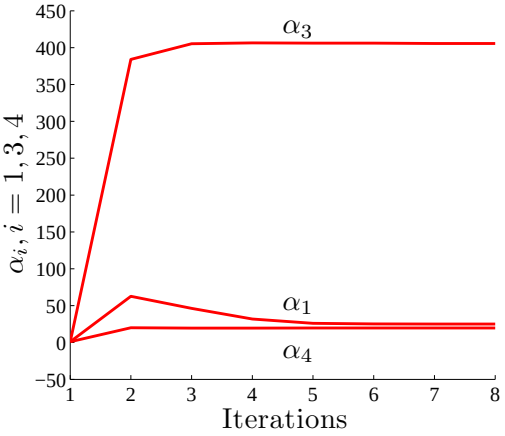


Рис. 9. Элемент матрицы A, соответствующий коррелирующему параметру модели

На рис. 10 представлены изменения скалярного гиперпараметры β. На рис. 11 представлены изменения параметров модели w_i в течении итерационного процесса. Коррелирующий параметр w_2 сначала возрастает, а затем стремится к нулю. Это происходит из-за того, что пространство параметров модели многоэкстремально.

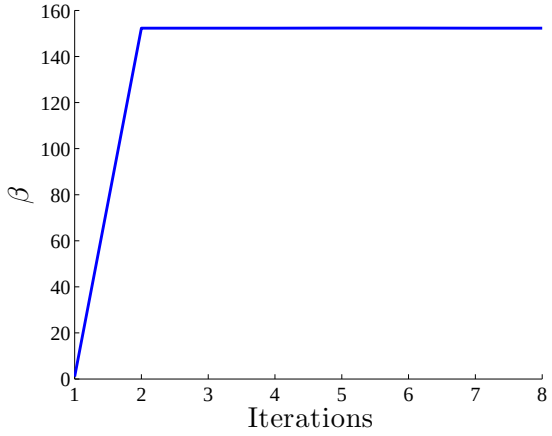


Рис. 10. Скалярный гиперпараметр β (случай зависимых параметров)

Заключение

В работе решена задача отсеивания шумовых и коррелирующих признаков, а также оценивалась ковариационная матрица параметров модели. При этом выбиралось некоторое множество активных признаков, которое доставляло минимум функции ошибки. Используя связанный байесовский вывод и предположение о нормальном распределении вектора зависимой переменной, была установлена связь между пространством данных и пространством параметров модели. На основании этой зависимости построен алгоритм, использующий оценку ковариационной матрицы параметров и позволяющий произвести отсев шумовых и коррелирующих признаков. Преимуществами данного алгоритма перед методами, описанными во введении, являются: 1) нет необходимости разделения данных на обучающую и контрольную выборку; 2) алгоритм не содержит никаких параметров, которые необходимо оценивать дополнительно (как, например, в методах регуляризации); 3)

добиваясь сходимости как параметров, так и гиперпараметров, предложенный алгоритм повышает устойчивость выбранной регрессионной модели.

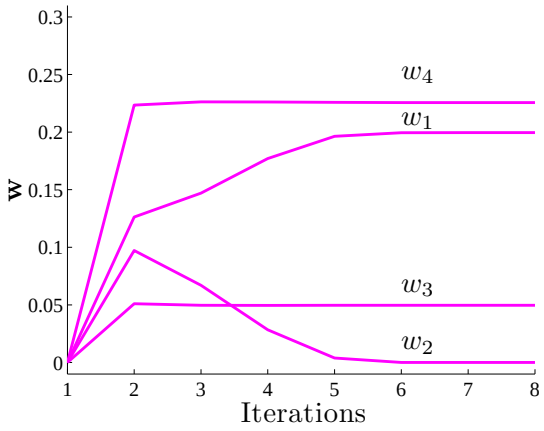


Рис. 11. Вектор параметров модели **w** (случай зависимых параметров)

Литература

[1] *Strijov V. V., Weber G.-W.* Nonlinear regression model generation using hyperparameter optimization // Computers and Mathematics with Applications, 2010, vol. 60, no. 4, pp. 981-988.

[2] *Стрижов В. В.* Поиск параметрической регрессионной модели в индуктивно заданном множестве // Вычислительные технологии, 2007, vol. 1, pp. 93-102.

[3] *Efroymson M. A.* Multiple regression analysis. – New York: Ralston, Wiley, 1960.

[4] *Efron B., Hastie T., Johnstone J., Tibshirani R.* Least Angle Regression // Annals of Statistics, 2004, vol. 32, no. 3, pp. 407-499.

[5] *Tibshirani R.* Regression shrinkage and Selection via the Lasso // Journal of the Royal Statistical Society, 1996, vol. 32, no. 1, pp. 267-288.

[6] *Ильин В. А.* О работах А. Н. Тихонова по методам решения некорректно поставленных задач // Успехи математичексих наук, 1997, vol. 1, pp. 168-175.

[7] *Тихонов А. Н.* Решение некорректно поставленных задач и метод регуляризации. – М.: ДАН, 1963, vol. 151, pp. 501-504.

[8] *Hoerl A. E., Kennard R. W.* Ridge regression: Biased estimation for nonortthogonal problems // Technometrics, 1970, vol. 3, no. 12, pp. 55-67.

[9] *Bjorkstrom A.* Ridge regression and inverse problems. Tech. rep.: Stockholm University. – Stockholm, 2001.

[10] *Belsley D. A.* Conditioning Diagnostics: Collinearity and Weak Data in Regression. New York: John Wiley and Sons, 1991.

[11] *Marquardt D. W.* Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation // Technometrics, 1996, vol. 12, no. 3, pp. 605-607.

[12] *MacKay D.* Laplace's Method /В кн.: Information Theory, Inference, and Learning Algotirhms. – Cambridge: Cambridge University Press, 2005, pp. 341-351.

[13] *Nabney I.* Bayesian Techniques /В кн.: Netlab: Algorithms for Pattern Recognition. – New York: Springer, 2002, pp. 325-366.

[14] *Стрижов В. В.* Методы выбора регрессионных моделей. – М.: ВЦ РАН, 2010.

[15] *Bishop C. M.* Linear models for classification / В кн.: Pattern Recognition and Machine Learning. Под ред.: M. Jordan, J. Kleinberg, B. Scholkopf. – New York: Springer Science+Business Media, 1960, pp. 213-216.

Уточнение ранговых экспертных оценок с использованием монотонной интерполяции

М. П. Кузнецов
mikhail.kuznecov@phystech.edu
Московский физико-технический институт

Описан способ построения интегральных индикаторов качества объектов с использованием экспертных оценок и измеряемых данных. Каждый объект описан набором признаков в линейных шкалах. Используются экспертные оценки качества объектов и важности признаков, которые корректируются в процессе вычисления. Предполагается, что оценки выставлены в ранговых шкалах. Рассматривается задача получения таких интегральных индикаторов, которые не противоречили бы экспертным оценкам. Предложено два подхода к уточнению экспертных оценок. При первом подходе вектор экспертных оценок рассматривается как выпуклый многогранный конус. Для уточнения экспертных оценок минимизируется расстояние между векторами в конусах. При втором подходе используется задача монотонной интерполяции с гиперпараметром. Проведен вычислительный эксперимент на следующих данных: экспертами оценивался фактор экологического воздействия на окружающую среду хорватских электростанций. Проведена процедура уточнения экспертных оценок.

Ключевые слова: *интегральный индикатор, экспертные оценки, монотонная интерполяция, ранговые шкалы.*

Введение.

При решении задач управления возникает необходимость дать каждому объекту оценку его качества. Интегральный индикатор — это число, поставленное в соответствие объекту, и рассматриваемое как оценка его качества. Интегральными индикаторами называется вектор оценок, поставленный в соответствие набору объектов.

При построении интегральных индикаторов выбирается критерий качества объектов. Формируется набор объектов, сравнимых в контексте выбранного критерия. Формируется набор показателей, которые эксперты считают необходимыми для описания этого критерия. Составляется матрица «объекты-признаки». Значения показателей приводятся к единой шкале и соответствуют принципу «чем больше, тем лучше»: большему значению показателя (при прочих равных) соответствует большее значение индикатора.

Ранее было предложено несколько подходов к построению интегральных индикаторов [1, 2, 3]. Подход «без учителя» заключается в нахождении интегральных индикаторов с помощью описаний объектов и выбранного метода их построения. Например, таковым является построение интегрального индикатора методом главных компонент, согласно которому интегральный индикатор является проекцией векторов-описаний объектов на первую главную компоненту матрицы «объекты-признаки» [4, 5].

Подход «с учителем» использует кроме описаний объектов экспертные оценки качества объектов или оценки важности показателей и заключается в нахождении компромисса между этими оценками и вычисленными индикаторами. Ранее был предложен подход, в котором восстанавливается регрессия описаний объектов на экспертные оценки качества объектов [6, 7].

Данная работа посвящена уточнению экспертных оценок, выставленных в ранговых шкалах. Для построения интегральных индикаторов принимается линейная модель: стро-

ится линейная комбинация признаков с их весами. Вектор весов признаков и начальный интегральный индикатор выставляются экспертами в ранговой шкале. В общем случае, построенный по вектору весов интегральный индикатор не совпадает с индикатором, заданным экспертами, то есть экспертные данные противоречат друг другу. Данная работа посвящена устранению разногласия в оценках экспертов.

В работе будут рассмотрены два метода. Первый метод развивает идеи, описанные в [6]. Метод заключается в следующем: ранговые экспертные оценки весов показателей задают выпуклый многогранный конус. Матрица «объекты-признаки» задает линейное отображение этого конуса из пространства показателей в пространство интегральных индикаторов. Полученный в результате отображения конус может пересекаться с конусом, заданным ранговыми экспертными оценками интегрального индикатора. В этом случае, экспертные оценки показателей и объектов считаются непротиворечивыми, и отыскивается наиболее устойчивый интегральный индикатор. В противном случае, выполняется процедура рангового уточнения оценок.

Второй метод состоит в решении задачи монотонной интерполяции [8, 9, 10]. В общем случае, задача монотонной интерполяции, или так называемая «isotonic regression», решает задачу наилучшего приближения произвольной последовательности точек размера n линейного пространства монотонной последовательностью точек пространства. Метод согласования экспертных данных заключается в том, что отыскивается вектор с монотонной последовательностью координат, наиболее близкий к заданному экспертами. Введенный в модель гиперпараметр отдает предпочтение экспертным оценкам индикаторов или оценкам весов признаков.

Предложенные алгоритмы используются для оценивания хорватских электростанций [11]. Данные являются матрицей «объекты-признаки» и заданными экспертами векторами оценок интегрального индикатора и весов признаков. Оценивается производительность электростанций.

Экспертные оценки, заданные в ранговых шкалах.

Задана матрица описаний объектов $X = \{x_{ij}\}_{i=1,j=1}^{m,n}$. Вектор $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{in})$ – описание $i-$го объекта.
Интегральный индикатор — линейная комбинация вида

$$y_i = \sum_{j=1}^{n} w_j g_j(x_{ij}),$$

где g_j — функция приведения показателей в единую шкалу, например:

$$g_j : x_{ij} \mapsto (-1)^{\zeta_j} \frac{x_{ij} - \min\limits_i x_{ij}}{\max\limits_i x_{ij} - \min\limits_i x_{ij}} + \zeta_j. \tag{1}$$

Параметр ζ_j назначается равным 1, если оптимальное значение показателя минимально, и 0 иначе. Если знаменатель дроби 1 равен нулю для некоторых значений индекса j , то соответствующий признак исключается из дальнейшего рассмотрения. Будем обозначать теперь за X приведенную таким способом матрицу «объекты-признаки». Таким образом,

$$\mathbf{y} = X\mathbf{w}.$$

Заданы в ранговых шкалах экспертные оценки: $\mathbf{y}_0, \mathbf{w}_0$, допускающие произвольные монотонные преобразования. Пусть на наборах экспертных оценок введено отношение порядка такое, что

$$y_1 \geqslant y_2 \geqslant ... \geqslant y_m \geqslant 0;\ w_1 \geqslant w_2 \geqslant ... \geqslant w_n \geqslant 0.$$

Множество всех таких векторов задается системой линейных неравенств

$$J\mathbf{y} \geqslant 0,$$

где

$$\underset{m\times m}{J} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Если же порядок $y_{i_1} \geqslant y_{i_2} \geqslant y_{i_m} \geqslant 0$ произвольный, то матрица системы будет получается из J перестановкой соответствующих столбцов.
Таким образом, заданным $\mathbf{y}_0$ и $\mathbf{w}_0$ можно поставить в соответствие матрицы J_m и J_n размеров соответственно $m \times m$ и $n \times n$.

Решение задачи согласования экспертных оценок с использованием конусов.

В этом параграфе опишем метод согласования экспертных оценок, предложенный в [6]. Дадим некоторые определения.

Определение 1. *Множество точек $\mathcal{Y}$ в $\mathbb{R}^m$ называется конусом, если для любой точки $y \in \mathcal{Y}$ точка λy также принадлежит $\mathcal{Y}$.*

Определение 2. *Выпуклый многогранный конус с вершиной в начале координат — это область решений системы однородных неравенств:*

$$\begin{cases} a_{11}w_1 & + & a_{12}w_2 & + & \ldots & + & a_{1n}w_n & \geqslant & 0, \\ a_{21}w_1 & + & a_{22}w_2 & + & \ldots & + & a_{2n}w_n & \geqslant & 0, \\ \ldots & & \ldots & & \ldots & & \ldots & & \ldots \\ a_{m1}w_1 & + & a_{m2}w_2 & + & \ldots & + & a_{mn}w_n & \geqslant & 0. \end{cases}$$

Эта система линейных неравенств задает в соответствующем пространстве выпуклый многогранный конус. Соответствуя данному определению, определим $\mathcal{Y}$ — конус, задаваемый матрицей J_m в пространстве интегральных индикаторов; $\mathcal{W}$ — конус, задаваемый матрицей J_n в пространстве весов признаков. Эти конусы характеризуются тем, что векторы внутри каждого из них имеют одинаковый ранговый порядок.
Поскольку A — линейное преобразование, оно переводит конус $\mathcal{W}$ в конус $A\mathcal{W}$, который лежит в пространстве интегральных индикаторов.

Задача 1. Требуется найти в конусах $\mathcal{W}$ и $\mathcal{Y}$ векторы $\mathbf{w}$ и $\mathbf{y}$, такие, что:

$$(\mathbf{y}_1, \mathbf{w}_1) = \min_{\mathbf{y}\in\mathcal{Y},\ \mathbf{w}\in\mathcal{W}} \|\mathbf{y} - A\mathbf{w}\|,$$

$$\text{при } \|A\mathbf{w}\| = 1, \|\mathbf{y}\| = 1,$$

где $\|.\|$ — евклидова метрика в пространстве $\mathbb{R}^m$.

Таким образом, отыскивается вектор весов $\mathbf{w}_1$, элементы которого имеют такой же ранговый порядок, что и $\mathbf{w}_0$. При этом приведенный в ранговую шкалу индикатор $A\mathbf{w}_1$ является ближайшим к $\mathbf{y}_0$.

В случае непустого пересечения конусов $\mathcal{Y}$ и $A\mathcal{W}$ решение задачи (1) дает вектор $\mathbf{y}$, который лежит в пересечении этих конусов. Если пересечение — пустое, предлагается найти ближайшие друг к другу лучи на ребрах или гранях конусов.

Отыскиваемая пара $(\mathbf{y}_1, \mathbf{w}_1)$ должна выполнять следующие условия:

$$\begin{array}{lll} \text{minimize} & \|\mathbf{y} - A\mathbf{w}\| & \\ \text{subject to} & \mathbf{y}^T\mathbf{y} = 1, & (A\mathbf{w})^T A\mathbf{w} = 1, \\ & J_n\mathbf{w} \geqslant \mathbf{0}, & J_m\mathbf{y} \geqslant \mathbf{0}. \end{array}$$

Постановка задачи согласования экспертных оценок с использованием монотонной интерполяции.

В данном параграфе рассмотрим новый метод согласования экспертных оценок. Пусть $\mathbf{y}_0$ — заданное экспертами начальное приближение вектора $\mathbf{y}$. Вектор, наиболее близкий в пространстве весов признаков к $\mathbf{y}_0$, в смысле наименьших квадратов:

$$\widetilde{\mathbf{w}} = X^+\mathbf{y}_0, \text{ где}$$

$$X^+ = (X^T X)^{-1} X^T.$$

Задача 2. Требуется найти такую монотонную последовательность $w_1 \leqslant \ldots \leqslant w_n$, что она лучше всего приближает вектор $\widetilde{\mathbf{w}}$ в смысле среднего квадрата ошибки:

$$\begin{cases} \widehat{\mathbf{w}} = \arg\min\limits_{w\in\mathbb{R}^n} \sum\limits_{j=1}^{n} (\widetilde{w}_j - w_j)^2, \\ w_1 \leqslant \ldots \leqslant w_n. \end{cases}$$

Такую задачу можно решить, например, методом, описанным в [8]. Однако, чтобы получить согласованные экспертные оценки, введем в модель гиперпараметр. С его помощью мы сможем варьировать нашу «степень доверия» от экспертных оценок весов признаков (то есть, монотонной последовательности $w_1 \leqslant \ldots \leqslant w_n$) к экспертным оценкам интегральных индикаторов (вектору $\widehat{\mathbf{w}}$).

Задача 3. Требуется найти такой вектор $\widehat{\mathbf{w}}$, что:

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \left(\frac{1}{2} \sum_{j=1}^{n} (\widetilde{w}_j - w_j)^2 + \lambda \sum_{j=1}^{n-1} (w_j - w_{j+1})_+ \right). \qquad (2)$$

Решение задачи монотонной интерполяции с гиперпараметром.

Для решения этой задачи воспользуемся идеями, описанными в [10].

Утверждение 1. *Пусть, для некоторого λ_0, совпадают две соседние координаты оценки: $\widehat{w}_j(\lambda_0) = \widehat{w}_{j+1}(\lambda_0)$. Тогда $\widehat{w}_j(\lambda) = \widehat{w}_{j+1}(\lambda)$ для всех $\lambda > \lambda_0$.*

Пусть при некотором λ совпадают некоторые соседние координаты вектора $\mathbf{w}$, и всего таких множеств совпадающих координат — K_λ. Обозначим за $A_1, ..., A_{K_\lambda}$ сами эти множества. Заметим, что $A_1 \cup ... \cup A_{K_\lambda} = \{1, ..., n\}$. Тогда функция потерь для задачи (2) перепишется в виде

$$\frac{1}{2}\sum_{k=1}^{K_\lambda}\sum_{l\in A_k}(\widetilde{w}_l - w_{A_k})^2 + \lambda\sum_{k=1}^{K_\lambda}(w_{A_k} - w_{A_{k+1}})_+.$$

Продифференцируем ее по всем w_{A_k}:

$$-\sum_{l\in A_k}\widetilde{w}_l + |A_k|\widehat{w}_{A_k}(\lambda) + \lambda(s_k - s_{k-1}) = 0$$

$$\text{для } k = 1, ..., K_\lambda,$$

где $s_k = 1$ при $\widehat{w}_{A_k}(\lambda) - \widehat{w}_{A_{k+1}}(\lambda) > 0$, и $s_k = 0$ иначе.

Пусть все $A_1, ..., A_{K_\lambda}$ не изменяются с увеличением λ. Тогда:

$$\frac{d\widehat{w}_{A_k}(\lambda)}{d\lambda} = \frac{s_{k-1} - s_k}{|A_k|}.$$

Когда λ увеличивается, множества A_k меняются. Однако, согласно утв. 1, они могут только объединяться, то есть, величины компонент $\widehat{w}_{A_k}(\lambda)$ внутри каждого множества A_k остаются равными. Можно посчитать величину следующего λ, при котором будут объединяться множества A_k, A_{k+1}. Обозначим это λ как $t_{k,k+1}$.

Утверждение 2. *Множества A_k и A_{k+1} будут объединяться при*

$$t_{k,k+1} = \frac{\widehat{w}_{A_{k+1}}(\lambda) - \widehat{w}_{A_k}(\lambda)}{D_k - D_{k+1}} + \lambda,$$

для всех $k = 1, .., K_\lambda - 1$, где

$$D_k = \frac{d\widehat{w}_{A_k}(\lambda)}{d\lambda}.$$

Доказательство. Поскольку производные

$$\frac{d\widehat{w}_{A_k}(\lambda)}{d\lambda}$$

не являются функциями λ, можно записать следующую систему уравнений:

$$\begin{cases}\widehat{w}_{A_k}(\lambda) = \lambda D_k + C_k,\\ \widehat{w}_{A_{k+1}}(\lambda,) = \lambda D_{k+1} + C_{k+1}.\end{cases}$$

В точке $t_{k,k+1}$ происходит объединение множеств A_k и A_{k+1}, то есть:

$$\widehat{w}_{A_k}(t_{k,k+1}) = \widehat{w}_{A_{k+1}}(t_{k,k+1}) \;\Rightarrow\; t_{k,k+1} = \frac{C_{k+1} - C_k}{D_k - D_{k+1}} =$$

$$= \frac{(\widehat{w}_{A_{k+1}}(\lambda) - \lambda D_{k+1}) - (\widehat{w}_{A_k}(\lambda) - \lambda D_k)}{D_k - D_{k+1}} =$$

$$= \frac{\widehat{w}_{A_{k+1}}(\lambda) - \widehat{w}_{A_k}(\lambda)}{D_k - D_{k+1}} + \lambda.$$

Таким образом, на каждой итерации нужно вычислять величину

$$\widehat{\lambda} = \min_{k: t_{k,k+1} > \lambda} t_{k,k+1}$$

и объединять множества $A_{k'}$ и $A_{k'+1}$, где

$$k' = \arg \min_{k: t_{k,k+1} > \lambda} t_{k,k+1}. \quad (3)$$

Алгоритм решения.

Вход: $\lambda = 0, K_\lambda = n, A_k = \{k\}, \widehat{w}_{A_k}(\lambda) = \widetilde{w}_k$.
1: **Повторять:**
2: $D_k := \frac{s_{k-1} - s_k}{|A_k|}$
3: $t_{k,k+1} := \frac{\widehat{w}_{A_{k+1}}(\lambda) - \widehat{w}_{A_k}(\lambda)}{D_k - D_{k+1}} + \lambda$
4: $\widehat{\lambda} := \min\limits_{k: t_{k,k+1} > \lambda} t_{k,k+1}$
5: $\widehat{w}_{A_k}(\lambda) := \widehat{w}_{A_k}(\lambda) + D_k(\widehat{\lambda} - \lambda)$
6: объединить $A_{k'}$ и $A_{k'+1}$, см. (3)
7: $\lambda := \widehat{\lambda}$
8: **пока** существует k: $t_{k,k+1} \geqslant \lambda$

Результат работы алгоритма монотонной интерполяции.

Проиллюстрируем работу алгоритма решения задачи монотонной интерполяции на модельной выборке, порожденной с помощью функции $y_i = x_i + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 20)$. Ломаная линия на рис. 1 — восстановленная зависимость, для различных значений регуляризатора λ.

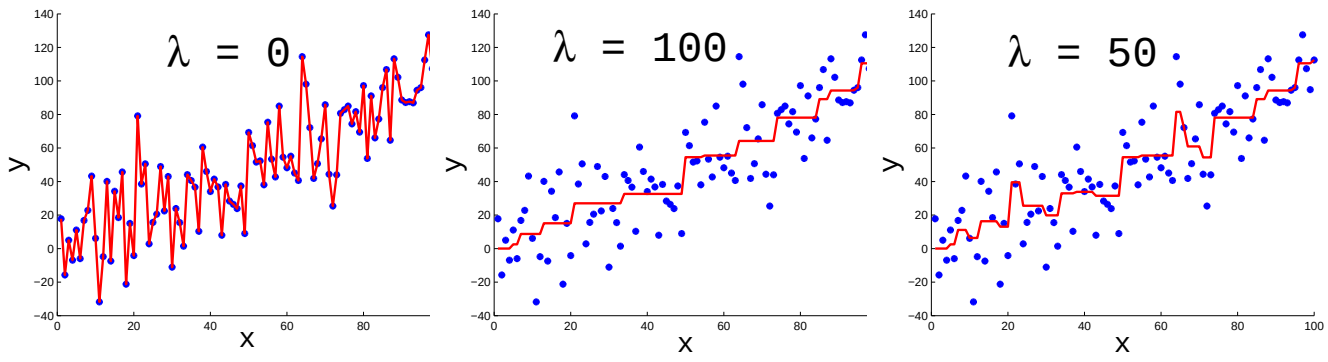


Рис. 1. Монотонная интерполяция

Видно, что при $\lambda = 100$ и более функция, восстанавливающая зависимость, монотонная. При $\lambda = 0$, наоборот, никакой монотонной коррекции нет.

Вычислительный эксперимент.

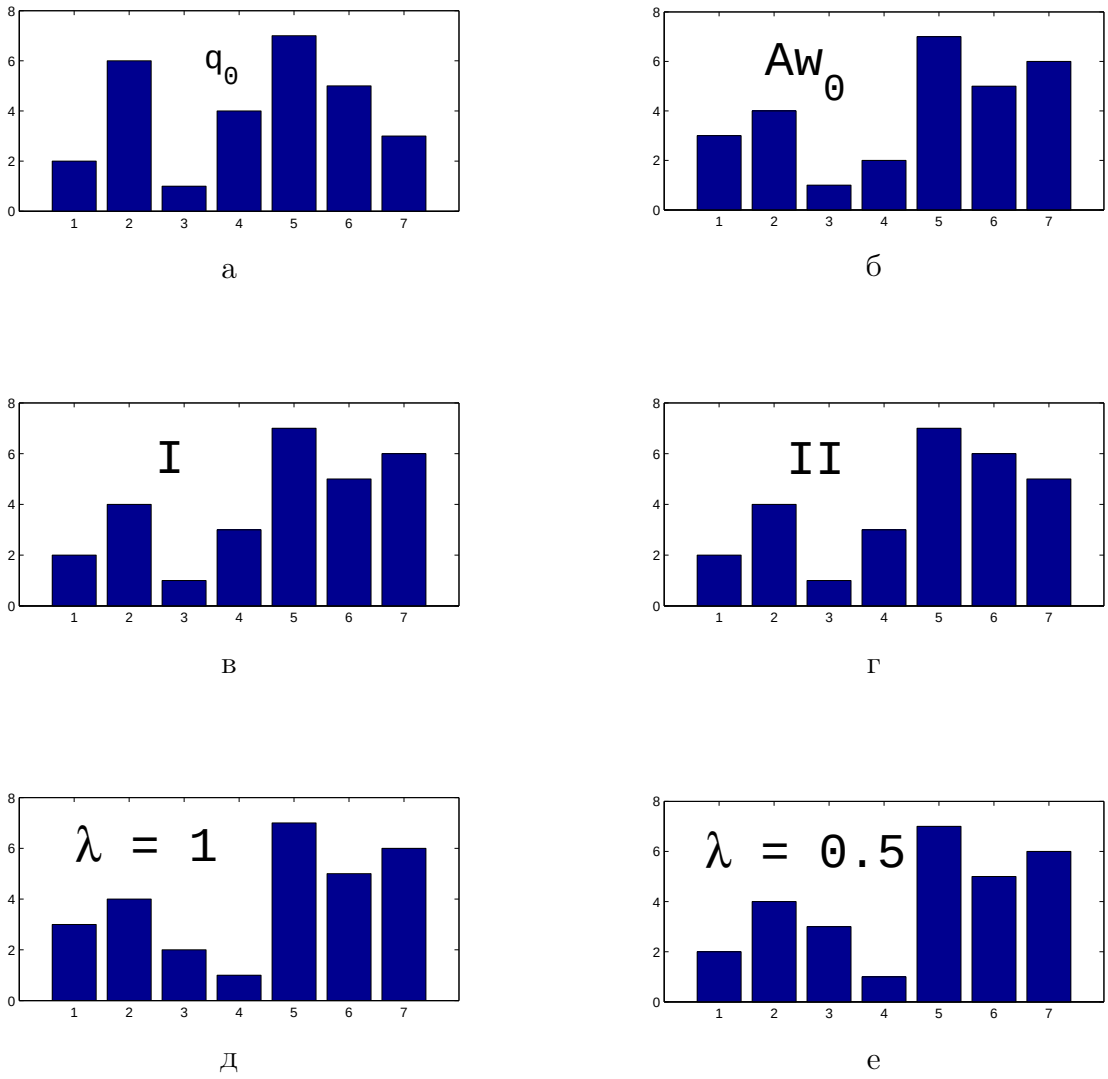


Рис. 2. Интегральные индикаторы электростанций, вычисленные различными алгоритмами. «а»: начальный интегральный индикатор $\mathbf{q}_0$. «б»: интегральный индикатор, построенный по $\mathbf{w}_0$. «в»: интегральный индикатор, построенный алгоритмом минимизации расстояния между векторами в конусах. «г»: интегральный индикатор, построенный алгоритмом максимизации корреляции между векторами в конусах. «д»: интегральный индикатор, построенный алгоритмом монотонной интерполяции со значением гиперпараметра $\lambda = 1$. «е»: интегральный индикатор, построенный алгоритмом монотонной интерполяции со значением гиперпараметра $\lambda = 0.5$.

Был проведен вычислительный эксперимент уточнения экспертных оценок экологического воздействия на окружающую среду хорватских электростанций. Для этого были собраны следующие данные: матрица «объекты-признаки», где объекты — это семь

N	Power Plant	Available net capacity (MW)	Electricity (GWh)	Heat (TJ)	SO_2 (t)	NO_x (t)	Particles (t)
1	Plomin 1 TPP	98	452	0	1950	1378	140
2	Plomin 2 TPP	192	1576	0	581	1434	60
3	Rijeka TPP	303	825	0	6392	1240	171
4	Sisak TPP	396	741	0	3592	1049	255
5	TE-TO Zagreb CHP	337	1374	481	2829	705	25
6	EL-TO Zagreb CHP	90	333	332	1259	900	19
7	TE-TO Osijek CHP	42	114	115	1062	320	35
	Optimal value	max	max	max	min	min	min

Рис. 3. Электростанции

электростанций, описываемых 11-ю признаками, экспертные оценки весов показателей и интегральных индикаторов электростанций. На рис. 3 показана часть этих данных: семь электростанций и шесть из 11 признаков.

Несмотря на то, что экспертные оценки не являются согласованными (рис. 2а и рис. 2б), по некоторым объектам можно выявить схожесть предпочтений. Например, и на рис. 1а, и на рис. 1б лучшим является объект 5; объект 6 всегда лучше объектов 1, 3 и 4; объект 1 всегда хуже объектов 2, 5, 6 и 7. Предложенные алгоритмы работают корректно, в том смысле, что они оставляют согласованными предпочтения экспертов: на всех рис. 2в, рис. 2г, рис. 2д , рис. 2е выполнены вышеописанные утверждения.

Из рис. 2в и рис. 2г видно, что алгоритмы поиска ближайших векторов в конусах сработали похожим образом.

На рис. 2д изображены интегральные индикаторы для $\lambda = 1$, то есть, когда в задаче 2 мы отдаем предпочтение экспертным оценкам весов. Видно, что интегральные индикаторы на рис. 2д похожи, соответственно, на интегральные индикаторы на рис. 2б.

Заключение.

В работе рассматривалась задача получения согласованных оценок качества объектов и важности показателей. В результате выполнения работы обобщены ранее полученные результаты по согласованию экспертных оценок с использованием конусов. Предложено использовать алгоритм монотонной интерполяции для уточнения экспертных оценок. Исследованы свойства этого алгоритма при различном значении гиперпараметра, введенного в модель. Проведен вычислительный эксперимент уточнения экспертных оценок качества хорватских электростанций, составлен рейтинг электростанций, основанный на оценках экспертов и измеряемых данных.

Литература

[1] *В. В. Подиновский* Многокритериальные задачи с упорядоченными по важности критериями. // Автоматика и телемеханика, стр. 118–127, 1976.

[2] *О. И. Ларичев, Е. М. Мошкович.* Качественные методы принятия решений. // Физматлит, 1996.

[3] *D. W. Marquardt.* Generalized inverses, ridge regression, biased linear estimation,and nonlinear estimation. // Technometrics, page 605–607, 1996.

[4] *I. T. Jolliffe.* Principal Component Analysis. // Springer, 2002.

[5] *A. J. Isenmann.* Modern multivariate statistical techniques. // Springer, 2008.

[6] *В. В. Стрижов.* Уточнение экспертных оценок с помощью измеряемых данных. // Заводская лаборатория. Диагностика материалов., page 59–64, 2006.

[7] *V. Strijov, G. Granić et al.* Integral indicator of ecological impact of the Croatian thermal power plants. // *Energy* doi:10.1016/j.energy.2011.04.30. — 2011.

[8] *J. de Leeuw, K. Hornik, P. Mair* Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods. // Jounal of Statistic Software. — 2009. — Vol. 29.

[9] *R. E. Barlow, H. D. Brunk* The Isotonic Regression Problem and Its Dual. // Journal of American Statistical Association. — 1972. — Vol. 67, — Pp. 140-147.

[10] *R. J. Tibshirani, H. Hoefling, R. Tibshirani* Nearly-Isotonic Regression. // Technometrics. — 2011. — Vol. 53.

[11] *R. Kos, Z. Krisic, T. Tarnik* Hrvatska elektoprivreda and the environment 2005-2006. // Zagreb, Hrvatska Elektroprivreda. — 2008.

Исследование устойчивости оценок ковариационной матрицы признаков*

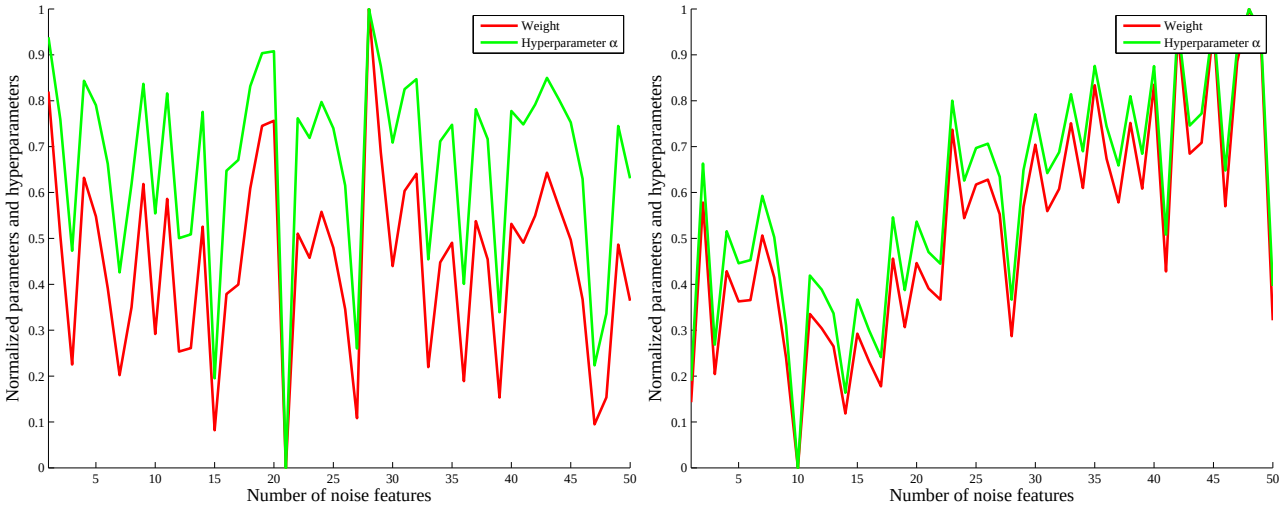
А. А. Зайцев
alexey.zaytsev@datadvance.net
Московский физико-технический институт, ФУПМ, каф. «Интеллектуальные системы»

В данной работе исследуется устойчивость оценок ковариационной матрицы параметров модели. Рассматриваются модели линейной и существенно нелинейной регрессии. Тогда вектор параметров модели соответствует набору признаков модели. Ковариационная матрица параметров строится в предположении о вероятностном распределении вектора параметров. Исследуется, зависит ли оценка ковариационной матрицы признаков от того, являются ли признаки мультикоррелирующими и шумовыми. Для такой матрицы плана получаем расширенный вектор параметров модели и оценку матрицы ковариации параметров модели. Сравнивается ковариационная матрица для нерасширенного и расширенного вектора параметров модели. Исследуется пространство параметров для информативных признаков. Эксперименты проводятся на реальных и модельных данных.

Ключевые слова: регрессионный анализ, линейная регрессия, символьная регрессия, оценка гиперпараметров.

Введение

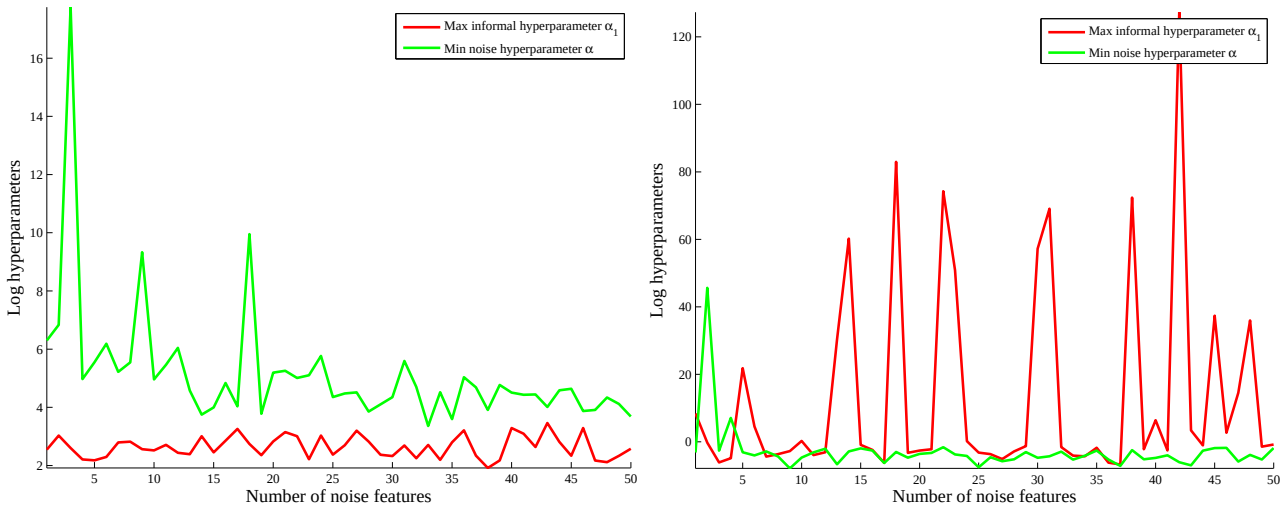
В данной работе рассматривается алгоритм выбора модели и настройки параметров модели линейной и существенно нелинейной регрессии, описанный в работе [1] для линейной и в работе [3] для существенно нелинейной регрессии.



(a) Линейная функция (b) Экспонента

Рис. 1. Зависимость параметров и гиперпараметров от числа шумовых признаков

Научный руководитель В. В. Стрижов



(a) Линейная функция (b) Экспонента

Рис. 2. Гиперпараметры для шумовых и информативного признака

Постановка задачи

Задана выборка $D = (X, \mathbf{y}) = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$. Вектор свободных переменных $\mathbf{x} \in \mathbb{R}^n$, зависимая переменная $y \in \mathbb{R}$. Предполагается, что

$$y = f(\mathbf{x}, \mathbf{w}) + \varepsilon, \tag{1}$$

где $f(\mathbf{x}, \mathbf{w})$ — некоторая параметрическая функция, $\mathbf{w} \in W$ — вектор ее параметров, ε — ошибка, распределенная нормально с нулевым математическим ожиданием и дисперсией β, $\varepsilon \sim \mathcal{N}(0, \beta)$. Предполагается, что вектор параметров $\mathbf{w}$ — распределенный нормально случайный вектор с нулевым математическим ожиданием и матрицей ковариаций A.

Рассматривается класс линейных функций $f(\mathbf{x}, \mathbf{w})$. Наиболее вероятные параметры $\mathbf{w}_{MP}$ имеют вид:

$$\mathbf{w}_{MP} = \arg\max_{\mathbf{w}} p(\mathbf{w}|D, A, \beta, f). \tag{2}$$

Для такого набора параметров исследуется матрица ковариации A, который мы тоже оцениваем, используя принцип максимального правдоподобия.

Описание алгоритма оценки матрицы ковариации

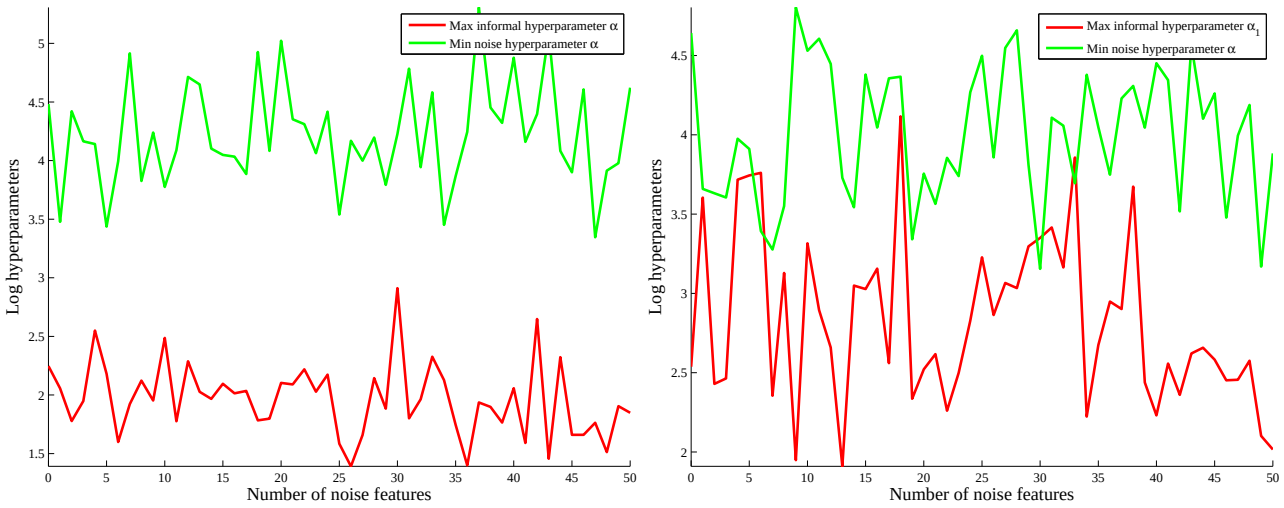
Для фиксированных гиперпараметров A, β вектор наиболее вероятных параметров минимизирует функционал

$$S(\mathbf{w}) = \mathbf{w}^T A \mathbf{w} + \beta \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \mathbf{w})^2 = E_{\mathbf{w}} + \beta E_D. \tag{3}$$

Набор наиболее вероятных гиперпараметров будем искать, максимизируя оценку правдоподобия по A, β

$$\ln p(D|A, \beta, f) = -\frac{1}{2} \ln |A| - \frac{m}{2} \ln 2\pi + \frac{m}{2} \ln \beta \underbrace{-E_{\mathbf{w}} - \beta E_D}_{S(\mathbf{w}_0)} - \frac{1}{2} \ln |H|, \tag{4}$$

здесь H — гессиан функционала (3).



(a) Линейная функция (b) Синусоида

Рис. 3. Гиперпараметры для шумовых и информативных признаков

В предположении о диагональности матрицы $A = diag(\boldsymbol{\alpha})$ и гессиана $H = diag(\mathbf{h})$, $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^m$, $\mathbf{h} = \{h_i\}_{i=1}^m$, приравняв производные по гиперпараметрам к нулю, получаем оценку для α_i

$$\alpha_i = \frac{1}{2}\lambda_i \left(\sqrt{1 + \frac{4}{w_i^2 \lambda_i}} - 1 \right), \tag{5}$$

здесь $\lambda_i = \beta h_i$.

Так же получаем оценку β

$$\beta = \frac{n - \gamma}{2E_D}, \tag{6}$$

здесь

$$\gamma = \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_j + \alpha_j}.$$

Используя оценки вектора параметров при фиксированных гиперпараметрах и гиперпараметров при фиксированных параметрах, выпишем итерационный алгоритм поиска наиболее вероятных параметров и гиперпараметров. Он состоит из шагов:

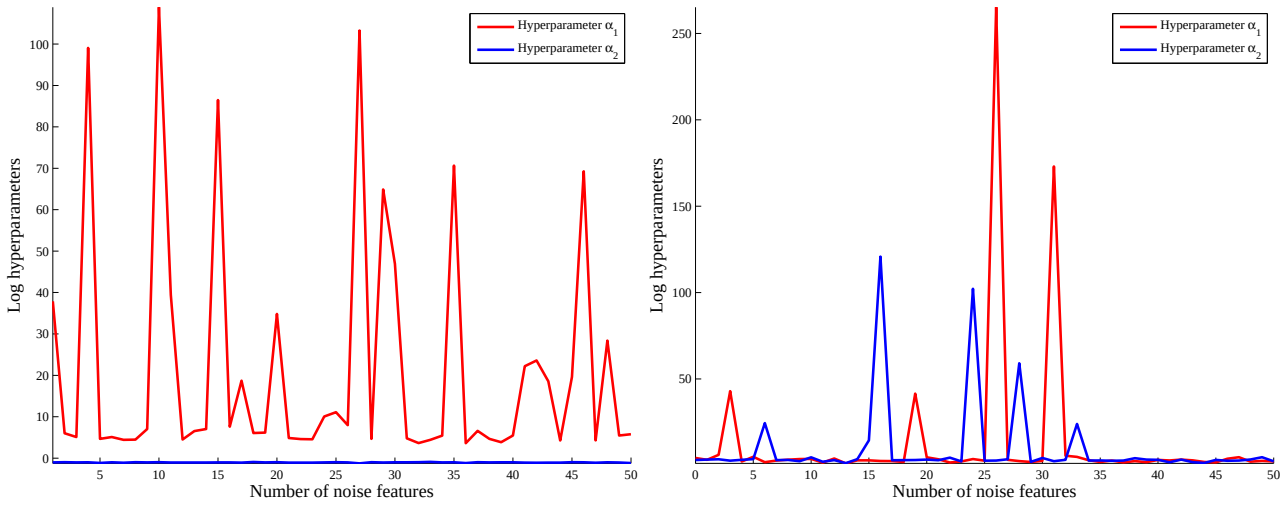
— поиск вектора параметров, максимизирующих (3),
— поиск гиперпараметров, максимизирующих правдоподобие (4),
— проверка критерия остановки.

Критерий остановки — малое изменение функционала (3) для двух последовательных итераций алгоритма.

Вычислительный эксперимент: шумовые признаки

В вычислительном эксперименте исследовалась устойчивость оценок гиперпараметров при добавлении шумовых и мультиколлиенарных признаков для линейной и существенно нелинейной регрессии.

Шумовые признаки:один признак. В выборках один информативный признак и n' шумовых. Вектор свободных переменных для каждого объекта генерируется из нормального распределения с нулевым математическим ожиданием и единичной дисперсией. Рас-



(a) Линейная функция (b) Синус

Рис. 4. Гиперпараметры для шумовых и информативных признаков

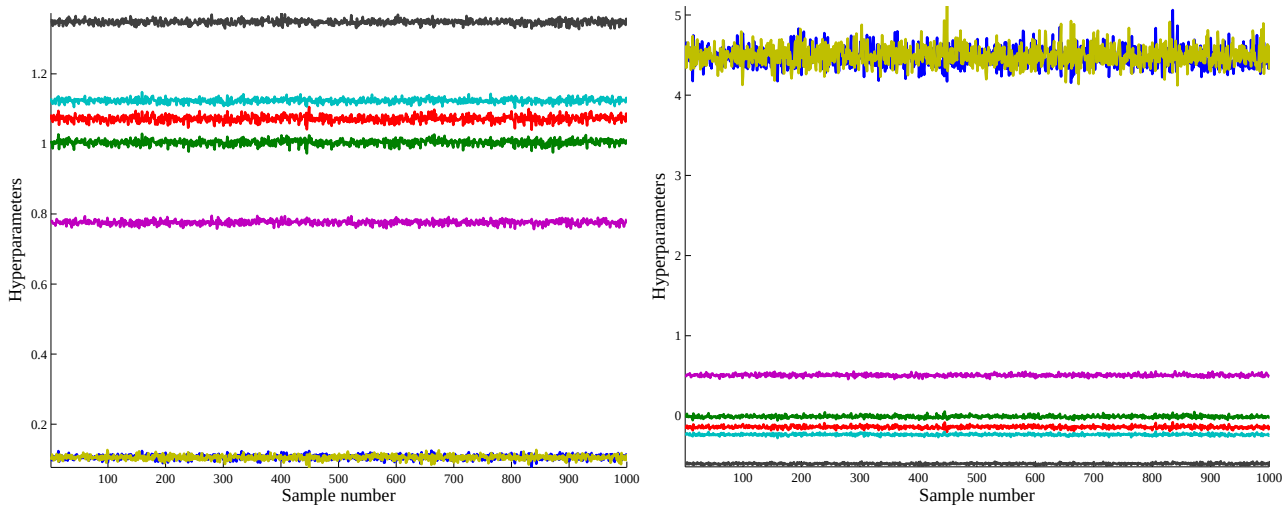
сматриваются выборки размером 100 и 1000. Зависимая переменная — зашумленная линейная или обобщенно-линейная функция входа. Рассматривались обобщенные-линейные функции $y = \exp(-\mathbf{w}^T\mathbf{x})$ и $y = \sin(\mathbf{w}^T\mathbf{x})$. Шум состоял из независимых нормальнораспределенных величин с дисперсией $\frac{1}{4}$.

Зависимость параметра от гиперпараметров. На рисунках приведена зависимость параметра w и гиперпараметра α, которые соответствуют нешумовому признаку. Мы видим, что параметр сильно коррелирует с гиперпараметром, при этом, нет зависимости от числа шумовых признаков.

Сравнение гиперпараметров для разных признаков. Гиперпараметры α_i могут [2] служить мерой информативности признаков. Сравнивались логарифм гиперпараметра значимого признака и минимальный из логарифмов гиперпарамтеров для незначимых признаков. Бралось усреднение логарифма по пяти различным выборкам. Результаты приведены на рисунках 2. На рисунке 2 видно, что в большинстве случаев значение гиперпараметра для значимого признака меньше, чем минимальное значение гиперпараметров для шумового, однако, в некоторых случаях наблюдаются выбросы.

Проводился аналогичный эксперимент для двух информативных признаков, причем сравнивался максимальное значение гиперпараметра для информативных признаков с минимальным значением признака для шумовых признаков. На рисунках 3 видно, что информативные признаки имели меньшие значения гиперпараметра α, чем информативные. Таким образом, удается выделить информативные и шумовые признаки. На рисунке 4показано сравнение информативности первого и второго информативных признаков, видно, что из-за большего веса один признак информативнее другого для линейной модели. Так же отметим, что для обобщенно-линейной функции не удается выделить наиболее информативный признак, в некоторых случаях гиперпараметры для одного из признаков стремятся к бесконечности.

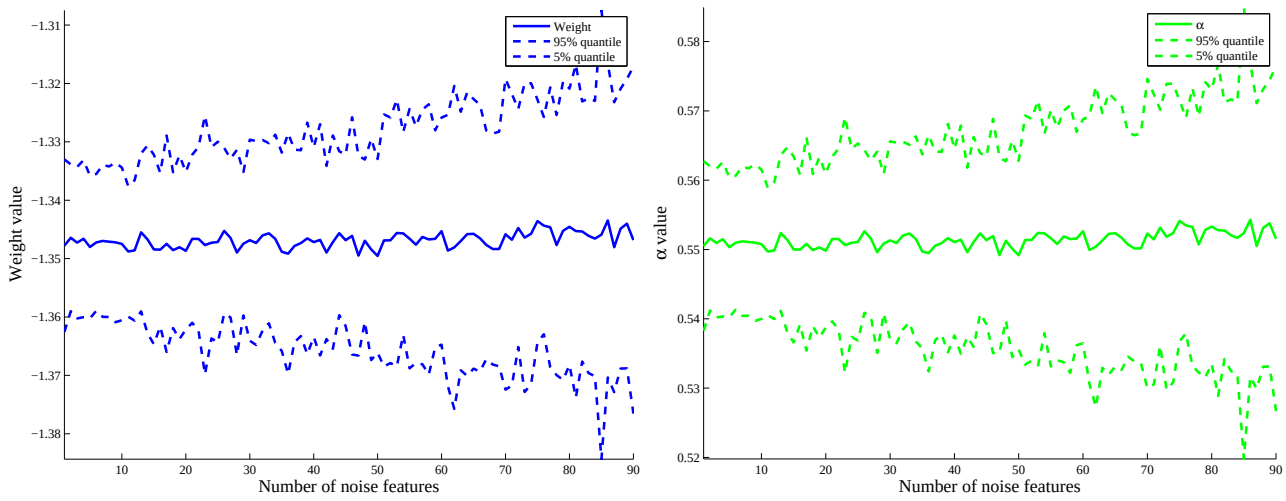
Реальные данные. Использовались реальные данные по определения характеристик цемента по его составу [4]. Данные были нормализованы так, что как у свободных, так и у зависимой переменной были нулевые математические ожидания и единичные дисперсии. Для данных без шумовых признаков алгоритм был запущен сто раз на разных подвы-



(a) Веса w_i (b) Гиперпараметры α_i

Рис. 5. Веса и гиперпараметры для выборки без шумовых признаков

борках размера 90 (размер полной выборки — 103). Результаты приведены на рисунке 5. Видно, что признаки разделяются по информативности и что информативность почти всегда эквивалента модулю веса.



(a) Вес w_7 (b) Гиперпараметр α_7

Рис. 6. Зависимость квантили оценки параметров и гиперпараметров при добавлении шумовых признаков

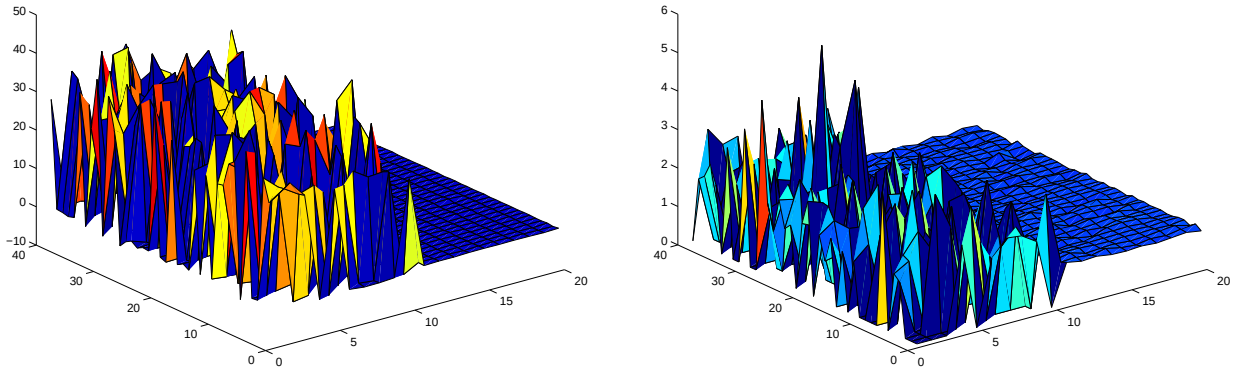
Так же был проведен следующий эксперимент. К начальному набору свободных переменных был добавлен ряд шумовых признаков, затем на ста запусках была оценена 95-процентная квантиль рассматриваемой величины. На рисунке 6 видно, что увеличение числа шумовых признаков увеличивает, хоть и не сильно, квантиль как оценки параметра, так и оценки гиперпараметра для разных признаков. Отметим, что, тем не менее, это не влияет на разделимость признаков по информативности.

Вычислительный эксперимент: мультиколлинеарные признаки

Модельные данные . Рассматривался следующий набор данных. Была сгенерирована выборка из нормального распределения размером 100 точек, количество признаков — двадцать. Ковариационная матрица первых десяти признаков имела вид:

$$\begin{pmatrix} 1.1 & 1 & 1 & \dots & 1 \\ 1 & 1.1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1.1 \end{pmatrix}$$

Ковариационная матрица для последних десяти признаков была единичной. Первая и вторая десятки признаков были порождены независимо.



(a) Гиперпараметры α_i (b) Параметры w_i

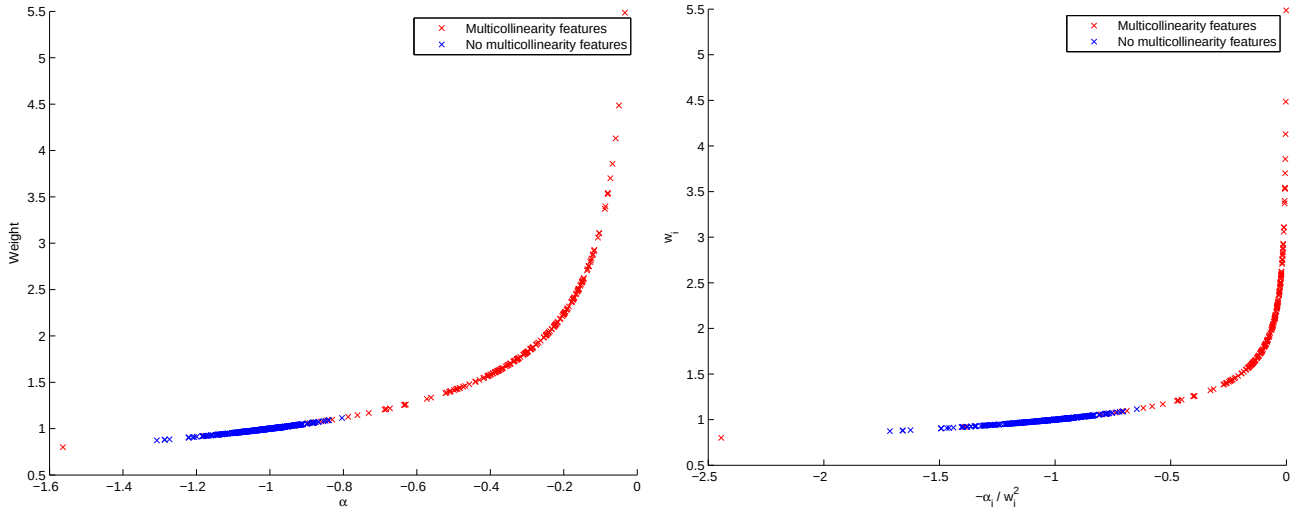
Рис. 7. Полученные значения параметров и гиперпараметров

Вектор откликов имел вид:

$$y = \sum_{i=1}^{n} x_i.$$

Было сделано 50 запусков эксперимента. Полученные значения логарифмов гиперпараметров α_i и параметров w_i изображены на рисунке 7. Ближе к читателю расположены оценки, полученные для коррелирующих признаков, дальше — для не коррелирующих признаков. Видно, что для признаков, не являющихся мультиколлинеарными, оценки значений гиперпараметров и параметров мало зависят от обучающей выборки. В то же время, для мультиколлинеарных признаков значения гиперпараметров и параметров сильно менялись от запуска к запуску.

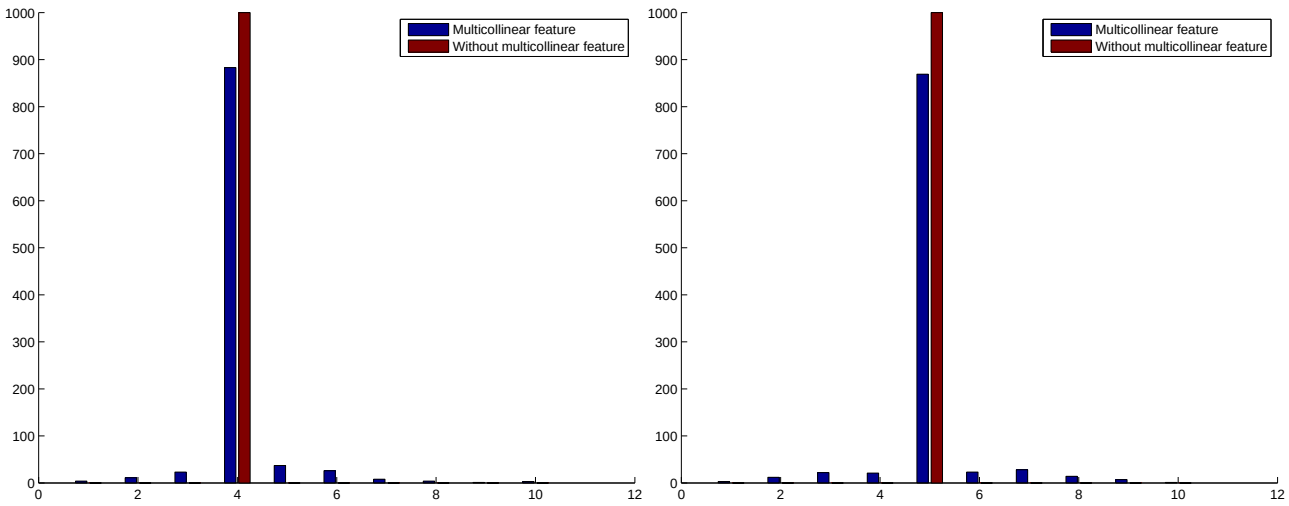
Для признаков с ненулевыми весами была построена кривая зависимости значений параметров от гиперпараметров (отметим, что истинное значение всех параметров равно единице). Полученная кривая приведена на рисунке 8 для тех признаков, параметры которых больше нуля. Мы видим, что при нормализации гиперпараметра α_i на w_i^2 признаки разделяются на две группы, в которых примерно одинаковые информативности. Таким образом, алгоритм верно классифицировал, что информативность признака, связанного с другими признаками посредством корреляции выше, чем информативность независимых



(a) Зависимость w_i от α_i (b) Зависимость w_i от $\frac{\alpha_i}{w_i^2}$

Рис. 8. Зависимость значения параметра w_i от гиперпараметра α_i

признаков. Отметим так же, что для некоторых признаков вес получался равным нулю. Все такие признаки принадлежали группе мультиколлинеарных.



(a) Первый признак (b) Второй признак

Рис. 9. Сравнительные гистограммы значений гиперпараметров

Реальные данные. Использовались реальные данные [4]. К данным добавлялся признак, сильно коррелирующий с одним из предложенных. Такой признак равнялся зашумленному стандартным нормальным шумом признаку. Сравнительные гистограммы значений гиперпараметров приведены на рисунке 9. Видно, что добавление мультикоррелирующего признака влияет на значение информативности исходного признака.

Существенно нелинейная регрессия. В этом эксперименте порождались модели существенно нелинейной регрессии [2, 3], затем рассматривалось распределение параметров и гиперпараметров для полученных моделей. Размер обучающей выборки — 10000

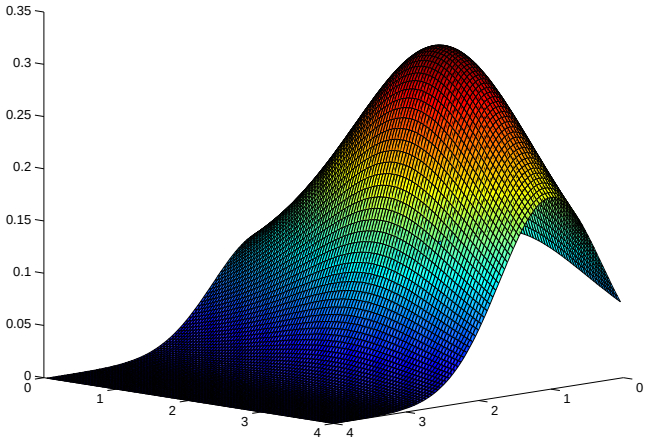


Рис. 10. Функция Котанчека

точек, делалась попытка аппроксимации функции, предложенной Котанчеком

$$f(x_1, x_2) = \frac{e^{-(x_1-1)^2}}{(x_2 - 2.5)^2 + 3.2}.$$

Вид функции показан на рисунке 10. Полученное распределение значений параметров и гиперпарамтеров — на рисунке 11. Видно, что значения параметров для разных моделей получают похожие значения, не зависимые от значения гиперпараметра. Это связано с линейным членом нелинейной модели, который появляется достаточно часто в функциях, точно приближающих искомую зависимость.

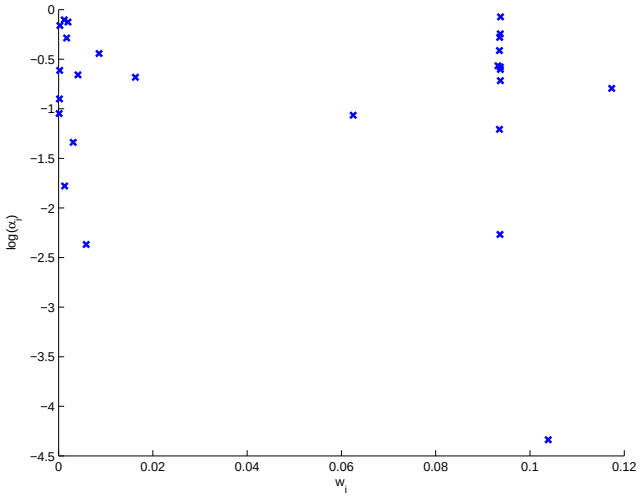


Рис. 11. Зависимость значения параметра w_i от гиперпараметра α_i для существенно нелинейных моделей

Выводы

Полученные результаты говорят о том, что предложенный подход является устойчивым к добавлению шумовых и мультиколлинеарных признаков.

Литература

[1] В. В. Стрижов, Р. А. Сологуб, *Индуктивное порождение регрессионных моделей предполагаемой волатильности для опционных торгов*. Вычислительные технологии, 14, 2009.

[2] В. В. Стрижов, Р. А. Сологуб, *Алгоритм выбора нелинейных регрессионных моделей с анализом гиперпараметров*. ММРО-14, 2009.

[3] А. А Зайцев, *Выбор моделей нелинейной регрессии с анализом гиперпараметров*. Конференция МФТИ, 2010.

[4] Yeh, I. and others, *Modeling slump flow of concrete using second-order regressions and artificial neural networks*. Cement and Concrete Composites, 29, 2007.

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 вышеназванных 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 $\varphi(\mathbf{x}, \mathbf{w}) \to \mathbb{R}^1$. 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 соответствен 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 $id(\mathbf{x})$.

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 $\mathbf{w} = [c_0, c_1, \ldots, c_n]$.

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

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 $j = 1, \ldots, size$(Примитивы)
  НовыйЭлемент:=Примитивы(j);
  ВременноеДерево:=[ТекущееДерево;НовыйЭлемент];
  если ПроверкаПравилПорождения(ВременноеДерево)==1 **то**
    ТекущееДерево:=ВременноеДерево;

НовыеЭлементы=Примитивы(НовыйЭлемент).КоличествоДетей;
для $i = 1, \ldots,$ 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:

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

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

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

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.

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

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:

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

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

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

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.

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

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:

$$\begin{aligned} f(\mathbf{w},\mathbf{x}) &= \nu \sum_{i=1}^{N} u(S_i(\mathbf{x},\mathbf{w})), \text{где } S_i \\ S_i(x,\mathbf{w}) &= \sum_j w_j u(A_j(\mathbf{x},\mathbf{w})) \\ A_j(x,\mathbf{w}) &= \mathbf{x}^T\mathbf{w} \end{aligned} \tag{4}$$

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 поддеревья, 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 созаднными 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^2_{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

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

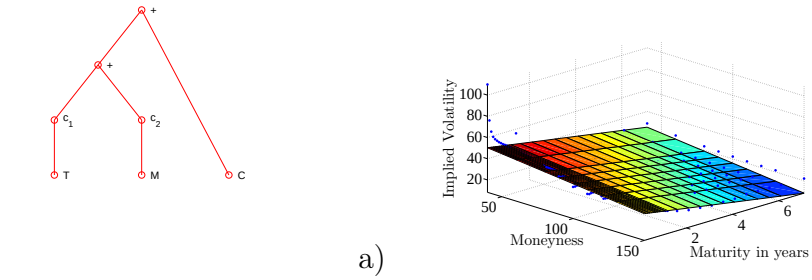
Для анализа выбраны данные торгов опционами Brent Crude Oil — опционы Chicago Mercantile Exchange на сырую нефть марки Brent. Фактически использованы все ликвидные по состоянию на 11 ноября 2011 года опционы — с датами исполнения от 15 ноября 2011 года до 15 декабря 2018 года. Волатильность строилась 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.

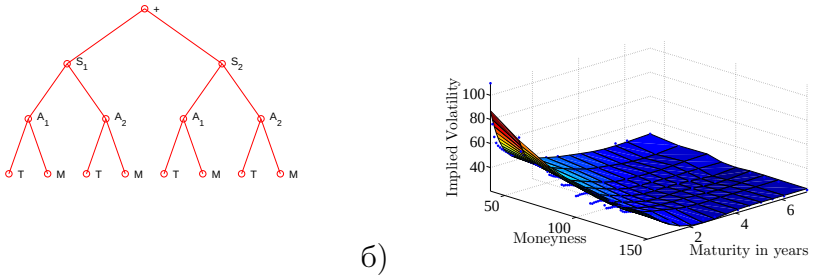
Регрессионная выборка $\{(\mathbf{x}_n, y_n)\} = \{(\langle M_n, t_n\rangle, \sigma_n)\}$ была построена на основе данных системы Bloomberg [17], [18] и дополнена с помощью исходных данных — исторических цен опциона $S_{K,t}$ и базового инструмента P_t, где $K \in \mathcal{K}, t \in T$, следующим образом.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

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

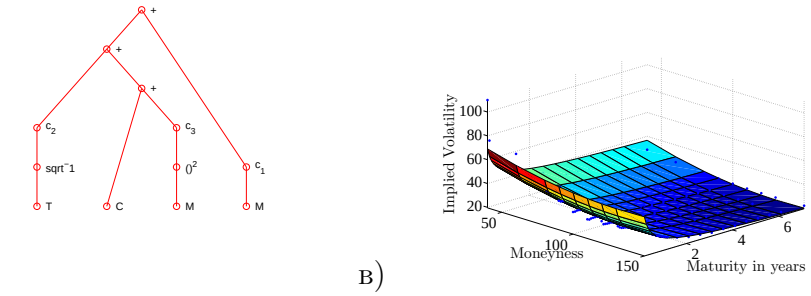
по значению волатильности. Здесь справедливая цена опциона C вычислена по формуле Блэка-Шоулза. Время t выраженное в годах до момента исполнения опциона рассчитывается по формуле $t = \tau/365$, где τ — число дней, оставшихся до исполнения опциона. Значение в искомой точке по шкале относительной цены 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.



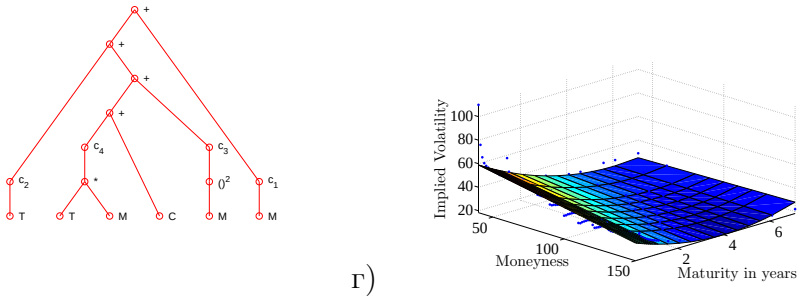
а)



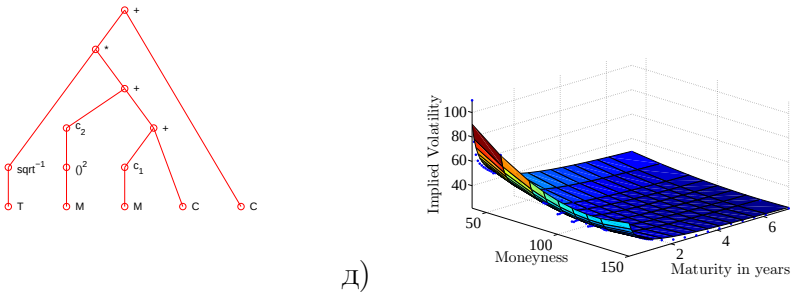
б)



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^2_{adj}. 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: $\sigma_{\text{imp}} = c_1 M + c_2 T + C$,
б) нейронная сеть: $\sigma_{\text{imp}} = \sum_i = 1^{10} S_i(\sum_j A_j(M, T)$,
в) обобщенно-линейная модель: $\sigma_{\text{imp}} = c_1 M + c_2 M^2 + c_3(\sqrt{(T)})^{-}1 + C$,
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 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 — Symbiloc Regression by Means of Arbitrfary 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 Р. 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 ofnonlinearity 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.

Индуктивное порождение суперпозиций в задачах нелинейной регрессии

Г. И. Рудой
rudoy@forecsys.ru
Московский физико-технический институт

При восстановлении нелинейной регрессии рассматривается набор индуктивно порожденных моделей с целью выбора оптимальной. В работе исследуется алгоритм индуктивного порождения допустимых существенно нелинейных моделей. Предлагается алгоритм, порождающий все возможные суперпозиции заданной сложности за конечное число шагов, и приводится его теоретическое обоснование. Приводятся результаты вычислительного эксперимента по моделированию волатильности опционов.

Ключевые слова: *Символьная регрессия, нелинейные модели, индуктивное порождение, волатильность опционов.*

Введение

В ряде приложений [1, 2, 3] возникает задача восстановления регрессии по набору известных данных. При этом предполагается, что модель должна иметь возможность быть проинтерпретированной экспертом в контексте предметной области.

Одним из методов, позволяющих получать интерпретируемые модели, является символьная регрессия [4, 5, 6], согласно которой известные данные приближаются некоторой математической формулой, например, $\sin x^2 + 2x$ или $\log x - \frac{e^x}{x}$. Эти формулы являются произвольными суперпозициями функций из некоторого заданного набора. Одна из возможных реализаций этого метода предложена Джоном Коза [7, 8], использовавшим эволюционные алгоритмы для реализации символьной регрессии. Иван Зелинка предложил дальнейшее развитие этой идеи [9], получившее название аналитического программирования.

Алгоритм построения требуемой математической модели выглядит следующим образом: дан набор примитивных функций, из которых можно строить различные формулы (например, степенная функция, +, sin, tan). Начальный набор формул строится либо произвольным образом, либо на базе некоторых предположений эксперта. Затем на каждом шаге производится оценка каждой из формул согласно функции ошибки либо другого функционала [10] качества. На базе этой оценки у некоторой части формул случайным образом заменяется одна элементарная функция на другую (например, sin на cos или + на $\times$), а у некоторой другой части происходит взаимный попарный обмен подвыражениями в формулах.

Получаемая формула является математической моделью [11] исследуемого процесса или явления — то есть, это математическое отношение, описывающее основные закономерности, присущие этому явлению.

Целью настоящей работы является теоретическое обоснование алгоритмов индуктивного порождения моделей и анализ этих алгоритмов. Одним из основных результатов является доказательство их принципиальной корректности, то есть, способности породить искомую формулу.

Алгоритм индуктивного порождения моделей, сформулированный в настоящей работе, решает некоторые типичные проблемы предложенных ранее методов, упомянутые, например, в [9], а именно:

— Порождение рекурсивных суперпозиций, суперпозиций, содержащих несоответствующее используемым функциям число аргументов, и т. д. — в предложенном алгоритме эти проблемы не возникают по построению.
— Несовпадение области определения некоторой примитивной функции и области значений ее аргументов (возможно, тоже некоторых суперпозиций).
— При ограничении числа примитивных функций, участвующих в суперпозиции, а также при соответствующем задании множества примитивных функций исключается проблема слишком сложных суперпозиций.

Во второй части данной работы формально поставлена задача построения алгоритма индуктивного порождения моделей. Затем, в третьей части строится искомый алгоритм для частного случая непараметризованных моделей и доказывается его корректность, а затем алгоритм обобщается на случай моделей, имеющих параметры. В четвертой части описываются вспомогательные технические приемы, использованные в практическом алгоритме порождения моделей, описанном в пятой части. Результаты вычислительного эксперимента приведены в шестой части настоящей работы.

Постановка задачи

Пусть дана регрессионная выборка:

$$D = \{(\mathbf{x}_i, y_i) \mid i \in \{1, \ldots, N\}, \mathbf{x}_i \in \mathbb{X} \subset \mathbb{R}^n, y_i \in \mathbb{Y} \subset \mathbb{R}\},$$

где N — объем регресионной выборки (число объектов), $\mathbf{x}_i$ — вектор значений независимых переменных i-ого объекта, y_i — значение зависимой переменной у i-ого объекта, $\mathbb{X}$ — множество значений независимых переменных, лежащее в $\mathbb{R}^n$, $\mathbb{Y}$ — множество значений зависимой переменной.

Требуется выбрать параметрическую функцию $f : \Omega \times \mathbb{R}^n \to \mathbb{R}$ из порождаемого множества $\mathcal{F} = \{f_r\}$, где Ω — пространство параметров, доставляющую минимум некоторому функционалу ошибки, определяемому ниже.

То есть, если множество всех суперпозиций:

$$\mathcal{F} = \{f_r \mid f_r : (\boldsymbol{\omega}, \mathbf{x}) \mapsto y \in \mathbb{Y}, r \in \mathbb{N}\},$$

то требуется найти такой индекс $\hat{r}$, что функция f_r среди всех $f \in \mathcal{F}$ доставляет минимум функционалу качества S при данной регрессионной выборке D:

$$\hat{r} = \arg\min_{r \in \mathbb{N}} S(f_r \mid \hat{\boldsymbol{\omega}}_{\boldsymbol{r}}, D), \tag{1}$$

где $\hat{\boldsymbol{\omega}}_r$ — оптимальный вектор параметров функции f_r для каждой $f \in \mathcal{F}$ при данной регрессионной выборке D:

$$\hat{\boldsymbol{\omega}}_{\boldsymbol{r}} = \arg\min_{\boldsymbol{\omega} \in \Omega} S(\boldsymbol{\omega} \mid f_r, D). \tag{2}$$

В качестве функционала качества S используется SSE:

$$S(\boldsymbol{\omega}, f, D) = \sum_{i=1}^{N} (y_i - f(\boldsymbol{\omega}, \mathbf{x}_i))^2 \mid (\mathbf{x}_i, y_i) \in D. \tag{3}$$

Сформулируем также постановку теоретической задачи. Для этого сначала введем понятие суперпозиции функций.

Если множество значений $\mathbb{Y}_i$ функции f_i содержится во множестве определения $\mathbb{X}_{i+1}$ функции f_{i+1}, то есть

$$f_i : \mathbb{X}_i \to \mathbb{Y}_i \subset \mathbb{X}_{i+1}, i = 1, 2, \ldots, \theta - 1,$$

то функция

$$f_\theta \circ f_{\theta-1} \circ \cdots \circ f_1, \theta \geqslant 2,$$

определяемая равенством

$$(f_\theta \circ f_{\theta-1} \circ \cdots \circ f_1)(\mathbf{x}) = f_\theta(f_{\theta-1}(\ldots(f_1(\mathbf{x})))), x \in \mathbb{X}_1,$$

называется *сложной функцией*[12] или *суперпозицией функций* $f_1, f_2, \ldots, f_\theta$.

Таким образом, получаем

Определение 1. *Суперпозиция функций — функция, представленная как композиция нескольких функций.*

Пусть $G = \{g_1, \ldots, g_l\}$ — множество данных порождающих функций, а именно, для каждой $g_i \in G$ заданы:

— сама функция g_i (например, sin, cos, $\times$),
— арность функции и порядок следования аргументов,
— домен ($\mathrm{dom}g_i$) и кодомен ($\mathrm{cod}g_i$) функции,
— область определения $\mathcal{D}g_i \subset \mathrm{dom}g_i$) и область значений $\mathcal{E}g_i \subset \mathrm{cod}g_i$.

Требуется построить упомянутую функцию f как суперпозицию порождающих функций из заданного множества G.

Поясним различие между последними двумя пунктами. Например, $\mathrm{dom}f$ показывает, значения из какого множества принимает функция f (целые числа, действительные числа, декартово произведение целых чисел и $\{0, 1\}$, и т. п.). Область определения же показывает, на каких значениях из $\mathrm{dom}f$ функция f определена и имеет смысл. Так, для функции $f(x_1, x_2) = \log_{x_1} x_2$:

$$\mathrm{dom}f = \mathbb{R} \times \mathbb{R},$$

$$\mathrm{cod}f = \mathbb{R},$$

$$\mathcal{D}f = \{(x_1, x_2) \mid x_1 \in (0; 1) \cup (1; +\infty), x_2 \in (0; +\infty)\},$$

$$\mathcal{E}f = (-\infty; +\infty).$$

Требуется также:

— построить алгоритм $\mathfrak{A}$, за конечное число итераций порождающий любую конечную суперпозицию данных примитивных функций,
— указать способ проверки изоморфности двух суперпозиций.

Заметим, что мы не требуем для примитивных функций свойства их непорождаемости в наиболее общей формулировке типа принципиальной невозможности породить в ходе работы искомого алгоритма суперпозицию, изоморфную некоторой функции из G. Такое требование является слишком ограничивающим. В частности, невозможно было бы иметь в G одновременно, например, функции id, exp и log, так как $\mathrm{id} \equiv \log \circ \exp$.

В дальнейшем будем также считать, что суперпозиция, соответствующая единственной свободной переменной ($f(\mathbf{x}) = x_i$), полностью эквивалентна функции вида $\mathrm{id}x_i$.

Алгоритм индуктивного порождения допустимых суперпозиций

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

— В вершинах V_i дерева Γ_f находятся соответствующие порождающие функции $g_s, s = s(i)$.
— Число дочерних вершин у некоторой вершины V_i равно арности соответствующей функции g_s.
— Порядок смежных некоторой вершине V_i вершин соотвествует порядку аргументов соответствующей функции $g_{s(i)}$.
— В листьях дерева Γ_f находятся свободные переменные x_i либо числовые параметры ω_i.
— Порядок вершин V_i в смысле уровня вершин определяет порядок вычисления примитивных функций: дерево вычисляется снизу вверх. То есть, сначала подставляются конкретные значения свободных переменных, затем вычисляются значения в вершинах, все дочерние вершины которых — свободные переменные, и так далее до тех пор, пока не останется единственная вершина, бывшая корнем дерева, содержащая результат выражения.

Таким образом, вычисление значения выражения f в некоторой точке с данным вектором параметров ω эквивалентно подстановке соответствующих значений свободных переменных x_i и параметров ω_i в дерево Γ_f выражения.

Заметим важное свойство таких деревьев: каждое поддерево Γ_f^i дерева Γ_f, соответствующее вершине V_i, также соответствует некоторой суперпозиции, являющейся составляющей исходной суперпозиции f.

Для примера рассмотрим дерево, соответствующиее суперпозиции $f = \sin(\ln x_1) + \frac{x_2^3}{2}$ (см. рис 1).

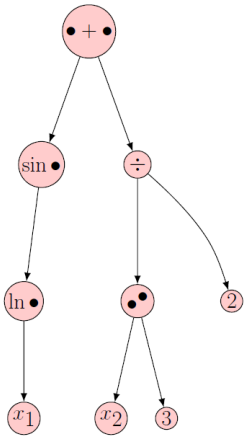


Рис. 1. Дерево выражения $\sin(\ln x_1) + \frac{x_2^3}{2}$

Здесь точками обозначены аргументы функций. Как видно, корнем дерева является вершина, соответствующая операции сложения, которая должна быть выполнена в последнюю очередь. Операция сложения имеет два различных поддерева, соответствующих двум аргументам этой операции. Заметим также, что здесь не использованы операции типа «разделить на два» или «возвести в куб». Вместо этого используются операции деления и возведения в степень в общем виде, а в данном конкретном дереве соответствующие аргументы зафиксированы соответствующими константами.

Алгоритм порождения суперпозиций

Сначала определим понятие *глубины суперпозиции*:

Определение 2. *Глубина суперпозиции f — максимальная глубина дерева Γ_f.*

Теперь опишем итеративный алгоритм $\mathfrak{A}^*$, порождающий суперпозиции, не содержащие параметров. Описанный алгоритм породит любую суперпозицию конечной глубины за конечное число шагов.

Пусть дано множество примитивных функций $G = \{g_1, \ldots, g_l\}$ и множество свободных переменных $X = \{x_1, \ldots, x_n\}$.

Для удобства будем исходить из предположения, что множество G состоит только из унарных и бинарных функций, и разделим его соответствующим образом на два подмножества: $G = G_b \cup G_u \mid G_b = \{g_{b_1}, \ldots, g_{b_k}\}, G_u = \{g_{u_1}, \ldots, g_{u_l}\}$, где G_b — множество всех бинарных функций, а G_u — множество всех унарных функций из G. Потребуем также наличия id в G_b.

Алгоритм 1. Алгоритм $\mathfrak{A}^*$ итеративного порождения суперпозиций.

1. Перед первым шагом зададим начальные значения множества $\mathcal{F}_0$ и вспомогательного индексного множества $\mathcal{I}$, служащего для запоминания, на какой итерации впервые встречена каждая суперпозиция:
$$\mathcal{F}_0 = X,$$
$$\mathcal{I} = \{(x, 0) \mid x \in X\}.$$
2. Для множества $\mathcal{F}_i$ построим вспомогательное множество U_i, состоящее из суперпозиций, полученных в результате применения функций $g_u \in G_u$ к элементам $\mathcal{F}_i$:
$$U_i = \{g_u \circ f \mid g_u \in G_u, f \in \mathcal{F}_i\}.$$
3. Аналогичным образом построим вспомогательное множество B_i для бинарных функций $g_b \in G_b$:
$$B_i = \{g_b \circ (f, h) \mid g_b \in G_b, f, h \in \mathcal{F}_i\}.$$
4. Обозначим $\mathcal{F}_{i+1} = \mathcal{F}_i \cup U_i \cup B_i$.
5. Для каждой суперпозиции f из $\mathcal{F}_{i+1}$ добавим пару $(f, i+1)$ в множество $\mathcal{I}_f$, если суперпозиция f еще там не присутствует.
6. Перейдем к следующей итерации.

Тогда $\mathcal{F} = \cup_{i=0}^{\infty} \mathcal{F}_i$ — множество всех возможных суперпозиций конечной длины, которые можно построить из данного множества примитивных функций.

Вспомогательное множество $\mathcal{I}$ позволяет запоминать, на какой итерации была впервые встречена данная суперпозиция. Это необходимо, так как каждая суперпозиция, впервые порожденная на i-ой итерации, будет порождена еще раз и на любой итерации после i. Одной из возможностей избежать необходимости в этом множестве является построение $\mathcal{F}_{i+1}$ как $\mathcal{F}_{i+1} = U_i \cup B_i$ (без $\mathcal{F}_i$), а множества U_i и B_i строить следующим образом:

$$U_i = \{g_u \circ f \mid g_u \in G_u, f \in \cup_{j=0}^{i} \mathcal{F}_j\},$$

$$B_i = \{g_b \circ (f, h) \mid g_b \in G_b, f, h \in \cup_{j=0}^{i} \mathcal{F}_j\}.$$

Алгоритм $\mathfrak{A}^*$ очевидным образом обобщается на случай, когда множество G содержит функции произвольной (но конечной) арности. Действительно, для такого обобщения достаточно строить аналогичным образом вспомогательные множества для этих функций, а

именно, для множества функций G_n арности n построим вспомогательное множество H_i^n вида:

$$H_i^n = \{g \circ (f_1, f_2, \ldots, f_n) \mid g \in G_n, f_j \in \mathcal{F}_i\}.$$

В этих обозначениях $U_i \equiv H_i^1$, а $B_i \equiv H_i^2$.

Тогда множество $\mathcal{F}_{i+1} = \mathcal{F}_i \cup_{n=0}^{n_{max}} H_i^n$, где n_{max} — максимальное значение арности функций из G.

Теорема 1. *Алгоритм $\mathfrak{A}^*$ действительно породит любую конечную суперпозицию за конечное число шагов.*

Доказательство. Чтобы убедиться в этом, найдем номер итерации, на котором будет порождена некоторая произвольная конечная суперпозиция f. Чтобы найти этот номер, пронумеруем вершины графа Γ_f по следующим правилам:

— Если это вершина со свободной переменной, то она имеет номер 0.
— Если вершина V соответствует унарной функции, то она имеет номер $i + 1$, где i — номер дочерней для этой функции вершины.
— Если вершина V соответствует бинарной функции, то она имеет номер $i + 1$, где $i = \max(l, r)$, а l и r — номера, соответственно, первой и второй дочерней вершины.

Нумеруя вершины графа Γ_f таким образом, мы получим номер вершины, соответствующей корню графа. Это и будет номером итерации, на которой получена суперпозиция f.

Иными словами, для любой суперпозиции мы можем указать конкретный номер итерации, на котором она будет получена, что и требовалось. ■

В предложенных ранее методах[9] построения суперпозиций необходимо было самостоятельно следить за тем, чтобы в ходе работы алгоритма не возникало «зацикленных» суперпозиций типа $f(x, y) = g(f(x, y), x, y)$. Заметим, что в предложенном алгоритме $\mathfrak{A}^*$ такие суперпозиции не могут возникнуть по построению.

Порождение параметризованных моделей

Алгоритм в таком виде не позволяет получать выражения, содержащие численные параметры $\boldsymbol{\omega}$ суперпозиции $f(\boldsymbol{\omega}, \mathbf{x})$. Покажем, однако, на примере конструирования множеств U_i и B_i, как исходный алгоритм $\mathfrak{A}^*$ может быть расширен с учетом таких параметров путем введения параметров:

$$U_i = g_u \circ (\alpha f + \beta),$$
$$B_i = g_b \circ (\alpha f + \beta, \psi h + \varphi).$$

Будем обозначать этот расширенный алгоритм как $\mathfrak{A}$. Здесь параметры α, β зависят только от комбинации g_u, f (или g_b, f, h для $\alpha, \beta, \psi, \varphi$). Соответственно, для упрощения их индексы опущены.

Иными словами, мы предполагаем, что каждая суперпозиция из предыдущих итераций входит в следующую, будучи умноженной на некоторой коэффициент и с константной поправкой.

Очевидно, при таком добавлении параметров $\alpha, \beta, \psi, \varphi$ мы не изменяем мощности получившегося множества суперпозиций, поэтому алгоритм и выводы из него остаются корректны. В частности, исходный алгоритм является частным случаем данного при $\alpha \equiv \psi \equiv 1, \beta \equiv \varphi \equiv 0$.

$\alpha, \beta, \psi, \varphi$ являются параметрами модели. В практических приложениях можно оптимизировать значения этих параметров у получившихся суперпозиций, например, алгоритмом Левенберга-Марквардта [13, 14].

Заметим также, что такая модификация алгоритма позволяет нам получить единицу, например, для построения суперпозиций типа $\frac{1}{x}$: $1 = \alpha\ id\ x + \beta \mid \alpha = 0, \beta = 1$.

Отдельно подчеркнем, что параметры $\boldsymbol{\omega}$ у различных суперпозиций различны. Однако, так как каждый из параметров зависит только от соответствующей комбинации функций, к которым он относится, конкретные значения параметров не учитываются при поиске одинаковых суперпозиций. Иными словами, при тестировании суперпозиций на равенство сравниваются лишь структуры соответствующих им деревьев и значения в узлах, соответствующих функциям и свободным переменным.

Заметим, что и этот алгоритм очевидным образом обобщается на случай множества G, содержащего функции произвольной арности.

Количество возможных суперпозиций

Посчитаем количество суперпозиций, получаемых после каждой итерации алгоритма $\mathfrak{A}$. Очевидно, с учетом вышеупомянутых оговорок касательно сравнения параметризованных суперпозиций, это количество равно количеству для алгоритма $\mathfrak{A}^*$.

Итак, пусть дано n независимых переменных: $|X| = n$, а мощность множества G распишем 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:

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

и 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:

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

И вообще, после k-ой итерации будет порождено:

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

Оценим порядок роста количества функций, порожденных после k-ой итерации.

Теорема 2. *Пусть в множестве примитивных функций G содержится l_p функций арности $p > 1$ и ни одной функции арности $p + k \mid k > 0$, и имеется $n > 1$ независимых переменных. Тогда справедлива следующая оценка количества суперпозиций, порожденных алгоритмом $\mathfrak{A}$ после k-ой итерации:*

$$|\mathcal{F}_k| = \mathcal{O}(l_p^{\sum_{i=0}^{k-1} p^i} n^{p^k}).$$

Доказательство. 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 $(n^m)^m = n^{m^2}$. И вообще, после k-ой итерации количество суперпозиций можно оценить как n^{m^k}.

Видно, что для оценки скорости роста количества порожденных суперпозиций можно 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

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

И вообще, скорость роста количества порожденных суперпозиций можно оценить как:

$$|\mathcal{F}_k| = \mathcal{O}(l_m^{\sum_{i=0}^{k-1} m^i} n^{m^k}).$$

Таким образом, получаем оценку для случая, когда в множестве G содержится l_p функций арности p и ни одной функции арности $p + k \mid k > 0$:

$$|\mathcal{F}_k| = \mathcal{O}(l_p^{\sum_{i=0}^{k-1} p^i} n^{p^k}).$$

■

Множество допустимых суперпозиций

Предложенный выше алгоритм позволяет получить действительно все возможные суперпозиции, однако, не все они будут пригодны в практических приложениях: например, $\ln x$ имеет смысл только при $x > 0$, а $\frac{x}{0}$ не имеет смысла вообще никогда. Выражения типа $\frac{x}{\sin x}$ имеют смысл только при $x \neq \pi k$.

Таким образом, необходимо введение понятия множества *допустимых* суперпозиций — то есть, таких суперпозиций, которые в условиях некоторой задачи корректны.

Определение 3. *Допустимая суперпозиция f — такая суперпозиция, значение которой определено для любой комбинации значений свободных переменных, область значений $\mathbb{X}$ которых определяется конкретной задачей, $\mathbb{X} \subset \mathbb{R}^n$ где n — число свободных переменных.*

Одним из способов построения только допустимых суперпозиций является модификация предложенного алгоритма таким образом, чтобы отслеживать совместность областей определения и областей значения соответствующих функций в ходе построения суперпозиций. Для свободных переменных это будет, в свою очередь, означать необходимость задания областей значений $\mathbb{X}$ пользователем при решении конкретных задач.

Заметим, что, хотя теоретически возможно выводить допустимость выражений вида $\frac{x}{\sin x}$ исходя из заданных условий на свободную переменную (например, что $x \in (\frac{\pi}{4}, \frac{\pi}{2})$), в общем случае это потребует решения неравенств в общем виде, что вычислительно неэффективно.

Таким образом, можно сформулировать очевидное *достаточное условие недопустимости* суперпозиции:

Определение 4. *Достаточное условие недопустимости суперпозиции f: в соответствующем дереве Γ_f хотя бы одна вершина V_i имеет хотя бы одну дочернюю вершину V_j такую, что область значений функции $g_{s(j)}$ шире, чем область определения функции $g_{s(i)}$:*

$$\exists i, j : V_i \in \Gamma_f, V_j \in \Gamma_f \wedge \exists \varkappa : \varkappa \in \mathcal{E}g_{s(j)} \wedge \varkappa \notin \mathcal{D}g_{s(i)}.$$

Говоря, что область значений функции f шире области определения функции g, мы имеем ввиду, что существует по крайней мере одно значение функции f, не входящее в область определения функции g.

Подчеркнем, что, хотя свободные переменные могут принимать, например, все значения из $\mathbb{R}$, выбором множества $\mathbb{X}$ можно обеспечить возможность использования их в качестве аргументов функциям с более узкой, чем $\mathbb{R}$, но не менее узкой, чем $\mathbb{X}$, областью определения, если это не противоречит данной регрессионной выборке.

Для построения множества допустимых суперпозиций достаточно построить множество всех возможных суперпозиций при помощи алгоритма $\mathfrak{A}$, а затем удалить из этого множества все суперпозиции, не удовлетворяющие сформулированному признаку.

Алгоритм Левенберга-Марквардта и мультистарт

Алгоритм Левенберга-Марквардта ($\mathcal{LM}$) [13, 14] предназначен для решения задачи минимизации функции, представляющей из себя сумму квадратичных членов. В частности, он используется для оптимизации параметров нелинейных регрессионных моделей в предположении, что в качестве критерия оптимизации используется среднеквадратичная ошибка модели на обучающей выборке:

$$S(\boldsymbol{\omega}) = \sum_{i=1}^{N} [y_i - f(\boldsymbol{\omega}, \mathbf{x}_i)]^2 \to \min,$$

где $\boldsymbol{\omega}$ — вектор параметров суперпозиции f.

$\mathcal{LM}$ может рассматриваться как комбинация методов Гаусса-Ньютона и градиентного спуска.

Перед началом работы алгоритма задается начальный вектор параметров $\boldsymbol{\omega}_0$. На каждой итерации этот вектор заменяется новой оценкой, $\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \boldsymbol{\delta}_k$. Для определения $\boldsymbol{\delta}_k = \boldsymbol{\delta}$ используется линейное приближение функции:

$$\mathbf{f}(\boldsymbol{\omega} + \boldsymbol{\delta}, \mathbf{X}) \approx \mathbf{f}(\boldsymbol{\omega}, \mathbf{X}) + \mathbf{J}\boldsymbol{\delta},$$

где $\mathbf{J}$ — якобиан функции $\mathbf{f}$ в точке $\boldsymbol{\omega}$.

Приращение $\boldsymbol{\delta}$ в точке $\boldsymbol{\omega}$, доставляющей минимум S, равно нулю, поэтому для нахождения последующего значения приращения $\boldsymbol{\delta}$ приравняем нулю вектор частных производных S по $\boldsymbol{\omega}$. То есть, в векторной нотации:

$$S(\boldsymbol{\omega} + \boldsymbol{\delta}) \approx \|\mathbf{y} - \mathbf{f}(\boldsymbol{\omega}) - \mathbf{J}\boldsymbol{\delta}\|^2.$$

Дифференциирование по $\boldsymbol{\delta}$ и приравнивание нулю приводит к следующему уравнению для $\boldsymbol{\delta}$:

$$(\mathbf{J}^T\mathbf{J})\boldsymbol{\delta} = \mathbf{J}^T[\mathbf{y} - \mathbf{f}(\boldsymbol{\omega})].$$

Левенберг предложил заменить $(\mathbf{J}^T\mathbf{J})$ на $(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})$, где λ — некоторый параметр регуляризации. Марквардт дополнил это предложение с целью более быстрого движения

по тем направлениям, где градиент меньше. Для этого вместо $\mathbf{I}$ используется диагональ матрицы $\mathbf{J}^T\mathbf{J}$, и искомое уравнение на $\boldsymbol{\delta}$ выглядит как:

$$(\mathbf{J}^T\mathbf{J} + \lambda \mathrm{diag}(\mathbf{J}^T\mathbf{J}))\boldsymbol{\delta} = \mathbf{J}^T[\mathbf{y} - \mathbf{f}(\boldsymbol{\omega})].$$

Решая это уравнение, получаем окончательное выражение для $\boldsymbol{\delta} = \boldsymbol{\delta}_k$:

$$\boldsymbol{\delta}_k = (\mathbf{J}^T\mathbf{J} + \lambda \mathrm{diag}(\mathbf{J}^T\mathbf{J}))^{-1}\mathbf{J}^T[\mathbf{y} - \mathbf{f}(\boldsymbol{\omega})].$$

Выбор λ

Для определения параметра регуляризации λ в настоящей работе применяется следующая эвристика.

В начале работы $\mathcal{LM}$ задается некоторое значение λ_0, например, 0.01, и фиксируется коэффициент $\nu > 1$. Затем, на каждой итерации алгоритма вычисляется значение функционала ошибки для $\lambda = \lambda_i$ и $\lambda = \frac{\lambda_i}{\nu}$. В случае, если хотя бы одно из этих значений доставляет функционалу ошибки меньшее значение, чем до этой итерации, то λ_{i+1} принимается равным этому значению. Иначе λ_i умножается на ν до тех пор, пока значение функционала ошибки не уменьшится.

Мультистарт

Как и всякий подобный алгоритм оптимизации, $\mathcal{LM}$ находит лишь локальный минимум. Для решения этой проблемы применяется метод *мультистарта*: случайным образом задается несколько начальных приближений, и для каждого из них запускается $\mathcal{LM}$. Если найдено несколько различных локальных минимумов, то выбирается тот из них, в котором значение $S(\boldsymbol{\omega})$ меньше всего.

Алгоритм итеративного стохастического порождения суперпозиций

Несмотря на то, что указанный ранее итеративный алгоритм порождения суперпозиций позволяет получить за конечное число шагов произвольную суперпозицию, для практических применений он непригоден, как и любой алгоритм, реализующий полный перебор, в связи с чрезмерной вычислительной сложностью. Вместо него можно использовать стохастические алгоритмы и ряд эвристик, позволяющих на практике получать за приемлемое время результаты, удовлетворяющие заранее заданным условиям «достаточной пригодности». В данном разделе описывается примененный в настоящей работе алгоритм.

Сначала опишем вспомогательный алгоритм случайного порождения суперпозиции:

Алгоритм 2. Алгоритм случайного порождения суперпозиции $\mathcal{RF}$.

Вход:

— Набор пороговых значений $0 < \xi_1 < \xi_2 < \xi_3 < 1$.
— Максимальная глубина порождаемой суперпозиции Td.

Алгоритм работает следующим образом. Генерируется случайное число ξ на интервале $(0; 1)$, и рассматриваются следующие случаи:

— $\xi \leqslant \xi_1$: результатом алгоритма является некоторая случайно выбранная свободная переменная.
— $\xi_1 < \xi \leqslant \xi_2$: результатом алгоритма является числовой параметр.

— $\xi_2 < \xi \leqslant \xi_3$: результатом алгоритма является некоторая случайно выбранная унарная функция, для определения аргумента которой данный алгоритм рекурсивно запускается еще раз.
— $\xi_3 < \xi$: результатом алгоритма является некоторая случайно выбранная бинарная функция, аргументы которой порождаются аналогичны образом.

При этом, порождение тривиальных суперпозиций (свободных переменных и параметров) запрещено: на самом первом шаге пороговые значения масштабируются таким образом, чтобы всегда породилась унарная или бинарная функция. Аналогично при превышении значения Td пороговые значения масштабируются таким образом, чтобы был порожден узел, соответствующий свободной переменной или параметру, и алгоритм завершился.

В ходе работы предлагаемого алгоритма каждой суперпозиции f ставится в соответствие ее *качество* Q_f (иногда будем говорить, что суперпозиция *оценивается*), рассчитываемое исходя из функции ошибки S_f этой суперпозиции на выборке D и ее сложности C_f — числа узлов в соответствующем графе Γ_f, по следующей формуле:

$$Q_f = \frac{1}{1+S_f}\left(\alpha\hat{Q} + \frac{1-\alpha\hat{Q}}{1+\exp(C_f-\tau)}\right), \tag{4}$$

где $\hat{Q}$ — минимальная приспособленность суперпозиции из критерия останова, α — некоторый коэффициент, $0 \ll \alpha < 1$, а τ — коэффициент, характеризующий желаемую сложность модели. Второй множитель в данной формуле выполняет роль штрафа за слишком большую сложность суперпозиции.

Таким образом, чем лучше результаты суперпозиции, тем ближе значение ее приспособленности к 1, и, наоборот, чем хуже — тем ближе к 0.

Итак, теперь опишем сам алгоритм:

Алгоритм 3. Итеративный алгоритм стохастического порождения суперпозиций.

Вход:

— Множество порождающих функций G, состоящее только из унарных и бинарных функций.
— Регрессионная выборка D.
— N_{max} — максимальное число одновременно рассматриваемых суперпозиций.
— I_{max} — максимальное число итераций алгоритма.
— Прочие параметры, используемые в (4) и алгоритме 2.

1. Инициализируется начальный массив $\mathcal{X}_f$ суперпозиций. А именно, порождается N_{max} суперпозиций алгоритмом 2.
2. Оптимизируются параметры $\boldsymbol{\omega}$ суперпозиций из $\mathcal{X}_f$ алгоритмом $\mathcal{LM}$.
3. Вычисляется значение Q_f для каждой еще не оцененной суперпозиции f из $\mathcal{X}_f$: для нее рассчитывается значение функции ошибки S_f согласно (3) на выборке D, и ставится в соответствие значение Q_f в соответствии с (4). Для суперпозиций, при вычислении Q_f которых была хотя бы раз получена ошибка вычислений из-за несовпадения областей определений и значений, принимается $Q_f = -\infty$.
4. Массив суперпозиций $\mathcal{X}_f$ сортируется согласно их приспособленности.
5. Наименее приспособленные суперпозиции удаляются из массива $\mathcal{X}_f$ до тех пор, пока его размер не станет равен N_{max}.

6. Отбирается некоторая часть наименее приспособленных суперпозиций из $\mathcal{X}_f$ (в данной работе — $\frac{1}{3}$ от числа всех суперпозиций). У этой части происходит случайная замена одной функции или свободной переменной на другую: генерируются две случайные величины, одна из которых служит для выбора вершины дерева Γ_f, которую предстоит изменить, а другая — для выбора нового элемента для этой вершины. Замена такова, чтобы сохранилась структура суперпозиции, а именно — в случае замены функции сохраняется арность, а свободная переменная заменяется только на другую свободную переменную. При этом исходные суперпозиции сохраняются в массиве $\mathcal{X}_f$.
7. Повторяются шаги $3-4$.
8. Производится случайный обмен поддеревьями наиболее приспособленных суперпозиций. Вершины, соответствующие этим поддеревьям, выбираются случайным образом. При этом исходные суперпозиции сохраняются в массиве $\mathcal{X}_f$.
9. Повторяются шаги $2-4$.
10. Проверяются условия останова: если либо число итераций больше I_{max}, либо в массиве $\mathcal{X}_f$ есть хотя бы одна суперпозиция с приспособленностью больше, чем $\hat{Q}$, то алгоритм останавливается, и результатом является наиболее приспособленная суперпозиция, иначе осуществляется переход к шагу 2.

Заметим, что выборка D не делится на обучающую и контрольную — контроль качества оставляется различным стандартным методикам типа скользящего контроля.

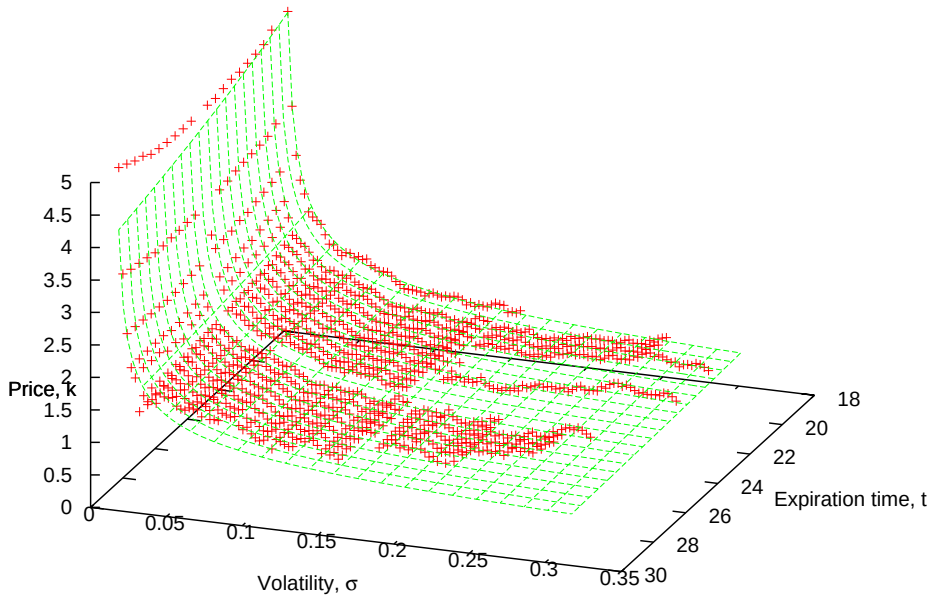
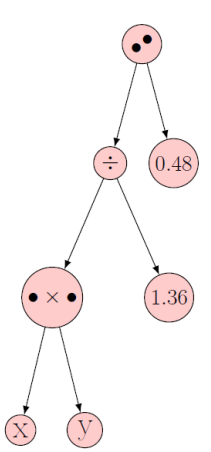


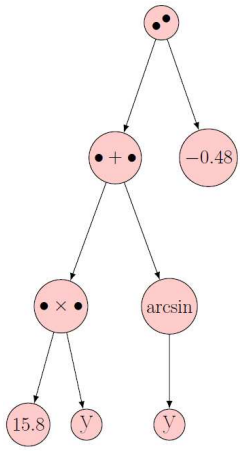
Рис. 2. Изометрическая проекция результирующей суперпозиции

Вычислительный эксперимент

В вычислительном эксперименте восстанавливается регрессионная зависимость волатильности опциона от его стоимости и сроков исполнения [15, 16]. Используются исторические данные о волатильности опционов Brent Crude Oil. Срок действия опциона —



(a) Граф суперпозиции $\left(\frac{1.36}{xy}\right)^{0.48}$



(c) Граф суперпозиции $(15.8y + \\ + \arcsin y)^{-0.48}$

Рис. 3. Графы результирующих суперпозиций

полгода, с 02.01.2001 по 26.06.2001, тип — право на продажу базового инструмента. Базовым инструментом в данном случае является нефть. Использовались ежедневные цены закрытия опциона и базового инструмета.

Данный инструмент имеет низкую волатильность, вследствие чего среди данных нет выбросов. В данных имеются пропуски, так как опционы с ценами, далекими от цен базового инструмента, не торговались сразу после выпуска опционов.

i	Суперпозиция	S_f	C_f
9	$\left(\frac{1.36}{xy}\right)^{0.48}$	≈ 0.0182	7
14	$(15.8y + \arcsin y)^{-0.48}$	≈ 0.0208	8
13	$(yx^{0.882} + \arcsin y)^{-0.482}$	≈ 0.0178	10
8	$0.125\frac{y}{(y^2)^{0.8+y}}$	≈ 0.0171	11
14	$\frac{\frac{3.86\cdot 10^{11}+y}{y\frac{1.227\cdot 10^{11}}{x^y}-2.46\cdot 10^8}}{y\cos\left(\frac{\frac{-5.89\cdot 10^{-3}+y}{y-5.47\cdot 10^{-3}}}{\frac{y\cos y}{y}}\right)}}{y^y\cos y + xy}$	≈ 0.0092	42

Таблица 1. Результаты вычислительного эксперимента

В ходе предобработки данных выяснено, что для больших значений волатильности зависимость принимает существенно неоднозначный характер, поэтому для облегчения аналитического описания моделировалась зависимость цены от волатильности и времени.

Использованные параметры алгоритма 3: $N_{max} = 200, I_{max} = 50, \hat{Q} = 0.95, \tau = 10, \alpha = \\ = 0.05$. При отсутствии улучшения результатов в течение нескольких итераций подряд алгоритм 3 также завершался.

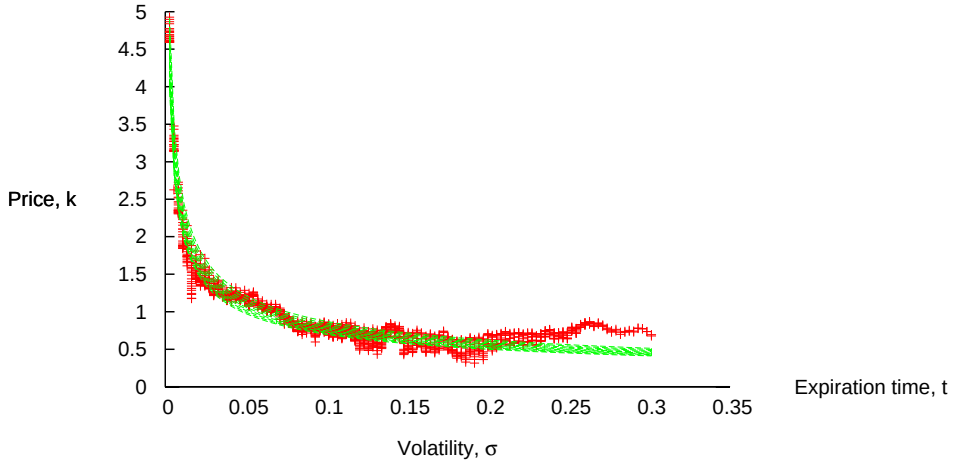


Рис. 4. Проекция результирующей суперпозиции

В таблице 1 приведены некоторые из наилучших суперпозиций, порожденных в результате работы алгоритма 3, в порядке возрастания их сложности. Указан номер итерации i, на которой суперпозиция была впервые получена, сама суперпозиция, среднеквадратичная ошибка (S_f) и сложность в смысле количества узлов в соответствующем графе выражения. Числовые коэффициенты в приведенных формулах и значения функционала S_f искусственно округлены до $2-3$ значащей цифры. В таблице 3 представлены графы первых двух упомянутых в таблице суперпозиций. На рисунках 2 и 4 отображены изометрическая проекция и проекция на одну из плоскостей для суперпозиции $\left(\frac{1.36}{xy}\right)^{0.48}$.

Заключение

В работе исследованы индуктивные алгоритмы порождения допустимых существенно нелинейных суперпозиций. Предложен переборный алгоритм, порождающий все возможные суперпозиции заданной сложности за конечное число шагов, и приведено его теоретическое обоснование.

Сформулированный алгоритм индуктивного порождения моделей решает некоторые типичные проблемы предложенных ранее методов, упомянутые, например, в [9].

Описан стохастический алгоритм индуктивного порождения существенно нелинейных суперпозиций и приведены результаты его работы для задачи моделирования волатильности опционов.

Литература

[1] John Duffy and Jim Engle-Warnick. Using symbolic regression to infer strategies from experimental data. In Shu-Heng Chen, editor, *Evolutionary Computation in Economics and Finance*, volume 100 of *Studies in Fuzziness and Soft Computing*, chapter 4, pages 61–84. Physica Verlag, 2002 2002.

[2] P. Barmpalexis, K. Kachrimanis, A. Tsakonas, and E. Georgarakis. Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation. *Chemometrics and Intelligent Laboratory Systems*, 107(1):75–82, 2011.

[3] Michael Schmidt and Hod Lipson. Symbolic regression of implicit equations. In Rick L. Riolo, Una-May O'Reilly, and Trent McConaghy, editors, *Genetic Programming Theory and Practice VII*, Genetic and Evolutionary Computation, chapter 5, pages 73–85. Springer, Ann Arbor, 14-16 May 2009.

[4] J. W. Davidson, D. A. Savic, and G. A. Walters. Symbolic and numerical regression: experiments and applications. In Robert John and Ralph Birkenhead, editors, *Developments in Soft Computing*, pages 175–182, De Montfort University, Leicester, UK, 29-30 June 2000. 2001. Physica Verlag.

[5] Claude Sammut and Geoffrey I. Webb. Symbolic regression. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, page 954. Springer, 2010.

[6] Vadim Strijov and Gerhard-Wilhelm Weber. Nonlinear regression model generation using hyperparameter optimization. *Computers & Mathematics with Applications*, 60(4):981–988, 2010.

[7] John R. Koza. Genetic programming. In James G. Williams and Allen Kent, editors, *Encyclopedia of Computer Science and Technology*, volume 39, pages 29–43. Marcel-Dekker, 1998. Supplement 24.

[8] John R. Koza. Introduction to genetic algorithms, August 15 1998.

[9] Ivan Zelinka, Zuzana Oplatkova, and Lars Nolle. I. Zelinka et al: Analytical programming ... Analytic programming – symbolic regression by means of arbitrary evolutionary algorithms, August 14 2008.

[10] Тырсин А.Н. Об эквивалентности знакового и наименьших модулей методов построения линейных моделей. *Обозрение прикладной и промышленной математики*, 12(4):879–880, 2005.

[11] Ю. Н. Павловский. *Имитационные модели и системы*. Фазис, 2000.

[12] А. Б. Иванов и др. В. И. Битюцков, М. И. Войцеховский. *Математическая энциклопедия*, volume 4. Советская Энциклопедия, 1984.

[13] D. W. Marquardt. An algorithm for least-squares estimation of non-linear parameters. *Journal of the Society of Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[14] J. J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In *G.A. Watson*, Lecture Notes in Mathematics 630, pages 105–116. Springer-Verlag, Berlin, 1978. Cited in Åke Björck's bibliography on least squares, which is available by anonymous ftp from `math.liu.se` in `pub/references`.

[15] T. Daglish, J. Hull, and W. Suo. Volatility surfaces: Theory, rules of thumb, and empirical evidence. *Quantitative Finance*, 7(5):507–524, 2007.

[16] В. В. Стрижов and Р. А. Сологуб. Индуктивное порождение регрессионных моделей предполагаемой волатильности для опционных торгов. *Вычислительные технологии*, 14(5):102–113, 2009.

Аппроксимация функции ошибки*

М. Е. Панов
panov.maxim@gmail.com
Московский физико-технический институт, ФУПМ, каф. «Интеллектуальные системы»

Введение

В работе рассматривается метод аппроксимации функции ошибки функцией многомерного нормального распределения. Рассматриваются случаи матрицы ковариации общего вида, диагональной матрицы ковариации, а также диагональной матрицы ковариации с равными значениями дисперсии. Для нормировки получившихся функций распределения используется аппроксимация Лапласа.

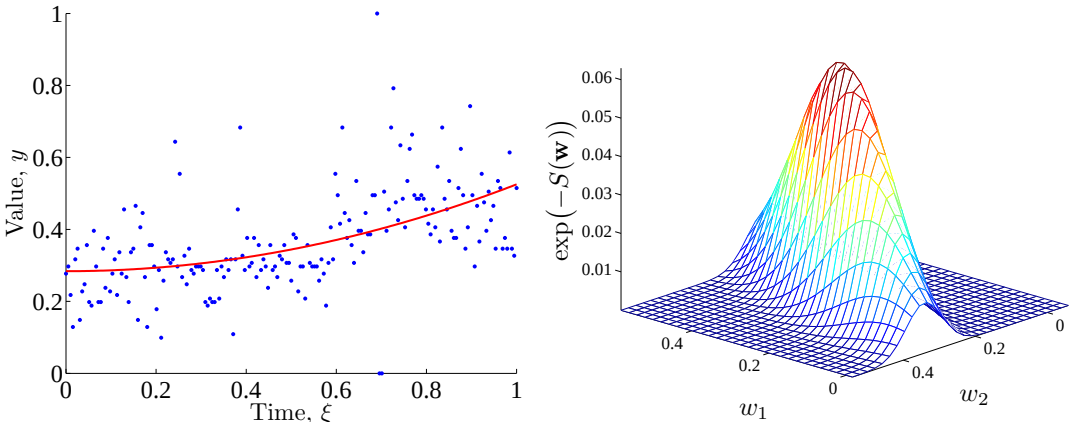


Рис. 1. Аппроксимация данных с помощью функции $f(x, w) = w_1 + w_2 * x^2$ при оптимальных значениях параметров $w = w_{MP}$ и ненормированная функции ошибки $exp(-S(w))$ в окрестности $w = w_{MP}$.

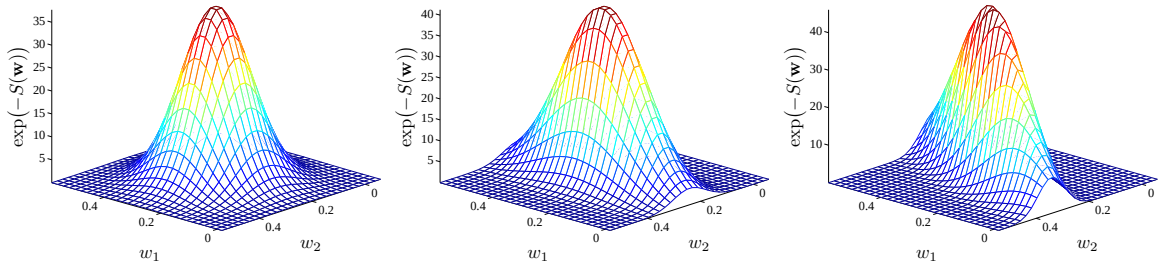


Рис. 2. Аппроксимации с помощью функции нормального распределения с различными видами матрицы ковариаций: с постоянной дисперсией, диагональной и общего вида.

Постановка задачи

Дана выборка $D = \{(x_i, y_i)\}_{i=1}^N$, где $x_i \in \mathbb{R}^n, i = 1, \ldots, N$ — вектора независимой переменной, а $y_i \in \mathbb{R}, i = 1, \ldots, N$ — значения зависимой переменной.

Научный руководитель В.В. Стрижов

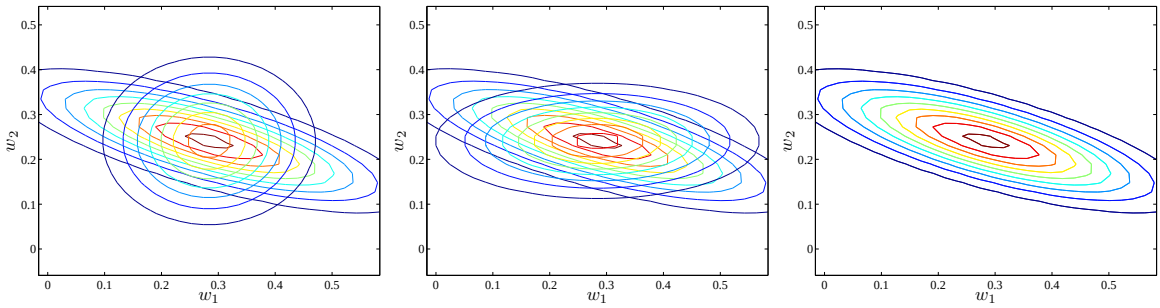


Рис. 3. Сравнение различных типов аппроксимации с реальной функцией ошибки.

Предполагается, что

$$y = f(x, w),$$

где $f(x, w)$ — некоторая параметрическая функция, $w \in W$ — вектор ее параметров. Предполагается, что задано апостериорное распределение параметров модели $p(w|D, f)$, которому соответствует функция ошибки $S(w)$: $p(w|D, f) = \frac{exp(-S(w))}{Z_S}$. Пусть $w_{MP} = \arg\max_w p(w|D, f)$ — наиболее вероятные параметры модели. Требуется найти аппроксимацию Лапласа для функции $p(w|D, f)$ в точке w_{MP}. Заметим, что в данной работе в качестве функции ошибки берется сумма квадратов ошибок аппроксимации $S(w) = \sum_{i=1}^{N}(y_i - f(x_i, w))^2$.

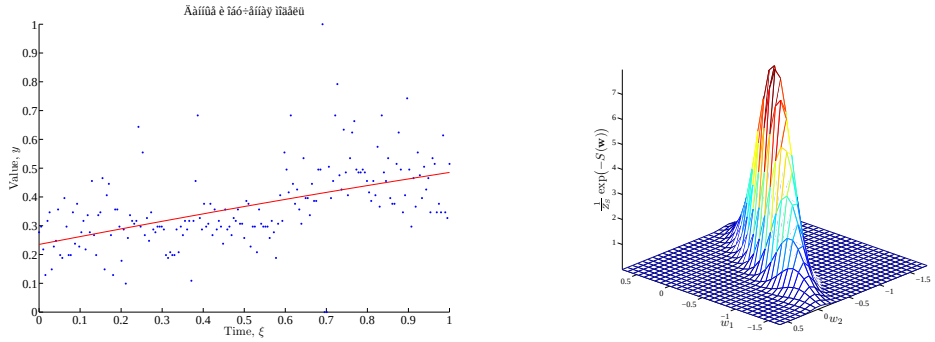


Рис. 4. Аппроксимация данных с помощью функции $f(x, w) = \frac{1-\exp(w_1+w_2*x)}{1+\exp(w_1+w_2*x)}$ при оптимальных значениях параметров $w = w_{MP}$ и ненормированная функции ошибки $exp(-S(w))$ в окрестности $w = w_{MP}$.

Описание алгоритма

Сначала находим оптимальные значения параметров модели w:

$$w_{MP} = \arg\max_w p(w|D, f).$$

Далее необходимо найти аппроксимацию Лапласа в точке w_{MP}: $p^*(w|k, A) = k * \exp(-(w - w_{MP})^T A(w - w_{MP}))$, где A — матрица, обратная к ковариационной матрице нормального распределения, а k — нормирующий коэффициент. Заметим, что в силу положительной определенности матрицы A ее можно представить в соответствии с разложением Холецкого: $A = LL^T$, где L — верхнетреугольная матрица. Параметризуем матрицу L следующим

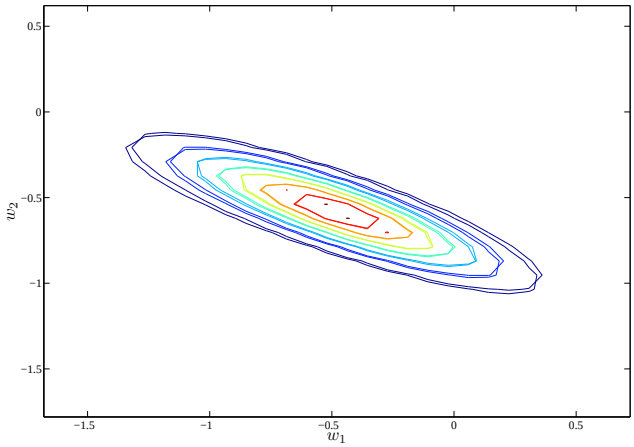


Рис. 5. Контурный график ее аппроксимации Лапласа с ковариационной матрицей общего вида.

образом: $L(i,j) = \begin{cases} exp(h_{ij}) & i = j, \\ sinh(h_{ij}) & j > i, \\ 0 & j < i, \end{cases}$ где $h_{ij} \in \mathbb{R}, i, j = 1, \ldots, N, j \geqslant i$. Также параметризуем нормирующий множитель $k = exp(h_0)$. Получаем, что $p^*(w|A,k) = p^*(w|h_{ij}, i, j = 1, \ldots, N, j \geqslant i, h_0)$. Построим обучающую выборку $D_S = (w_k, S(w_k)), k = 1, \ldots, N_S$, где точки w_k берутся равномерно из окрестности наиболее вероятных параметров w_{MP}, в которой мы хотим построить аппроксимацию. Для нахождения неизвестных параметров $h_{ij}, i, j = 1, \ldots, N, j \geqslant i, h_0$ минимизируем квадратичный критерий для точек обучающей выборки D_S:

$$\sum_{k=1}^{N_S} (S(w_k) - p^*(w_k|h_{ij}, h_0))^2 \to \min_{h_{ij}, h_0}. \tag{1}$$

Заметим, что получаемые в результате решения оптимизационной задачи [1] значения параметров могут существенно отличаться в зависимости от используемого для ее решения оптимизационного алгоритма. В данной работе рассматриваются два алгоритма оптимизации: Левенберг-Марквардт и Trust region.

После нахождения оптимальных значений параметров полученные распределения остается отнормировать в соответствии с аппроксимацией Лапласа: $Z_S = exp(-S(w_{MP})) * \sqrt{\frac{(2\pi)^n}{\det A}}$.

Вычислительный эксперимент: качество аппроксимации

В эксперименте в качестве обучающей выборки использовался временной ряд цен на хлеб из 195 точек. Для приближения использовалась модель линейной регрессии $f(x,w) = w_1 + w_2 * x^2$, а в качестве алгоритма оптимизации — алгоритм Левенберга-Марквардта [3, 4]. На картинках ниже графически представлены результаты.

Вычислительный эксперимент: устойчивость по начальным данным

Для сравнения устойчивости алгоритмов Левенберга-Марквардта и Trust region [2, 1] в качестве обучающей выборки использовался временной ряд цен на хлеб из 195 точек. Для приближения использовалась регрессионная модель $f(x,w) = \frac{1-\exp(w_1+w_2*x)}{1+\exp(w_1+w_2*x)}$. При таком

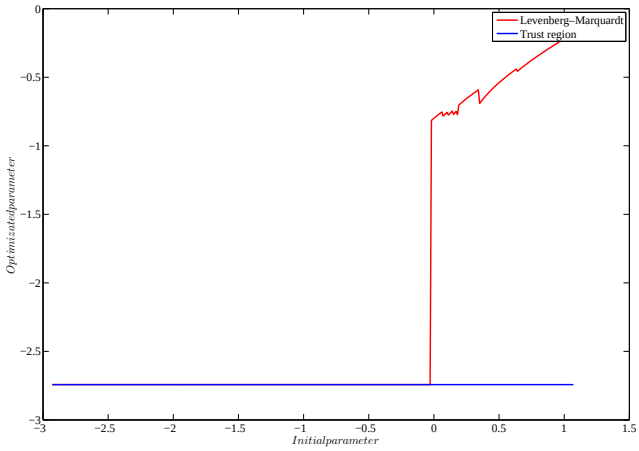


Рис. 6. Зависимость значения параметра h_0, полученного в результате оптимизации от его начального значения.

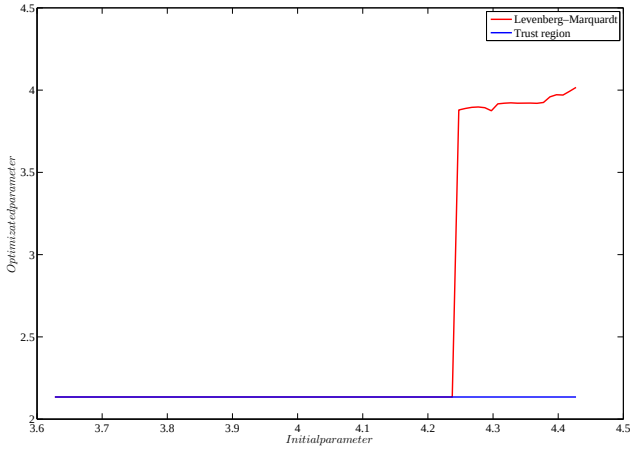


Рис. 7. Зависимость значения параметра h_{22}, полученного в результате оптимизации от его начального значения.

виде целевой функции вид функции ошибки в окрестности оптимума несколько отличается от нормального.

Рассматривалась зависимость оптимизированного значения параметров h_0 и h_{22} от начального значения.

Заключение

Функция ошибки в рассмотренных случаях хорошо аппроксимируется предложенным методом, причем качество аппроксимации возрастает с увеличением качества модели. Хорошее качество аппроксимации обусловлено тем, что функция ошибки в рассматриваемом примере принадлежит тому же классу, что и функция аппроксиматор, либо близка к нему. Сравнение алгоритмов оптимизации Левенберга-Марквардта и Trust region в применении к рассматриваемой задаче показало, что алгоритм Trust region гораздо более устойчив по начальным данным.

Литература

[1] *Coleman T., Li Y.* An interior, trust region approach for nonlinear minimization subject to bounds // *SIAM Journal on Optimization.* — 1994. — Vol. 6. — Pp. 418–445.

[2] *Coleman T., Li Y.* On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds // *Mathematical Programming.* — 1994. — Vol. 67, no. 2. — Pp. 189–224.

[3] *Levenberg K.* A method for the solution of certain problems in least-squares // *Quarterly Applied Math.* — 1944. — Vol. 2. — Pp. 164–168.

[4] *Marquardt D.* An algorithm for least-squares estimation of nonlinear parameters // *SIAM Journal Applied Math.* — 1963. — Vol. 11. — Pp. 431–441.

Выбор признаков в задачах логистической регрессии

К. С. Скипор
skiporkonstantin@mail.ru
Московский физико-технический институт

Предлагается и исследуется алгоритм отбора признаков для решения задач восстановления логистической регрессии. Алгоритм основан на методе наименьших углов для модели линейной регрессии с использованием дополнительно линеаризации функционала качества. Приводится математическое обоснование предложенного алгоритма. Работа алгоритма проиллюстрирована задачей изучения факторов риска ишемических заболеваний сердца.

Ключевые слова: *логистическая регрессия, выбор признаков, метод наименьших углов, линейное программирование*

Введение

В работе рассматривается отыскание из множества признаков такого его подмножества, для которого их линейная комбинация наиболее точно описывает данные. В 1966 году Дрейпером был предложен ступенчатый алгоритм выбора признаков (Forward Stagewise) [1, 2, 3]. На каждой итерации алгоритма выбирается признак, имеющий наибольшую проекцию на вектор ответов, после этого делается небольшое смещение текущего приближения функции регрессии в направлении выбранного признака. Среди полученных на каждой итерации моделей находится оптимальная, тем самым производится отбор признаков. Алгоритм Forward Selection [4] представляет собой модифицированную версию Forward Stagewise. Основное отличие заключается в выборе величины смещения. Смещение выбирается таким, чтобы максимизировать приращение функционала качества для выбранного признака.

В 1970 году Хоэрл и Кеннард предложили метод гребневой регрессии (Ridge Regression) [5], в котором использовался метод регуляризации [6]. Было введено дополнительное регуляризующее слагаемое в минимизируемый функционал; стало возможным улучшить устойчивость решения [7]. Еще один метод регуляризации, Лассо (The Lasso), был предложен Тибширани в 1996 году [8]. В нем вводится ограничение на L_1-норму вектора параметров модели, что приводит к обнулению части параметров модели и улучшению устойчивости решения. В модели логистической регрессии этот метод также называется L_1-regularized Logistic Regression [2].

В 2002 году Эфрон, Хасти, Джонстон и Тибширани предложили метод наименьших углов (Least Angle Regression) [9]. Изначально медод был предложен для линейных моделей, его реализацией является алгоритм последовательного добавления признаков LARS. На каждом шаге алгоритма признак выбирается таким образом, что вектор регрессионных остатков равноуголен [10] добавленным в модель признакам. Данный метод был предложен авторами для разрешения проблемы слишком быстрой сходимости к локальному оптимуму в многоэкстремальных задачах выбора признаков [11, 12, 13]. В 2004 году Мадиган и Ридгевэй предложили идею применения данного метода при использовании линеаризации для обобщенных линейных моделей, в частности, для модели логистической регрессии [14]. Реализация этой идеи лежит в основе написания данной работы.

Данная работа состоит из пяти частей. В разделе «Постановка задачи отбора признака» ставится задача отбора признаков в модели логистической регрессии, решаемая в

даной работе. Раздел «Описание алгоритма» разделен на три сегмента. Вначале кратко реферируются основные принципы работы алгоритма LARS для линейных моделей. Далее предлагается алгоритм последовательного добавления признаков в модели логистической регрессии LALR, решающий поставленную задачу. Отличие алгоритмов состоит в используемых функционалах качества. Предложенный алгоритм использует функционал качества, соответствующий бернуллиевской гипотезе порождения данных. После формального описания дается математическое обоснование предложенного алгоритма. Доказательство основных утверждений приводится в разделе «Приложение». В разделе «Вычислительные эксперименты» иллюстрируется работа предложенного алгоритма на модельных данных и на реальных данных «SAHD». Также работа предложенного алгоритма сравнивается с работой алгоритма Forward Stagewise.

Постановка задачи отбора признаков

Дана выборка $D = \{(\mathbf{x}^i, y^i)\}_{i=1}^m$, в которой i-й объект описывается строкой из n числовых признаков, $\mathbf{x}^i = (x_j^i)_{j=1}^n \in \mathbb{R}^n$ и метки класса $y^i \in \{0, 1\}$. Верхний индекс i указывает порядковый номер объекта выборки, нижний индекс j — порядковый номер признака. Векторы признаков $\mathbf{x}_j = (x_j^1, \ldots, x_j^i, \ldots, x_j^m)^T$ являются линейно независимыми свободными переменными, а вектор $\mathbf{y} = (y^1, \ldots, y^i, \ldots, y^m)^T$ является зависимой переменной. Без ограничения общности будем считать, что признаки $\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_n$ стандартизованы

$$\|\mathbf{x}_j\|_1 = \sum_{i=1}^m x_j^i = 0, \quad \|\mathbf{x}_j\|_2 = \sum_{i=1}^m (x_j^i)^2 = 1, \quad j = 1, \ldots, n. \tag{1}$$

Предполагается, что зависимая переменная y^i имеет распределение Бернулли. Для удобства описания алгоритма обозначим матрицу признаков $X = (\mathbf{x}_1 \ldots \mathbf{x}_j \ldots \mathbf{x}_n)$ и вектор параметров $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_j, \ldots, \beta_n)^T$. Принята модель логистической регрессии, согласно которой

$$\mathbf{y} = \boldsymbol{\sigma}(X, \boldsymbol{\beta}) + \varepsilon, \tag{2}$$

где $\boldsymbol{\sigma}(X, \boldsymbol{\beta})$ — сигмоидная функция

$$\boldsymbol{\sigma}(X, \boldsymbol{\beta}) = \frac{1}{1 + \exp(-X\boldsymbol{\beta})}. \tag{3}$$

Критерием качества модели назначен функционал логарифма правдоподобия

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^n \big(y^i \mathbf{x}^i \boldsymbol{\beta} - \ln(1 + \exp(\mathbf{x}^i \boldsymbol{\beta}))\big). \tag{4}$$

Требуется построить такой алгоритм последовательного добавления признаков, что на каждом шаге:

— определяются набор *активных признаков* с *активным множеством* индексов $\mathcal{A}$ и соответствующий набору ненулевой вектор параметров $\boldsymbol{\beta}_{\mathcal{A}}$, такой что $\boldsymbol{\beta}_{\mathcal{A}^c} = \mathbf{0}$, $\mathcal{A} \sqcup \mathcal{A}^c = \{1, \ldots, n\}$;
— набор *активных признаков* и вектор параметров $\boldsymbol{\beta}_{\mathcal{A}}$ доставляют максимум приращению логарифма правдоподобия ℓ;
— скорость роста функционала ℓ по любому активному признаку не меньше скорости роста по любому неактивному признаку.

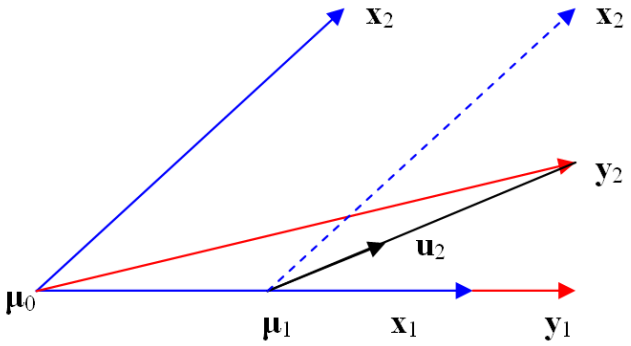


Рис. 1. *Пример работы алгоритма LARS в случае двух признаков* $\mathbf{x}_1$ *и* $\mathbf{x}_2$*. Пусть вектор* $\mathbf{y}_2$ *является проекцией вектора* $\mathbf{y}$ *на линейное подпространство* $\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2)$*. Назначим начальное приближение* $\boldsymbol{\mu}_0 = \mathbf{0}$*. Вектор регрессионных остатков* $\mathbf{y}_2 - \boldsymbol{\mu}_0$ *коррелирует с вектором* $\mathbf{x}_1$ *больше, чем с вектором* $\mathbf{x}_2$*. Первый шаг заключается в оценке* $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \gamma_1 \mathbf{x}_1$*. Скаляр* γ_1 *выбирается таким образом, что вектор остатков* $\mathbf{y}_2 - \boldsymbol{\mu}_1$ *делит пополам угол между векторами* $\mathbf{x}_1$ *и* $\mathbf{x}_2$*. Далее получаем значение* $\boldsymbol{\mu}_2 = \boldsymbol{\mu}_1 + \gamma_2 \mathbf{u}_2$*, где* $\mathbf{u}_2$ *- нормированный вектор, делящий этот угол пополам. Так как мы рассматриваем случай двух переменных, то* $\boldsymbol{\mu}_2 = \mathbf{y}_2$*.*

Описание алгоритма

Метод наименьших углов.

В данном подразделе предлагается краткое описание метода наименьших углов для задач линейной регрессии, см. [9]. Будем считать, что принята линейная модель

$$\mathbf{y} = \boldsymbol{\mu}(X, \boldsymbol{\beta}) + \varepsilon,$$

где функция регрессии $\boldsymbol{\mu}(X, \boldsymbol{\beta})$, представляющая собой приближение вектора y, имеет вид

$$\boldsymbol{\mu}(X, \boldsymbol{\beta}) = \sum_{j=1}^{n} \mathbf{x}_j \beta_j = X\boldsymbol{\beta}, \tag{5}$$

Критерием качества назначена среднеквадратичная ошибка

$$S(X, \boldsymbol{\beta}) = \|\mathbf{y} - \boldsymbol{\mu}(X, \boldsymbol{\beta})\|^2.$$

Требуется построить такой алгоритм последовательного добавления признаков, что на каждом шаге:

— определяются набор активных признаков с активным множеством индексов $\mathcal{A}$ и соответствующий набору ненулевой вектор параметров $\boldsymbol{\beta}_{\mathcal{A}}$, такой что $\boldsymbol{\beta}_{\mathcal{A}^c} = \mathbf{0}$, $\mathcal{A} \sqcup \mathcal{A}^c = \{1, \ldots, n\}$;

— набор активных признаков и вектор параметров $\boldsymbol{\beta}_{\mathcal{A}}$ доставляют наибольшую корреляцию векторов $\mathbf{y}$ и $\boldsymbol{\mu}$;

— абсолютная корреляция вектора регрессионных остатков $\mathbf{y} - \boldsymbol{\mu}$ с любым активным признаком не меньше абсолютной корреляции вектора остатков с любым неактивным признаком.

Для решения этой задачи был предложен метод наименьших углов, реализацией которого является алгоритм LARS [9]. Рассмотрим некоторый шаг алгоритма. Пусть на этом шаге определено множество индексов $\mathcal{A}$, которое соответствует выбранным до этого шага признакам, и некоторое приближение функции регрессии $\boldsymbol{\mu}_{\mathcal{A}}$. Корреляция c_j вектора остатков

$\mathbf{y} - \boldsymbol{\mu}_{\mathcal{A}}$ на некоторый признак $\mathbf{x}_j$ вычисляется как

$$c_j = \mathbf{x}_j^T(\mathbf{y} - \boldsymbol{\mu}_{\mathcal{A}}).$$

На первом шаге выбирается признак, имеющий наибольшую абсолютную корреляцию с вектором $\mathbf{y}$.

Далее вычисляется единичный вектор $\mathbf{u}$, лежащий на биссекторе выбранных признаков. Алгоритм смещает текущее приближение $\boldsymbol{\mu}_{\mathcal{A}}$ в направлении вектора $\mathbf{u}$,

$$\boldsymbol{\mu}_{\mathcal{A}_+} = \boldsymbol{\mu}_{\mathcal{A}} + \gamma\mathbf{u},$$

где γ — коэффициент смещения, который определяется из условия, что корреляция нового вектора остатков $\mathbf{y} - \boldsymbol{\mu}_{\mathcal{A}_+}$ на некоторый неактивный признак $\mathbf{x}_d$ будет равна корреляции на все активные признаки. Здесь $\mathcal{A}_+$ есть новое активное множество индексов $\mathcal{A} \cup \{d\}$. Смещение в направлении вектора $\mathbf{u}$ обеспечивает равенство корреляций вектора остатков $\mathbf{y} - \boldsymbol{\mu}_{\mathcal{A}_+}$ на выбранные признаки, или другими словами, обеспечивает равенство углов между вектором остатков и выбранными признаками.

Рис. 1 иллюстрирует работу алгоритма в случае $n = 2$ признаков, $X = (\mathbf{x}_1, \mathbf{x}_2)$.

В следующем подразделе описывается алгоритм отбора признаков для модели логистической регрессии, после чего будет дано математическое обоснование приведенного алгоритма.

Алгоритм LALR.

В настоящей работе предлагается новый алгоритм выбора признаков при восстановлении логистической регрессии — «Least Angle Logistic Regression (LALR)». Принята модель логистической регрессии (2), (3). Обозначим множество индексов параметров $\mathcal{I} = \{1, 2, \ldots, n\}$. Для некоторого подмножества индексов $\mathcal{A} \subseteq \mathcal{I}$, назовем его *активным множеством*, определим матрицу *активных признаков*

$$X_{\mathcal{A}} = (\cdots\ s_j\mathbf{x}_j\ \cdots)_{j\in\mathcal{A}}, \tag{6}$$

где s_j, назовем его *знаком корреляции*, принимает значения ± 1. Определим также матрицы разностей и сумм между активными признаками и некоторым фиксированным неактивным признаком $\mathbf{x}_d$, где $d \in \mathcal{A}^c$, в разбиении $\mathcal{A} \sqcup \mathcal{A}^c = \mathcal{I}$,

$$\begin{aligned} M_{d-} &= (\ \cdots\ \ s_j\mathbf{x}_j - s_d\mathbf{x}_d\ \ \cdots\ )_{j\in\mathcal{A}}, \\ M_{d+} &= (\ \cdots\ \ s_j\mathbf{x}_j + s_d\mathbf{x}_d\ \ \cdots\ )_{j\in\mathcal{A}}. \end{aligned} \tag{7}$$

Опишем алгоритм последовательного добавления признаков. Начальные значения положим

$$\boldsymbol{\mu} = \mathbf{0}, \quad \boldsymbol{\beta} = \mathbf{0}, \quad \mathcal{A} = \varnothing. \tag{8}$$

Рассмотрим некоторый шаг алгоритма. Пусть $\boldsymbol{\mu}_{\mathcal{A}}$ есть текущее приближение функции регрессии на этом шаге. Тогда вектор текущих корреляций между признаками и вектором регрессионных остатков $\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}})$ имеет вид:

$$\mathbf{c} = X^T\left(\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}})\right). \tag{9}$$

Положим знак корреляции

$$s_j = \operatorname{sign}(c_j) \qquad j \in \mathcal{I}. \tag{10}$$

Вычисляем матрицы $X_{\mathcal{A}}$, M_{d-} и M_{d-}, согласно (6) и (7), для $d \in \mathcal{A}^c$. Для удобства изложения введем матрицу весов объектов W, матрицы A_{d-}, A_{d+} и векторы $\mathbf{b}_{d-}$, $\mathbf{b}_{d+}$. Обозначим диагональную $m \times m$ матрицу W с элементами

$$W_{ii} = \sigma_i(\boldsymbol{\mu}_{\mathcal{A}})(1 - \sigma_i(\boldsymbol{\mu}_{\mathcal{A}})), \tag{11}$$

где i — номер объекта. Также обозначим матрицы A_{d-}, A_{d+} и векторы $\mathbf{b}_{d-}$, $\mathbf{b}_{d+}$

$$A_{d\pm} = M_{d\pm}^T W X_{\mathcal{A}}, \tag{12}$$

$$\mathbf{b}_{d\pm} = M_{d\pm}^T(\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}})), \tag{13}$$

для всех $d \in \mathcal{A}^c$. Двойной знак «$\pm$» используется для компактной записи двух выражений с «$+$» и «$-$».

Далее, используя введенные обозначения, вычисляем множество векторов Υ, которое, как будет доказано ниже, содержит оптимальный вектор коэффициентов,

$$\Upsilon = \{A_{d-}^{-1}\mathbf{b}_{d-}, A_{d+}^{-1}\mathbf{b}_{d+}\}_{d\in\mathcal{A}^c}. \tag{14}$$

В приложении показано, что в предположениях поставленной задачи матрица A всегда имеет обратную матрицу A^{-1}. Алгоритм обновляет текущее приближение функции регрессии $\boldsymbol{\mu}_{\mathcal{A}}$

$$\boldsymbol{\mu}_{\mathcal{A}_+} = \boldsymbol{\mu}_{\mathcal{A}} + X_{\mathcal{A}}\boldsymbol{\gamma}_{\mathcal{A}}, \tag{15}$$

где оптимальный вектор коэффициентов $\boldsymbol{\gamma}_{\mathcal{A}}$ определяется из условия

$$\boldsymbol{\gamma}_{\mathcal{A}} = \arg\min_{\boldsymbol{\gamma}\in\Upsilon}{}^{+}\big((\mathbf{s}_{\mathcal{A}} \circ \mathbf{c}_{\mathcal{A}})^T\boldsymbol{\gamma}\big), \tag{16}$$

"$\min^{+}$" означает, что минимум берется только из положительных значений минимизируемой функции. Операция «$\circ$» означает поэлементное (адамарово) умножение векторов. В другой интерпретации вектор $\mathbf{s}_{\mathcal{A}} \circ \mathbf{c}_{\mathcal{A}}$ есть вектор абсолютных корреляций с компонентами $|c_j|$.

Найденное решение $\boldsymbol{\gamma}_{\mathcal{A}}$ принадлежит множеству Υ, поэтому для некоторого индекса параметров $d^* \in \mathcal{A}^c$ выполнено либо $\boldsymbol{\gamma}_{\mathcal{A}} = A_{d^*-}^{-1}\mathbf{b}_{d^*-}$, либо $\boldsymbol{\gamma}_{\mathcal{A}} = A_{d^*+}^{-1}\mathbf{b}_{d^*+}$. Так определяется оптимальный индекс d^* соответствует найденному решению $\boldsymbol{\gamma}_{\mathcal{A}}$,

$$d^* = \arg\boldsymbol{\gamma}_{\mathcal{A}}. \tag{17}$$

В случае, когда $\mathcal{A} = \varnothing$, что соответствует первому шагу, d^* находится из условия максимума абсолютной корреляции:

$$d^* = \arg\max_{d\in\mathcal{I}}|c_d|. \tag{18}$$

Таким образом, определяется индекс d^*, соответствующий оптимальному добавляемому признаку $\mathbf{x}_{d^*}$, и обновляется активное множество индексов $\mathcal{A}$, путем добавления к нему d^*:

$$\mathcal{A}_+ = \mathcal{A} \cup \{d^*\}. \tag{19}$$

Также обновляется вектор параметров $\boldsymbol{\beta}$, с учетом знака корреляции (10):

$$\boldsymbol{\beta}_{\mathcal{A}} = \boldsymbol{\beta}_{\mathcal{A}} + \mathbf{s}_{\mathcal{A}} \circ \boldsymbol{\gamma}_{\mathcal{A}}.$$

Нижний индекс $\boldsymbol{\beta}_{\mathcal{A}}$ указывает, что изменяются только компоненты, соответствующие активным признакам. Этим завершается шаг алгоритма. Формула (16) дает приближенное значение вектора коэффициентов $\boldsymbol{\gamma}_{\mathcal{A}}$, поэтому алгоритм можно проитерировать для получения точного значения.

На последнем шаге, когда активный набор индексов соответствует полному, т.е $\mathcal{A} = \mathcal{I}$, все дополнительные условия на скорость роста функционала ℓ выполнены автоматически. Поэтому оптимальный вектор параметров находится из условия максимизации логарифма правдоподобия ℓ, с помощью итерационного метода наименьших квадратов с перевзвешиванием элементов (IRLS) [15].

Далее приводится обоснование используемых выше формул.

Обоснование алгоритма.

Стратегия построения метода наименьших углов, которая была использована для выбора признаков в линейной регрессии, применяется также и в логистической регрессии, но с использованием дополнительно линеаризации.

Пусть имеется некоторое активное множество $\mathcal{A}$ и пусть к тому же известно текущее приближение функции регрессии $\boldsymbol{\mu}_{\mathcal{A}}$.

Запишем логарифм правдоподобия (4) через функцию регрессии $\boldsymbol{\mu}_{\mathcal{A}}$, (5):

$$\ell(\boldsymbol{\mu}_{\mathcal{A}}) = \sum_{i=1}^{m} \big(y^i \mu_{\mathcal{A}}(\mathbf{x}^i) - \ln(1 + \exp(\mu_{\mathcal{A}}(\mathbf{x}^i)))\big). \tag{20}$$

Рассмотрим производную логарифма правдоподобия по некоторому признаку $\mathbf{x}_j$, обозначим ее c_j:

$$c_j = \frac{d}{d\gamma} \ell(\boldsymbol{\mu}_{\mathcal{A}} + \mathbf{x}_j \gamma)\Big|_{\gamma=0}, \tag{21}$$

откуда, пользуясь (3), получим:

$$c_j = \mathbf{x}_j^T \left(\mathbf{y} - \frac{\exp(\boldsymbol{\mu}_{\mathcal{A}})}{1 + \exp(\boldsymbol{\mu}_{\mathcal{A}}))} \right) = \mathbf{x}_j^T \left(\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}}) \right), \tag{22}$$

В матричном виде (22) принимает следующий вид

$$\mathbf{c} = X^T \left(\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}}) \right). \tag{23}$$

Замечание 1. Как и в случае LARS, вектор $\mathbf{c}$ есть вектор текущих корреляций векторов признаков и вектора остатков $\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}})$. Поэтому далее под вектором корреляций будем понимать вектор производных по направлению.

Обозначим знак корреляции, как это было сделано в (10),

$$s_j = \operatorname{sign}(c_j), \quad j \in \mathcal{I}. \tag{24}$$

Таким образом, определим матрицу активных признаков $X_{\mathcal{A}}$, согласно (6). Выразим новое приближение функции регрессии (15) через неизвестные коэффициенты $\boldsymbol{\gamma}$:

$$\boldsymbol{\mu}_{\mathcal{A}_+} = \boldsymbol{\mu}_{\mathcal{A}} + X_{\mathcal{A}} \boldsymbol{\gamma}. \tag{25}$$

Основная цель алгоритма заключается в поиске оптимального вектора коэффициентов $\boldsymbol{\gamma}$ и нового активного множества индексов $\mathcal{A}_+$ следующего шага.

Перейдем теперь к формальной интерпретации решаемой задачи. Под скоростью роста функционала $\ell(\boldsymbol{\mu}_{\mathcal{A}_+})$ по некоторому признаку понимается абсолютное значение производной функционала по этому признаку. Поэтому решаемая задача заключается в максимизации приращения логарифма правдоподобия (20)

$$\ell(\boldsymbol{\mu}_{\mathcal{A}_+}) - \ell(\boldsymbol{\mu}_{\mathcal{A}}) \to \max_{\boldsymbol{\gamma}}, \tag{26}$$

при условии, что абсолютная корреляция нового вектора остатков $\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}_+})$ на любой активный признак $\mathbf{x}_j$, $j \in \mathcal{A}$ не меньше абсолютной корреляции на любой неактивный признак $\mathbf{x}_d$, $d \in \mathcal{A}^c$, см. замечание 1. Запишем это условие через производную по направлению (21):

$$\Big|\frac{d}{d\alpha}\ell(\boldsymbol{\mu}_{\mathcal{A}_+} + \mathbf{x}_j\alpha)\Big|_{\alpha=0} \geqslant \Big|\frac{d}{d\alpha}\ell(\mathbf{x}_{\mathcal{A}_+} + \mathbf{x}_d\alpha)\Big|_{\alpha=0}, \tag{27}$$

для любых $j \in \mathcal{A}$ и $d \in \mathcal{A}^c$. Пользуясь обозначениями (12), (13) сформулируем лемму о линеаризации решаемой задачи.

Лемма 1. *Задача (26), (27) при линеаризации эквивалентна задаче линейного программирования:*

$$\begin{gathered}(\mathbf{s}_{\mathcal{A}} \circ \mathbf{c}_{\mathcal{A}})^T\boldsymbol{\gamma} \to \max_{\boldsymbol{\gamma}}, \\ \begin{cases} A_{d-}\boldsymbol{\gamma} \leqslant \mathbf{b}_{d-}, \\ A_{d+}\boldsymbol{\gamma} \leqslant \mathbf{b}_{d+}, \\ \forall d \in \mathcal{A}^c. \end{cases}\end{gathered} \tag{28}$$

Задачу линейного программирования (28) можно решать обычным симплекс-методом [16, 17], но с этим возрастает трудоемкость. Следующая теорема 4 позволяет существенно сократить количество опорных точек, которые могут являться решением задачи (28). Для доказательства теоремы 4 сформулируем некоторые вспомогательные утверждения.

Лемма 2. *Пусть векторы* $\mathbf{a}_1, \ldots, \mathbf{a}_k, \mathbf{a}_{k+1} \in \mathbb{R}^n$ *линейно независимы. Тогда векторы* $(\mathbf{a}_1 + \mathbf{a}_{k+1}), \ldots, (\mathbf{a}_k + \mathbf{a}_{k+1}), \mathbf{a}_{k+1}$ *также линейно независимы.*

Для использования леммы 3 определим понятие афинной зависимости векторов [16].

Определение 1. *Точки* $\mathbf{a}_1, \ldots, \mathbf{a}_k \in \mathbb{R}^n$ *называются аффинно зависимыми, если существуют* $\lambda_1, \ldots, \lambda_k$*, не равные нулю одновременно и такие, что*

$$\sum_{i=1}^{k} \lambda_i \mathbf{a}_i = 0, \qquad \sum_{i=1}^{k} \lambda_i = 0.$$

Лемма 3. *Пусть векторы* $\mathbf{a}_1, \ldots, \mathbf{a}_k, \mathbf{a}_{k+1} \in \mathbb{R}^n$ *линейно независимы. Обозначим матрицы*

$$A = (\mathbf{a}_1, \ldots, \mathbf{a}_k),\ A_+ = (\mathbf{a}_1, \ldots, \mathbf{a}_k, \mathbf{a}_{k+1}),\ C = (\mathbf{a}_1 - \mathbf{a}_{k+1}, \ldots, \mathbf{a}_k - \mathbf{a}_{k+1}).$$

Матрица A^TC *имеет полный ранг тогда и только тогда, когда столбцы матрицы* A^TA_+ *аффинно независимы.*

Лемма 3 используется при доказательстве существования множества Υ, определенного в (14).

С помощью следующей теоремы формулируется утверждение о решении задачи линейного программирования (28).

Теорема 4. *Если ЗЛП (28) имеет решение* $\boldsymbol{\gamma}^*$, *то*

$$\boldsymbol{\gamma}^* \in \Upsilon, \tag{29}$$

причем

$$\boldsymbol{\gamma}^* = \arg\min\nolimits^{+}_{\gamma\in\Upsilon}\{(\mathbf{s}_{\mathcal{A}} \circ \mathbf{c}_{\mathcal{A}})^T\boldsymbol{\gamma}\}; \tag{30}$$

где " $\min^{+}$" *означает, что минимум берется только из положительных значений.*

Следствие 1. *На каждом шаге алгоритма абсолютная корреляция текущего вектора остатков на любой активный признак при линеаризации одинакова и больше абсолютной корреляции на любой неактивный признак, т.е справедливо*

$$\begin{cases} s_i c_i = s_j c_j, & \forall i, j \in \mathcal{A}; \\ s_i c_i > s_d c_d, & \forall i \in \mathcal{A}, \forall d \in \mathcal{A}^c. \end{cases}$$

Следствие 1 представляет собой аналог основного свойства метода наименьших углов в линейных моделях: на каждом шаге вектор остатков лежит на биссекторе добавленных признаков.

Все доказательства приведенных утверждений приводятся в приложении.

Вычислительные эксперименты

Сравним предложенный алгоритм с описанным в [2, 3] итеративным алгоритмом Forward Stagewise. На каждом шаге алгоритм выбирает признак $\mathbf{x}_{j^*}$, имеющий наибольшую корреляцию c_{j^*} с текущим вектором остатков $\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu})$ и делает небольшое смещение γ текущего приближения в направлении выбранного признака $\mathbf{x}_{j^*}$,

$$j^* = \arg\max|c_j| \quad \text{и} \quad \boldsymbol{\mu} \to \boldsymbol{\mu} + \gamma\,\mathrm{sign}(c_{j^*})\mathbf{x}_{j^*}.$$

Чем меньше абсолютная величина смещения γ, тем точнее получается оценка параметров $\boldsymbol{\beta}$. Но с уменьшением смещения увеличивается количество шагов и, тем самым, возрастает время выполнения алгоритма.

Модельные данные.

Сгенерируем $m = 50$ объектов с пятью независимыми, нормально распределенными признаками $\mathbf{x}_1, \cdots, \mathbf{x}_5$, т.е $\mathbf{x}_i = (x_{i1}, \cdots, x_{i5}) \sim \mathcal{N}_5(\mathbf{0}, \mathbf{I})$. Примем модель

$$\mathbf{y} = \frac{1}{1 + \exp(-(\beta_0 + \mathbf{x}_1\beta_1 + \mathbf{x}_2\beta_2 + \mathbf{x}_3\beta_3))} + \varepsilon,$$

в качестве параметров $\boldsymbol{\beta}$ возьмем, например, вектор $(\beta_0, \beta_1, \beta_2, \beta_3)^T = (1, -2, 6, 3)^T$. В нашей модели признаки $\mathbf{x}_4$ и $\mathbf{x}_5$ являются шумовыми. Результатом работы алгоритма является последовательность весов признаков, выбираемых на каждом шаге. В данном случае алгоритм сделает шесть шагов.

В таблице (1) представлены результаты работы алгоритма. Первый столбец — номера признаков, первая строка — номер шага, а соответствующая ячейка таблицы — вес признака. Признаку с номером 0 соответствует константный признак. На рис. 2 показано сравнение оценок коэффициетов, полученных с помощью LALR и Forward Stagewise.

По полученным результатам можно сделать вывод, что последовательность выбираемых признаков и их весов согласуется с исходной моделью.

Данные «SAHD».

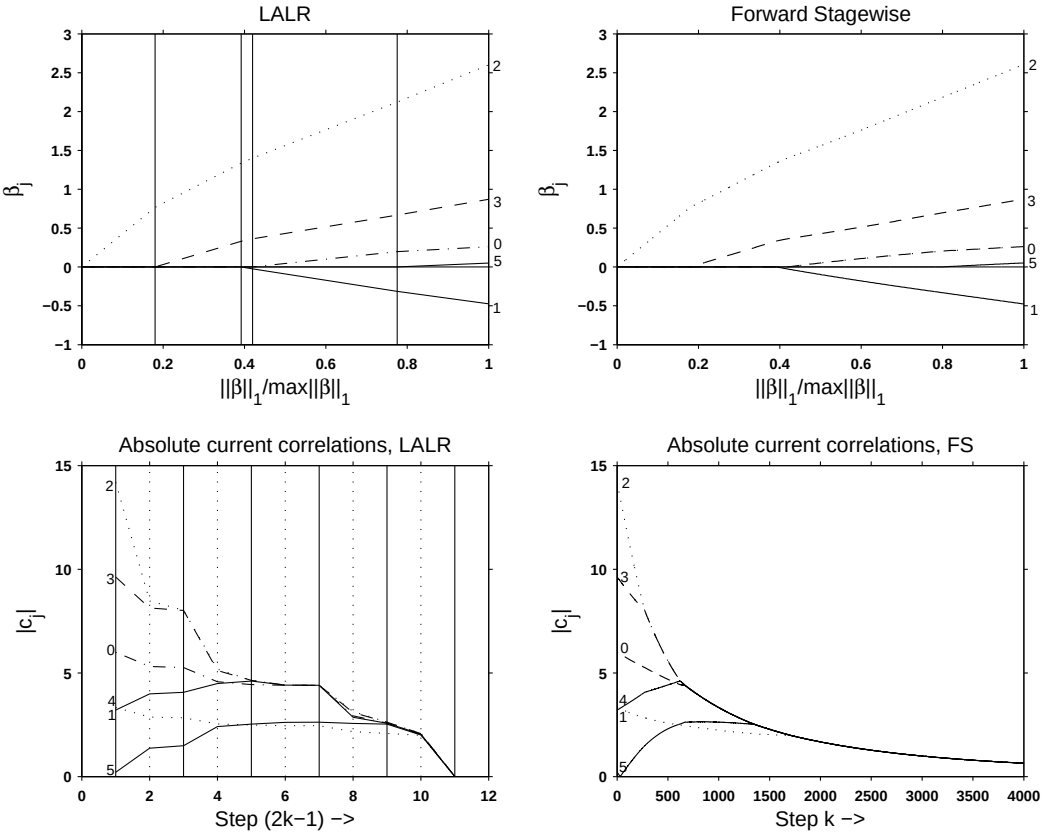


Рис. 2. Сравнение оценок коэффициентов для LALR и Forward Stagewise для модельных данных. Номера кривых соответствуют номерам признаков. Сплошные вертикальные линии обозначают шаги, а штриховые вертикальные — дополнительную итерацию для каждого шага.

Таблица 1. Результаты работы LALR

№	1	2	3	4	5	6
0	0	0	0	0.1969	0.2606	9.2868
1	0	0	-0.0250	-0.3142	-0.4733	-21.4689
2	0.7769	1.3359	1.4005	2.1215	2.5999	91.6048
3	0	0.3313	0.3615	0.6677	0.8713	36.5161
4	0	0	0	0	0	-8.2624
5	0	0	0	0	0.0513	1.6560

Проанализирована работа алгоритма на реальных данных «South African Heart Disease», см. [2]. Данные были впервые рассмотрены в [18]. Целью исследований являлось изучение факторов риска ишемических заболеваний сердца в районах с высокой заболеваемостью. Данные SAHD представляют собой сведения о физическом состоянии 462-х пациентов мужского пола белой расы возраста от 15 до 64 лет. Описание данных состоит из 9 признаков: $\mathbf{x}_1$ — sbp (sistolic blood pressure), $\mathbf{x}_2$ — tobacco, $\mathbf{x}_3$ — ldl (low-density lipoprotein), $\mathbf{x}_4$ — adiposity, $\mathbf{x}_5$ — famhist (family history), $\mathbf{x}_6$ — typea, $\mathbf{x}_7$ — obesity, $\mathbf{x}_8$ — alcohol, $\mathbf{x}_9$ — age; а также вектора меток класса chd: наличие — «1», или отсутствие — «0» инфаркта миокарда (MI) за время обследования. Перед использованием данные были стандартизованы согласно (1). На рис. 3 представлены результаты работы алгоритма.

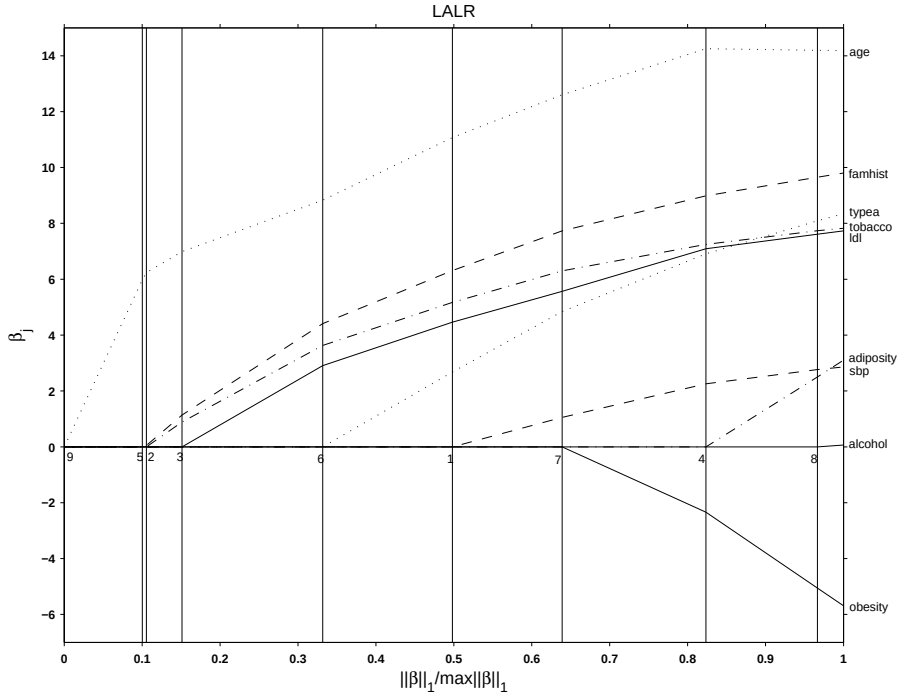


Рис. 3. Оценка коэффициентов алгоритма LALR для данных «South African Heart Disease». Вертикальные линии соответствуют шагам алгоритма.

Заключение

В данной работе предложен и исследован новый алгоритм LALR, решающий задачу отбора признаков в модели логистической регрессии. Разработан, исследован и математически обоснован алгоритм LALR, представляющий собой линеаризованный аналог алгоритма LARS для модели логистической регрессии. Проведена серия численных экспериментов на модельных и реальных данных «SAHD», результаты которых позволяют говорить об эффективности использования предложенного алгоритма.

Литература

[1] N. Draper and H. Smith. *Appied Regression Analysis*. Wiley, New York, 1966.

[2] T. Hastie, R. Tibshirani, and J. Friedman. *The ELemmaents of Statistical Learning: Data mining, Inference and Prediction*. Springer, New York, 2001.

[3] T. Hastie, J. Taylor, R. Tibshirani, and G. Walther. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1:1–29, 2007.

[4] R. R. Hocking. The analysis and selection of variables in linear regression. *Biometrics*, 32(1):1–49, 1976.

[5] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal probLemmas. *Technometrics*, 12(1):55–67, 1970.

[6] A. N. Tikhonov. Regularization of incorrectly posed probLemmas. *SMD*, 4(3):1624–1627, 1963.

[7] A. Bjorkstrom. Ridge regression and inverse probLemmas. Technical report, Stockholm University, 2001.

[8] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

[9] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[10] L. Lawson and R. J. Hanson. *Solving Least Squares ProbLemmas*. Englewood Cliffs: Prentice Hall, 1974.

[11] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

[12] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[13] Ye Jianming. On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93(441):120–131, Mar 1998.

[14] D. Madigan and G. Ridgeway. Discussion of least angle regression. *Annals of Statistics*, 32(2):465–469, 2004.

[15] D. B. Rubin. Iteratively reweighted least squares. *Encyclopedia of statistical sciences*, 4:272–275, 1983.

[16] А. Г. Сухарев, А. В. Тимохов, and В. В. Федоров. *Курс методов оптимизации*. Физматлит, 2005.

[17] А. Ф. Измаилов. *Численные методы оптимизации*. Физматлит, Москва, 2005.

[18] J. Rousseauw, J. du Plessis, A. Benade, P. Jordan, J. Kotze, P. Jooste, and J. Ferreira. Coronary risk factor screening in three rural communities. *South African Medical Journal*, 64:430–436, 1983.

Приложение к статье "Выбор признаков в задачах логистической регрессии"

Доказательство.[Леммы 1] Используя (20) и (23) разложим $\ell(\boldsymbol{\mu}_{\mathcal{A}_+})$ до первого члена,

$$\begin{aligned}\ell(\boldsymbol{\mu}_{\mathcal{A}_+}) = \ell(\boldsymbol{\mu}_{\mathcal{A}} + X_{\mathcal{A}}\boldsymbol{\gamma}) \approx \ell(\boldsymbol{\mu}_{\mathcal{A}}) + \sum_{j\in\mathcal{A}} \left(\frac{\partial \ell(\boldsymbol{\mu}_{\mathcal{A}_+})}{\partial \gamma_j}\right)_{\boldsymbol{\gamma}=\mathbf{0}} \gamma_j = \\ = \ell(\boldsymbol{\mu}_{\mathcal{A}}) + \sum_{j\in\mathcal{A}} s_j \mathbf{c}_j \gamma_j = \ell(\boldsymbol{\mu}_{\mathcal{A}}) + (\mathbf{s}_{\mathcal{A}} \circ \mathbf{c}_{\mathcal{A}})^T \boldsymbol{\gamma}.\end{aligned} \tag{31}$$

Далее, будем считать, что знак корреляции s_j не изменяется для любого $j \in \mathcal{A}$. Таким образом, используя выражение для корреляции (22), перепишем, раскрыв модуль, условия (27) в виде систем неравенств

$$(s_j \mathbf{x}_j - s_d \mathbf{x}_d)^T \left(\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}_+})\right) \geqslant 0, \tag{32}$$

если $s_d \mathbf{x}_d^T \left(\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}_+})\right) \geqslant 0$ и

$$(s_j \mathbf{x}_j + s_d \mathbf{x}_d)^T \left(\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}_+})\right) \geqslant 0, \tag{33}$$

если $s_d \mathbf{x}_d^T \left(\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}_+})\right) < 0$, для всех $j \in \mathcal{A}$ и $d \in \mathcal{A}^c$. Линеаризуем $\boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}_+})$ при достаточно малых γ, используя (11)

$$\begin{aligned}\boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}_+}) = \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}} + X_{\mathcal{A}}\boldsymbol{\gamma}) \approx \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}}) + \sum_{j\in\mathcal{A}} \left(\frac{\partial \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}_+})}{\partial \gamma_j}\right)_{\boldsymbol{\gamma}=\mathbf{0}} \gamma_j = \\ = \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}}) + \sum_{j\in\mathcal{A}} W_{jj} s_j \mathbf{x}_j \gamma_j = \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}}) + W X_{\mathcal{A}} \boldsymbol{\gamma}.\end{aligned} \tag{34}$$

Таким образом, пользуясь (34), системы неравенств (32) и (33) перепишутся в следующем виде:

$$(s_j \mathbf{x}_j \pm s_d \mathbf{x}_d)^T W X_{\mathcal{A}} \boldsymbol{\gamma} \leqslant (s_j \mathbf{x}_j \pm s_d \mathbf{x}_d)^T (\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}})), \tag{35}$$

для всех $j \in \mathcal{A}$ и $d \in \mathcal{A}^c$. Двойной знак «$\pm$» используется для компактной записи двух неравенств с «$+$» и «$-$». Как было введено ранее в (7), обозначим матрицы M_{d-} и M_{d+}:

$$M_{d\pm} = (\ \cdots \quad s_j \mathbf{x}_j \pm s_d \mathbf{x}_d \quad \cdots\ )_{j\in\mathcal{A}}, \tag{36}$$

для всех $d \in \mathcal{A}^c$. Тогда система (35) принимает следующий вид:

$$\left\{ \begin{array}{c} \vdots \\ M_{d\pm}^T W X_{\mathcal{A}} \boldsymbol{\gamma} \leqslant M_{d\pm}^T (\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}})), \\ \vdots \end{array} \right. \tag{37}$$

для всех $d \in \mathcal{A}^c$. Обозначим матрицы A_{d-}, A_{d+} и векторы $\mathbf{b}_{d-}$, $\mathbf{b}_{d+}$

$$A_{d\pm} = M_{d\pm}^T W X_{\mathcal{A}},$$

$$\mathbf{b}_{d\pm} = M_{d\pm}^T (\mathbf{y} - \boldsymbol{\sigma}(\boldsymbol{\mu}_{\mathcal{A}})),$$

тогда, используя (31) и (37), общая задача (26), (27) принимает линеаризованный вид (28). Что и требовалось доказать. ■

Доказательство.[Леммы 2] Докажем это от противного. Предположим, что векторы $(\mathbf{a}_1+$ $+\mathbf{a}_{k+1}), \ldots, (\mathbf{a}_k+\mathbf{a}_{k+1}), \mathbf{a}_{k+1}$ линейно зависимы. Тогда существуют $\gamma_1, \ldots, \gamma_{k+1}$ одновременно ненулевые, для которых справедливо

$$\gamma_1(\mathbf{a}_1 + \mathbf{a}_{k+1}) + \ldots + \gamma_k(\mathbf{a}_k + \mathbf{a}_{k+1}) + \gamma_{k+1}\mathbf{a}_{k+1} = 0.$$

Из этого следует, что

$$\gamma_1\mathbf{a}_1 + \ldots + \gamma_k\mathbf{a}_k + (\gamma_1 + \ldots + \gamma_k + \gamma_{k+1})\mathbf{a}_{k+1} = 0,$$

что противоречит линейной независимости векторов $\mathbf{a}_1, \ldots, \mathbf{a}_k, \mathbf{a}_{k+1}$. Поэтому векторы $(\mathbf{a}_1 + \mathbf{a}_{k+1}), \ldots, (\mathbf{a}_k + \mathbf{a}_{k+1}), \mathbf{a}_{k+1}$ линейно независимы. ■

Доказательство.[Леммы 3] Из линейной независимости векторов $\mathbf{a}_1, \ldots, \mathbf{a}_k, \mathbf{a}_{k+1}$ сразу следует, что матрица A имеет полный ранг по столбцам. Согласно лемме 2, матрица $(C|\mathbf{a}_{k+1})$ также имеет полный ранг по столбцам. Здесь знак $<< \mid >>$ обозначает присоединение вектора $\mathbf{a}_{k+1}$ к матрице C. Пользуясь свойствами ранга произвольной матрицы, получаем

$$\operatorname{rank}(A) = \operatorname{rank}(A^TA) = k \tag{38}$$

и аналогично

$$\operatorname{rank}\left((C|\mathbf{a}_{k+1})\right) = \operatorname{rank}\left((C|\mathbf{a}_{k+1})^T(C|\mathbf{a}_{k+1})\right) = k+1.$$

Из последнего заключаем, что матрица $(C|\mathbf{a}_{k+1})^T(C|\mathbf{a}_{k+1})$ является квадратной и полного ранга, а значит вектор-столбцы

$$(C|\mathbf{a}_{k+1})^T(\mathbf{a}_1 - \mathbf{a}_{k+1}), \ldots (C|\mathbf{a}_{k+1})^T(\mathbf{a}_k - \mathbf{a}_{k+1}), (C|\mathbf{a}_{k+1})^T\mathbf{a}_{k+1}$$

линейно независимы. Тогда, согласно лемме 2, векторы

$$(C|\mathbf{a}_{k+1})^T\mathbf{a}_1, \ldots, (C|\mathbf{a}_{k+1})^T\boldsymbol{a}_k$$

также линейно независимы. А из этого следует, что

$$\operatorname{rank}\left((C|\mathbf{a}_{k+1})^TA\right) = k.$$

Из последнего получаем, что матрица $A^T(C|\mathbf{a}_{k+1})$ имеет ровно k линейно независимых столбцов. Покажем, что при выполнении условия афинной независимости столбцов матрицы A^TA_+, вектор $A^T\mathbf{a}_{k+1}$ раскладывается в линейную комбинацию столбцов матрицы A^TC с ненулевыми коэффициентами.

Согласно (38), квадратная матрица A^TA имеет полный ранг, поэтому столбцы этой матрицы образуют базис в пространстве $\mathbb{R}^k$. А значит, любой ненулевой вектор этого пространства раскладывается по базису с ненулевыми коэффициентами единственным образом. Поэтому, для вектора $A^T\mathbf{a}_{k+1} \in \mathbb{R}^k$ существует и единственный ненулевой вектор $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)^T$, такой что

$$A^TA\boldsymbol{\lambda} = A^T\mathbf{a}_{k+1}. \tag{39}$$

Теперь, пусть выполнено условие афинной независимости столбцов матрицы A^TA_+. Это означает, что дополнительно выполняется условие

$$\eta = \sum_{i=1}^{k} \lambda_i \neq 1. \tag{40}$$

Покажем, что в этом случае вектор $A^T\mathbf{a}_{k+1}$ раскладывается по системе столбцов матрицы A^TC с коэффициентами

$$\boldsymbol{\xi} = \frac{1}{1-\eta}\boldsymbol{\lambda}. \tag{41}$$

Итак, пусть для некоторого $\boldsymbol{\xi}$ имеет место разложение

$$A^TC\boldsymbol{\xi} = A^T\mathbf{a}_{k+1}. \tag{42}$$

Преобразуем

$$A^TC = A^T(A - (\underbrace{\mathbf{a}_{k+1} \ \dots \ \mathbf{a}_{k+1}}_{k})) = A^TA - A^T\mathbf{a}_{k+1}\mathbf{1}^T.$$

Подставляя последнее в (42) получим $A^TA\boldsymbol{\xi} = A^T\mathbf{a}_{k+1}(1 + \mathbf{1}^T\boldsymbol{\xi})$. Делаем замену (39), $A^TA\boldsymbol{\xi} = A^TA\boldsymbol{\lambda}(1 + \mathbf{1}^T\boldsymbol{\xi})$. Откуда полагаем

$$\boldsymbol{\xi} = \boldsymbol{\lambda}(\underbrace{1 + \mathbf{1}^T\boldsymbol{\xi}}_{\text{число } \alpha}).$$

или подставляя $\boldsymbol{\xi} = \boldsymbol{\lambda}\alpha$ в предыдущее равенство, получим

$$\boldsymbol{\lambda}\alpha = \boldsymbol{\lambda}(1 + \underbrace{\mathbf{1}^T\boldsymbol{\lambda}}_{\eta}\alpha),$$

откуда $\alpha = \frac{1}{1-\eta}$. По условию (40) знаменатель дроби не обращается в 0. Таким образом, существует разложение (42) вектора $A^T\mathbf{a}_{k+1}$ по системе столбцов матрицы A^TC с ненулевыми коэффициентами (41). Поэтому

$$\operatorname{rank}(A^TC) = \operatorname{rank}\left(A^T(C|\mathbf{a}_{k+1})\right) = k,$$

т. е матрица A^TC имеет полный ранг.

Теперь, пусть столбцы матрицы A^TA_+ афинно зависимы. В этом случае, для разложения (39) дополнительно выполняется условие $\eta = \sum_{i=1}^{k}\lambda_i = 1$. Подставляя его в (39), получим

$$A^TA\boldsymbol{\lambda} = A^T\mathbf{a}_{k+1} = A^T\mathbf{a}_{k+1}\sum_{i=1}^{k}\lambda_i = A^T\mathbf{a}_{k+1}\mathbf{1}^T\boldsymbol{\lambda},$$

или, перенеся в одну сторону, имеем

$$A^T(A - \mathbf{a}_{k+1}\mathbf{1}^T)\boldsymbol{\lambda} = A^TC\boldsymbol{\lambda} = 0.$$

А полученная однородная система имеет нетривиальное решение $\boldsymbol{\lambda}$ тогда и только тогда, когда

$$\det(A^TC) = 0.$$

Таким образом, в этом случае матрица A^TC имеет неполный ранг. Что и требовалось доказать. ■

Доказательство.[Теоремы 4] Для доказательства воспользуемся методом математической индукции по шагам алгоритма.

1. База индукции. Возьмем в качестве базы первый шаг алгоритма, когда выбран первый активный признак $\mathbf{x}_i$. Матрицы $A_{d\pm}$ и векторы $\mathbf{b}_{d\pm}$ в этом случае представляют собой

действительные числа. Если решение ЗЛП существует, то оно достигается на границе области допустимых значений, т. е для некоторого d выполняется равенство $A_d\gamma^* = \mathbf{b}_d$, что эквивалентно выполнению условия (29). Здесь и далее, под $A_d\gamma^* = \mathbf{b}_d$ подразумевается выполнение $A_{d-}\gamma^* = \mathbf{b}_{d-}$, либо $A_{d+}\gamma^* = \mathbf{b}_{d+}$.

Покажем, что условие (30) также выполняется. Допустим, что это не так. Пусть $\gamma^* \in \Upsilon$ есть решение ЗЛП, тогда существует $d \in \mathcal{I}\backslash\{i\}$, для которого $A_d\gamma^* = \mathbf{b}_d$. Предположим, что существует $\tilde{\gamma} \in \Upsilon$, для которого выполнено $s_i c_i \tilde{\gamma} < s_i c_i \gamma^*$. Т. к. $\tilde{\gamma} \in \Upsilon$, то существует $\tilde{d} \in \mathcal{I}\backslash\{i\}$, для которого $A_{\tilde{d}}\tilde{\gamma} = \mathbf{b}_{\tilde{d}}$. Рассматриваются только значения $s_i c_i \gamma > 0$, которые соответствуют положительному приращению логарифма правдоподобия. А так как абсолютная корреляция $s_i c_i > 0$, то по предположению получаем, что $0 < \tilde{\gamma} < \gamma^*$. Из того, что вектор $\mathbf{x}_i$ имеет наибольшую абсолютную корреляцию с вектором остатков, следует $\mathbf{b}_{j\pm} = s_i c_i \pm s_j c_j > 0$, для $j \in \{d, \tilde{d}\}$. А это означает, что $A_{\tilde{d}} > 0$. Поэтому, учитывая, что γ^* есть решение, справедливо неравенство

$$\mathbf{b}_{\tilde{d}} = A_{\tilde{d}}\tilde{\gamma} < A_{\tilde{d}}\gamma^* \leqslant \mathbf{b}_{\tilde{d}}.$$

Получаем противоречие. Значит действительно, для данного шага условие (30) выполняется. Тем самым доказана справедливость теоремы для первого шага.

2. Допустим теперь, что утверждение верно для k-го шага алгоритма. Пусть $\boldsymbol{\gamma}^k$ есть решение ЗЛП на k-м шаге, тогда существует $d \in \mathcal{A}^c$, для которого верно $A_d\boldsymbol{\gamma}^k = \mathbf{b}_d$, причем для любого $j \in \mathcal{A}^c\backslash\{d\}$ справедливо $A_{j\pm}\boldsymbol{\gamma}^k < \mathbf{b}_{j\pm}$. Эти два условия есть не что иное, как линеаризованный вид условий (27). А это означает, что при линеаризации справедливо

$$\Big|\frac{d}{d\alpha}\ell(\boldsymbol{\mu}_{\mathcal{A}_+} + \mathbf{x}_i\alpha)\Big|_{\alpha=0} = \Big|\frac{d}{d\alpha}\ell(\boldsymbol{\mu}_{\mathcal{A}_+} + \mathbf{x}_d\alpha)\Big|_{\alpha=0}, \tag{43}$$

для любого $i \in \mathcal{A}$, и

$$\Big|\frac{d}{d\alpha}\ell(\boldsymbol{\mu}_{\mathcal{A}_+} + \mathbf{x}_i\alpha)\Big|_{\alpha=0} > \Big|\frac{d}{d\alpha}\ell(\boldsymbol{\mu}_{\mathcal{A}_+} + \mathbf{x}_j\alpha)\Big|_{\alpha=0}, \tag{44}$$

для любых $i \in \mathcal{A}$ и $j \in \mathcal{A}^c\backslash\{d\}$, где

$$\boldsymbol{\mu}_{\mathcal{A}_+} = \boldsymbol{\mu}_{\mathcal{A}} + X_{\mathcal{A}}\boldsymbol{\gamma}^k. \tag{45}$$

Теперь можно доказать утверждение для следующего шага.

3. Рассмотрим $(k+1)$-й шаг. В активное множество $\mathcal{A}$ добавился индекс d.

3.1 Сначала покажем что множество Υ не пусто. Для этого покажем существование обратных матриц A_d^{-1} для $d \in \mathcal{A}^c$. Покажем на примере матрицы A_{d-}. Условие (43), полученное на предыдущей итерации, означает, что при линеаризации для абсолютной корреляции справедливо $\mathbf{s}_{\mathcal{A}}\mathbf{c}_{\mathcal{A}} = c \cdot \mathbf{1}_{\mathcal{A}}$, где константа c есть значение абсолютной корреляции для активных признаков, а $\mathbf{1}_{\mathcal{A}}$ есть единичный вектор, размерности $|\mathcal{A}|$. Аналогично, условие (44) означает, что для любого $d \in \mathcal{A}^c$ верно $s_d c_d < c$. Далее, согласно (12), $A_{d-} = X_{\mathcal{A}}^T W X_{\mathcal{A}} - X_d^T W X_{\mathcal{A}}$ для любого $d \in \mathcal{A}^c$. Т. к признаки независимы, то матрица $X_{\mathcal{A}}^T W X_{\mathcal{A}}$ имеет полный ранг. Поэтому существует и единственный $\boldsymbol{\lambda} \neq \mathbf{0}$, для которого выполнено

$$X_{\mathcal{A}}^T W X_{\mathcal{A}}\boldsymbol{\lambda} = \mathbf{1}_{\mathcal{A}}, \tag{46}$$

причем, т. к $X_d = (s_d\mathbf{x}_d \ldots s_d\mathbf{x}_d)$, то

$$X_d^T W X_{\mathcal{A}}\boldsymbol{\lambda} = \tau \cdot \mathbf{1}_{\mathcal{A}}. \tag{47}$$

Откуда заключаем два важных результата.

3.1.1 Если для некоторого $d \in \mathcal{A}^c$ справедливо $\tau \neq 1$, то из (46) и (47) сразу следует, что матрица

$$\begin{pmatrix} X_{\mathcal{A}}^T W X_{\mathcal{A}} & \mathbf{1}_{\mathcal{A}} \\ s_d \mathbf{x}_d^T W X_{\mathcal{A}} & 1 \end{pmatrix}^T$$

имеет полный ранг. А это, в свою очередь, эквивалентно тому, что векторы

$$\{\ldots, X_{\mathcal{A}}^T W s_j \mathbf{x}_j, \ldots\}_{j \in \mathcal{A}} \quad \text{и} \quad X_{\mathcal{A}}^T W s_d \mathbf{x}_d$$

являются афинно независимыми. Теперь, применяя лемму 3 для векторов

$$\{\ldots, W^{\frac{1}{2}} s_j \mathbf{x}_j, \ldots\}_{j \in \mathcal{A}} \quad \text{и} \quad W^{\frac{1}{2}} s_d \mathbf{x}_d$$

получим, что матрица A_{d-} имеет полный ранг, а значит, для нее существует обратная. Для матрицы A_{d+} аналогичное условие $\tau \neq -1$.

3.1.2 Если для некоторого $d \in \mathcal{A}^c$ справедливо $\tau = 1$, то по лемме 3 получаем, что обратной матрицы A_{d-}^{-1} не существует, но в тоже время по лемме 3 существует обратная матрица A_{d+}^{-1}. Аналогично и для $\tau = -1$.

Тем самым показано, что множество Υ не пусто.

3.2 Покажем теперь выполнимость условия (29). Если решение ЗЛП существует, то оно достигается на границе области допустимых значений. Таким образом, решением ЗЛП является решение некоторой подсистемы ограничений максимального ранга. Докажем от противного, что множество Υ содержит решение.

Пусть $\boldsymbol{\gamma}$ есть решение ЗЛП, причем $\boldsymbol{\gamma}$ является решением некоторой подсистемы, отличной от $A_d \boldsymbol{\gamma} \leqslant \mathbf{b}_d$, $d \in \mathcal{A}^c$. Тогда для некоторых различных подсистем с индексами $d, d' \in \mathcal{A}^c$, $d \neq d'$, существуют строки с индексами $i, i' \in \mathcal{A}$, $i \neq i'$, для которых выполнено

$$\begin{cases} (\mathbf{x}_i \pm \mathbf{x}_d)^T W X_{\mathcal{A}} \boldsymbol{\gamma} = (c \pm s_d c_d) \cdot \mathbf{1}_{\mathcal{A}}, \\ (\mathbf{x}_{i'} \pm \mathbf{x}_{d'})^T W X_{\mathcal{A}} \boldsymbol{\gamma} = (c \pm s_{d'} c_{d'}) \cdot \mathbf{1}_{\mathcal{A}}. \end{cases}$$

3.2.1 Если справедливо

$$(s_i \mathbf{x}_i \pm s_{d'} \mathbf{x}_{d'})^T W X_{\mathcal{A}} \boldsymbol{\gamma} > (c \pm s_{d'} c_{d'}) \cdot \mathbf{1}_{\mathcal{A}},$$

то ограничение не выполнено и это противоречит тому, что $\boldsymbol{\gamma}$ решение ЗЛП.

3.2.2 Если справедливо

$$(s_i \mathbf{x}_i \pm s_{d'} \mathbf{x}_{d'})^T W X_{\mathcal{A}} \boldsymbol{\gamma} < (c \pm s_{d'} c_{d'}) \cdot \mathbf{1}_{\mathcal{A}},$$

то найдем

$$\begin{aligned} (s_{i'} \mathbf{x}_{i'} \pm s_d \mathbf{x}_d)^T W X_{\mathcal{A}} \boldsymbol{\gamma} &= (\mp s_{d'} \mathbf{x}_{d'}^T W X_{\mathcal{A}} \boldsymbol{\gamma} + (c \pm s_{d'} c_{d'}) \cdot \mathbf{1}_{\mathcal{A}}) - (\mp \mathbf{x}_i^T W X_{\mathcal{A}} \boldsymbol{\gamma} - (c \pm s_d c_d) \cdot \mathbf{1}_{\mathcal{A}}) \\ &= \pm (s_i \mathbf{x}_i - s_{d'} \mathbf{x}_{d'})^T W X_{\mathcal{A}} \boldsymbol{\gamma} + (c \pm s_{d'} c_{d'}) \cdot \mathbf{1}_{\mathcal{A}} + (c \pm s_d c_d) \cdot \mathbf{1}_{\mathcal{A}} \\ &> (c \pm s_d c_d) \cdot \mathbf{1}_{\mathcal{A}}. \end{aligned}$$

Ограничение не выполнено и это противоречит тому, что $\boldsymbol{\gamma}$ решение ЗЛП.

3.2.3 Последний случай, если справедливо

$$(s_i \mathbf{x}_i \pm s_{d'} \mathbf{x}_{d'})^T W X_{\mathcal{A}} \boldsymbol{\gamma} = (c \pm s_{d'} c_{d'}) \cdot \mathbf{1}_{\mathcal{A}},$$

то это означает, что для подсистем $d, d' \in \mathcal{A}^c, d \neq d'$ множество активных индексов, в которых справедливы ограничения-равенства, совпадают. Поэтому если рассмотреть некоторые k строк для подсистемы d в которых достигается равенство, то равенство будет достигаться в соответствующих строках для подсистемы d'. Причем разность любых двух строк для d будет равна разности соответствующих строк для d'. Отсюда можно сделать вывод, что ранг рассматриваемой матрицы равенств не больше $k+1$. Поэтому требуется, чтобы равенства достигались в каждой строке матриц A_d и $A_{d'}$, т. е на двух подсистемах сразу. В этом случае в активное множество придется добавлять сразу два индекса $\{d, d'\}$.

Отсюда следует выполнение (29).

3.3 Покажем выполнимость условия (30). Докажем это от противного.

Пусть $\boldsymbol{\gamma}_z \in \Upsilon$ есть решение ЗЛП. Индекс z соответствует номеру выбранного признака. Предположим, что условие (30) не выполняется. Тогда существует $\boldsymbol{\gamma}_d \in \Upsilon$, для которого верно $0 < (\mathbf{s}_\mathcal{A} \circ \mathbf{c}_\mathcal{A})^T \boldsymbol{\gamma}_d < (\mathbf{s}_\mathcal{A} \circ \mathbf{c}_\mathcal{A})^T \boldsymbol{\gamma}_z$. При линеаризации из последнего следует

$$0 < (\mathbf{s}_\mathcal{A} \circ \mathbf{c}_\mathcal{A})^T (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d) = c \cdot \mathbf{1}_\mathcal{A}^T (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d), \tag{48}$$

Из того, что $\boldsymbol{\gamma}_d \in \Upsilon$, получаем, что $A_d \boldsymbol{\gamma}_d = \mathbf{b}_d$. Будем использовать двойной знак, чтобы учесть возможные случаи. Итак, в матричном виде при линеаризации системы запишутся в виде $(X_\mathcal{A} \pm X_d)^T W X_\mathcal{A} \boldsymbol{\gamma}_d = (c \pm s_d c_d) \cdot \mathbf{1}_\mathcal{A}$. По предположению $\boldsymbol{\gamma}_z$ есть решение, поэтому справедливо $(X_\mathcal{A} \pm X_d)^T W X_\mathcal{A} \boldsymbol{\gamma}_z < (c \pm s_d c_d) \cdot \mathbf{1}_\mathcal{A}$. Отнимем от второго первое, получим

$$(X_\mathcal{A} \pm X_d)^T W X_\mathcal{A} (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d) < 0. \tag{49}$$

Нетрудно показать, что для выбранных $\boldsymbol{\gamma}_z$ и $\boldsymbol{\gamma}_d$ справедливо $X_\mathcal{A} W X_\mathcal{A} (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d) = \alpha \cdot \mathbf{1}_\mathcal{A}$. Покажем, что $\alpha > 0$. Т. к матрица $X_\mathcal{A} W X_\mathcal{A}$ положительно определена, то по определению

$$0 < (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d)^T X_\mathcal{A} W X_\mathcal{A} (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d) = \alpha \cdot \mathbf{1}_\mathcal{A} (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d).$$

И, пользуясь (48), получаем $\alpha > 0$.

Согласно (46), $\boldsymbol{\lambda} = \frac{\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d}{\alpha}$. Следовательно, из (47) получаем $X_d W X_\mathcal{A} (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d) = \alpha \cdot \tau \cdot \mathbf{1}_\mathcal{A}$, причем мы рассматриваем только те ограничения, для которых справедливо: $\tau < 1$ для матрицы A_{d-}, и $\tau > -1$ для матрицы A_{d+}, т. к в противном случае $\boldsymbol{\gamma}_z$ и $\boldsymbol{\gamma}_d$ не являются решениями. В итоге, получаем

$$\begin{cases} (X_\mathcal{A} - X_d)^T W X_\mathcal{A} (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d) = \alpha \cdot (1 - \tau) \cdot \mathbf{1}_\mathcal{A} > 0, & \tau < 1; \\ (X_\mathcal{A} + X_d)^T W X_\mathcal{A} (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d) = \alpha \cdot (1 + \tau) \cdot \mathbf{1}_\mathcal{A} > 0, & \tau > -1. \end{cases}$$

или просто

$$(X_\mathcal{A} \pm X_d)^T W X_\mathcal{A} (\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_d) > 0, \tag{50}$$

что противоречит (49). Значит действительно, для данного шага условие (30) выполняется.

Таким образом, утверждение справедливо и для $(k+1)$-го шага, а значит, согласно методу математической индукции, утверждение справедливо для любого шага. Что и требовалось доказать. ■

Оценка параметров смеси распределений

К. В. Павлов
kirill.pavlov@phystech.edu
Московский физико-технический институт

В работе рассматриваются способы построения смеси моделей и экспертов. Предлагается EM-алгоритм для совместного нахождения параметров моделей и их весов в смеси, а так же для нахождения параметров смеси обобщенных линейных моделей.

Ключевые слова: *смеси моделей, обобщенно-линейные модели, смеси экспертов.*

Введение

При решении задачи анализа данных строится модель — отображение известных характеристик объекта в неизвестные. Часто оказывается, что качество алгоритма можно улучшить с помощью комбинирования нескольких моделей [3, p. 653–676]. Например, можно обучить l моделей и в качестве ответа выводить усредненный ответ по всем моделям. Подобные комбинации моделей называются комитетами. Один из наиболее важных случаев комитета является бустинг. Алгоритмы в комитет добавляются последовательно и их параметры зависят от уже созданного на момент добавления комитета. Другим важным частным случаем комитета является смесь экспертов. В этом случае ответы алгоритмов взвешиваются в зависимости от области пространства, в которой находится объект. Рассмотрим способы построения композиций.

Общий подход к оценке параметров моделей

В случае, когда одной модели для описания данных не хватает, используют смеси моделей. Предполагается, что исходная зависимость $p(\boldsymbol{y} \mid \boldsymbol{x})$ выражается как композиция моделей $p(y \mid \boldsymbol{x}, \boldsymbol{w}_k)$ формулой:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{k=1}^{l} p(\boldsymbol{w}_k \mid \boldsymbol{x}) p(y \mid \boldsymbol{x}, \boldsymbol{w}_k) = \sum_{k=1}^{l} \pi_k p(y \mid \boldsymbol{x}, \boldsymbol{w}_k), \tag{1}$$

где $\pi_k = p(\boldsymbol{w}_k \mid \boldsymbol{x})$ — вероятность принадлежности к модели k. На π_k накладываются условия нормировки: вероятность каждой модели неотрицательна и сумма вероятностей равна единице.

$$\sum_{k=1}^{l} \pi_k = 1, \quad \pi_k \geqslant 0 \quad \forall k. \tag{2}$$

Далее предполагается, что объекты в выборке независимы и плотность совместного распределения преобразуется в произведение плотностей распределения каждого объекта.

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{k=1}^{l} \pi_k \prod_{i=1}^{n} p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}_k) = \prod_{i=1}^{n} \sum_{k=1}^{l} \pi_k p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}_k). \tag{3}$$

В формуле 3 произведена смена порядка суммирования перемножения. Используя принцип максимума правдоподобия, будет максимизировать $p(\boldsymbol{y} \mid \boldsymbol{x})$. Проще это делать, введя функцию правдоподобия $Q(\boldsymbol{w_1}, \ldots, \boldsymbol{w_l}, \boldsymbol{\pi})$ как логарифм плотности вероятности данных.

$$Q(\boldsymbol{w}^1,\ldots,\boldsymbol{w}^l,\boldsymbol{\pi}) = \ln p(\boldsymbol{y}\,|\,\boldsymbol{x}) = \sum_{i=1}^{m} \ln\left[\sum_{k=1}^{l} \pi_k p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}_k)\right]. \tag{4}$$

Обозначим через $p(y,\boldsymbol{w}_k\,|\,\boldsymbol{x})$ вероятность того, что объект $(\boldsymbol{x},y)$ был порожден компонентой $\boldsymbol{w}_k$, $\gamma_{ik} = p(\boldsymbol{w}_k|y^i,\boldsymbol{x}^i)$ — вероятность того, что i-объект порожден j - компонентой. Каждый объект был порожден какой-либо моделью, по формуле полной вероятности

$$\sum_{k=1}^{l} \gamma_{ik} = 1, \quad \forall i. \tag{5}$$

Для произвольного объекта $(\boldsymbol{x},y)$ вероятность его получения моделью w_k по формуле условной вероятности равна:

$$p(y,\boldsymbol{w}_k\,|\,\boldsymbol{x}) = p(\boldsymbol{w}_k\,|\,\boldsymbol{x})p(y\,|\,\boldsymbol{x},\boldsymbol{w}_k) \equiv \pi_k p(y\,|\,\boldsymbol{x},\boldsymbol{w}_k). \tag{6}$$

Подставим это равенство в формулу Байеса для γ_{ik}

$$\gamma_{ik} = \frac{\pi_k p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}_k)}{\sum_{s=1}^{l} \pi_s p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}_s)}. \tag{7}$$

Для определения параметров смеси необходимо решить задачу максимизации правдоподобия $Q(\boldsymbol{w}^1,\ldots,\boldsymbol{w}^l,\boldsymbol{\pi}) \to \max$, это можно сделать с использованием функции Лагранжа [1], которая имеет вид:

$$L = \sum_{i=1}^{m} \ln\left[\sum_{k=1}^{l} \pi_k p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^k)\right] - \lambda\left(\sum_{k=1}^{l} \pi_k - 1\right). \tag{8}$$

Необходимым условием экстремума функции является равенство нулю первых производных. Приравняем производную функции Лагранжа по π_k к нулю:

$$\frac{\partial L}{\partial \pi_k} = \sum_{i=1}^{m} \frac{p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^k)}{\sum_{s=1}^{l} \pi_s p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^s)} - \lambda = 0. \tag{9}$$

Умножим обе части равенства на π_k и просуммируем по $k = 1..l$

$$m = \sum_{k=1}^{l}\sum_{i=1}^{m} \frac{\pi_k p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^k)}{\sum_{s=1}^{l} \pi_s p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^s)} = \lambda \sum_{s=1}^{l} \pi_k = \lambda. \tag{10}$$

Получилось, что $\lambda = m$ необходимое условие минимума. В выражении для производной $\frac{\partial L}{\partial \pi_k}$ заменим λ на m и домножим обе части равенства на π_k:

$$\pi_k = \frac{1}{m}\sum_{i=1}^{m} \frac{\pi_k p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^k)}{\sum_{s=1}^{l} \pi_s p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^s)} = \frac{1}{m}\sum_{i=1}^{m} \gamma_{ik}. \tag{11}$$

Равенство 11 позволяет находить коэффициенты π_k смеси модели при известных γ_{ik}. Вычислим производную функции Лагранжа по параметрам k-й модели:

$$\frac{\partial L}{\partial \boldsymbol{w}^k} = \sum_{i=1}^{m} \frac{\pi_k \frac{\partial p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^k)}{\partial \boldsymbol{w}^k}}{\sum_{s=1}^{l} \pi_s p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^s)} = \sum_{i=1}^{m} \frac{\pi_k p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^k)}{\sum_{s=1}^{l} \pi_s p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^s)} \frac{\partial \ln p(y^i\,|\,\boldsymbol{x}^i,\boldsymbol{w}^k)}{\partial \boldsymbol{w}^k}. \tag{12}$$

Преобразуем выражение:

$$\frac{\partial L}{\partial \boldsymbol{w}^k} = \frac{\partial}{\partial \boldsymbol{w}^k} \sum_{i=1}^{m} \gamma_{ik} \ln p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}^k) = 0. \tag{13}$$

Полученное равенство совпадает с необходимым условием максимума в задаче максимизации взвешенного правдоподобия:

$$\sum_{i=1}^{m} \gamma_{ik} \ln p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}^k) \to \max_{\boldsymbol{w}^k}. \tag{14}$$

В общем случае задача оптимизации $Q(\boldsymbol{w}^1, \ldots, \boldsymbol{w}^l, \boldsymbol{\pi}) \to \max$ трудна, для её решения используют EM-алгоритм, заключающийся в итеративном повторении двух шагов. На E-шаге вычисляются ожидаемые значения вектора скрытых переменных γ_{ik} по текущему приближения параметров моделей $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_l)$. На M-шаге решается задача максимизации правдоподобия Q при начальном приближении параметров моделей и значений γ_{ik}.

E-шагу соответствует выражение

$$\gamma_{ik} = \frac{\pi_k p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}_k)}{\sum_{s=1}^{l} \pi_s p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}_s)}. \tag{15}$$

M-шаг заключается в оптимизации параметров распределений.

$$Q(\boldsymbol{w}^1, \ldots, \boldsymbol{w}^l \mid \boldsymbol{\pi}) \to max \tag{16}$$

Формула на M-шаге может упроститься для случая конкретного распределения. Для упрощения дальнейших рассуждений введем обозначения

$$G = (\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_l) = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1l} \\ \vdots & \ddots & \vdots \\ \gamma_{m1} & \cdots & \gamma_{ml} \end{pmatrix}, \quad G_k = \begin{pmatrix} \gamma_{1k} & & 0 \\ & \ddots & \\ 0 & & \gamma_{mk} \end{pmatrix}. \tag{17}$$

Перейдем к рассмотрению линейный и обобщенных линейных моделей.

Оценка параметров смеси линейных моделей

Линейная модель имеет вид:

$$\boldsymbol{y} = X\boldsymbol{w} + \boldsymbol{\varepsilon}, \tag{18}$$

где $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, B)$ — вектор нормально распределенных ошибок. В данной постановке вектор $\boldsymbol{y}$ является нормальным с математическим ожиданием $\mathsf{E}(y \mid \boldsymbol{x}) = \mu = \boldsymbol{x}^\intercal \boldsymbol{w}$, и корреляционной матрицей B. Плотность распределения $\boldsymbol{y}$ задается формулой:

$$p(\boldsymbol{y} \mid X, \boldsymbol{w}) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|\det B|}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - X\boldsymbol{w})^\intercal B (\boldsymbol{y} - X\boldsymbol{w})\right). \tag{19}$$

Применим для задачи описанный EM-алгоритм.Шаг E сводится к применению формулы 15, а шаг M алгоритма принимает следующий вид:

$$G_k \ln \left[\frac{1}{(2\pi)^{\frac{n}{2}} \sqrt{|\det B|}}\right] - \frac{1}{2}\left(G_k (\boldsymbol{y} - X\boldsymbol{w})^\intercal B (\boldsymbol{y} - X\boldsymbol{w})\right) \to \max_{\boldsymbol{w}} \tag{20}$$

Первое слагаемое не зависит от $\boldsymbol{w}_k$, его можно не учитывать. Преобразование второго слагаемого дает

$$\frac{1}{2}\boldsymbol{w}^\intercal X^\intercal G_k B X \boldsymbol{w} - \boldsymbol{w}^\intercal X^\intercal G_k B \boldsymbol{y} \to \min_{\boldsymbol{w}} \tag{21}$$

Задача квадратична по $\boldsymbol{w}$, решение находится аналитически

$$\boldsymbol{w}^* = (X^\intercal G_k B X)^{-1} G_k B X \boldsymbol{y}. \tag{22}$$

Оценка параметров смеси обобщенно-линейных моделей

В случае обобщенных линейный моделей функция плотности распределения имеет вид

$$p(\boldsymbol{y} \mid \boldsymbol{\theta}) = \exp\left(\boldsymbol{T}(\boldsymbol{y})^\intercal \boldsymbol{\eta}(\boldsymbol{\theta}) - b(\boldsymbol{\theta}) + c(\boldsymbol{y})\right). \tag{23}$$

M-шаг алгоритма сводится к максимизации

$$\boldsymbol{T}(\boldsymbol{y})^\intercal G_k \boldsymbol{\eta}(\boldsymbol{\theta}) - b(G_k\boldsymbol{\theta}) + c(G_k\boldsymbol{y}) \to \max_{\boldsymbol{\theta}}. \tag{24}$$

Последнее слагаемое не зависит от параметров модели θ, что позволяет упростить функционал

$$\boldsymbol{T}(\boldsymbol{y})^\intercal G_k \boldsymbol{\eta}(\boldsymbol{\theta}) - b(G_k\boldsymbol{\theta}) \to \max_{\boldsymbol{\theta}}. \tag{25}$$

Дальнейшая минимизация зависит от конкретного семейства из обобщенного класса распределений.

Оценка параметров смеси экспертов

Понятие смеси экспертов было введено Джорданом и Якобсом в 1991г [2]. Предполагается, что параметры смеси π являются функциями от объекта, т.е.

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{k=1}^{l} \pi_k(\boldsymbol{x}) p(y \mid \boldsymbol{x}, \boldsymbol{w}_k). \tag{26}$$

Компоненты $\pi_k(\boldsymbol{x})$ называются функциями селективности, а $p(y \mid \boldsymbol{x}, \boldsymbol{w}_k)$ экспертами. Функция селективности отвечает за компетентность эксперта в определенной области.

Оказывается [4], что наличие функции компетенции допускает решение задачи с помощью EM-алгоритма, причем, E-шаг остается прежним:

$$\gamma_{ik} = \frac{\pi_k(\boldsymbol{x}^i) p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}_k)}{\sum_{s=1}^{l} \pi_s(\boldsymbol{x}^i) p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}_s)}. \tag{27}$$

M-шаг принимает вид:

$$\pi_k = \frac{1}{m}\sum_{i=1}^{m} \gamma_{ik}. \tag{28}$$

$$\sum_{i=1}^{m} \gamma_{ik}(\boldsymbol{x}^i) \ln p(y^i \mid \boldsymbol{x}^i, \boldsymbol{w}^k) \to \max_{\boldsymbol{w}^k}. \tag{29}$$

Уравнение 29 можно решить с помощью метода итеративно перевзвешенных наименьших квадратов (IRLS).

Литература

[1] *Воронцов К. В.* Курс лекций: Линейные методы классификации. — 2009. — 01. `http://www.machinelearning.ru/wiki/images/6/68/Voron-ML-Lin.pdf`.

[2] Adaptive mixtures of local experts / R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton // *Neural Computation*. — 1991. — no. 3. — Pp. 79–87.

[3] *Bishop C. M.* Pattern Recognition and Machine Learning. — Springer, Series: Information Science and Statistics, 2006. — 740 pp.

[4] *Jordan M. I., Jacobs R. A.* Hierarchical mixtures of experts and the EM algorithm // *Neural Computation*. — 1994. — no. 6. — Pp. 181–214. `citeseer.ist.psu.edu/article/jordan94hierarchical.html`.

Многоклассовый прогноз вероятности наступления инфаркта*

А. П. Мотренко
pastt.petrovna@gmail.com
Московский физико-технический институт, ФУПМ, кафедра "Интеллектуальные системы"

В работе описан алгоритм, позволяющий классифицировать четыре группы пациентов: перенесших инфаркт; больных, имеющих предрасположенность к инфаркту и здоровых пациентов двух групп. Признаками для определения состояния пациента служат измерения концентрации белков в крови. Одной из задач работы является выбор набора маркеров, оптимального для разделения между собой соответствующих групп. Классификация осуществляется по принципу «каждый против каждого», то есть решаются задачи классификации всевозможных пар групп. В силу высокой стоимости анализа крови, объемы данных невелики, поэтому одним из результатов исследования является оценка необходимого объема выборки пациентов.

Ключевые слова: *логистическая регрессия, многоклассовая классификация, выбор признаков, оценка необходимого объема выборки, расстояние Кульбака-Лейблера.*

Введение

При выборе линейных моделей, включающих относительно небольшое количество признаков, решается задача оценки совместной сложности модели и оценки необходимого числа параметров объектов. Особенность исследуемой задачи в том, что стоимость получения выборки высока. Поэтому часть работы посвящена развитию методов оценки необходимого объема выборки по измеренным данным, сопряженному с задачей выбора моделей.

Заболевания сердечно-сосудистой системы могут протекать, не проявляясь клинически. Тем не менее, обнаружение нарушений, связанных с работой сердца, по косвенным признакам, или биомаркерам, вполне возможно [3]. Традиционными маркерами являются возраст, давление крови и уровень холестерина; существуют и другие показатели, например, определенные группы генов [7]. В данной работе предлагается использовать в качестве маркеров концентрации белков в клетках крови. Разделение пациентов по состоянию здоровья на четыре группы: больные, перенесшие инфаркт; больные, имеющие расположенность к инфаркту и здоровые двух типов приводит к задаче многоклассового прогнозирования. Такую задачу можно свести к задаче двуклассовой классификации, используя один из следующих подходов:

1) один против всех. Данный подход заключается в следующем: выделяем один группу пациентов как отдельный класс, а все остальные группы объединяем во второй класс и решаем, таким образом, задачу выделения определенной группы.
2) каждый против каждого. В этом случае перебираются все возможные пары групп.

Другим словами, в первом случае ставится вопрос «относится ли пациент к данной группе?», во втором — «к какому из двух данных групп он принадлежит с большей вероятностью?». Комбинация этих подходов приводит к еще одному способу: разделив каким-либо образом все множество групп на два непересекающихся подмножества, образовать два класса. Например, отличать две имеющиеся группы больных от двух групп здоровых

Научный руководитель В. В. Стрижов

пациентов. Выбор каждой стратегии зависит от конкретных задач; в работе приводится решение задачи с использованием второй стратегии, так как она дает наиболее подробные результаты. Обратим внимание на различие понятий «класс» и «группа». Под группой везде далее будем понимать группу, определяющую состояние здоровья пациента. В свою очередь, класс — понятие, связанное с задачей классификации, он может состоять как из одной, так и из нескольких групп пациентов.

В работе решаются задачи многоклассовой классификации с выбором признаков и оценки минимального объема выборки, достаточного для проведения классификации. Первая часть работы посвящена отбору наиболее информативных признаков, т.е выбору набора признаков, наилучшим образом разделяющего классы. На практике снижение количества измеряемых признаков диагностируемых пациентов приводит к уменьшению финансовых затрат на получение признаков и позволяет увеличить количество исследуемых пациентов, то есть объем выборки.

В работе рассматривается задача разделения пар классов. Предполагается, что число измеряемых признаков избыточно. Задача состоит в отыскании оптимального набора признаков, эффективно разделяющего между собой классы пациентов.

Для каждой пары групп решается задача логистической регрессии [4]. В ее основе лежит предположение о биномиальном распределении независимой переменной с оценкой параметров функции регрессии по методу Ньютона-Рафсона. Выбор признаков в логистической регрессии производится с помощью шаговой регрессии [6] или полного перебора. В данной работе используется полный перебор, т.к он дает экспертам уверенность в том, что рассмотрены все возможные сочетания признаков при выборе модели, а в как функционал качества используется площадь под графиком ROC-кривой [10].

Во второй части работы оценивается минимальный объем выборки, необходимый для проведения вычислительного эксперимента. Для оценки применяются следующие методы: метод доверительных интервалов [9], метод скользящего контроля [8], а также используется расстояние Кульбака-Лейблера [2] для сравнения предполагаемых распределений на различных подвыборках.

Задача классификации

Дана выборка $D = \mathbf{x}_i, y_i, i = 1, \ldots, m$ объектов (пациентов), каждый из которых описывается n признаками (биомаркерами) и принадлежит одному из двух классов $y_i \in \{0, 1\}$. Объединим признаки в столбцы $\boldsymbol{\chi}_j \in \mathbb{R}^m, j = 1, \ldots, n$ и составим из них матрицу $X = [\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_n] = \left[\mathbf{x}_1^T, \ldots, \mathbf{x}_m^T\right]^T$. Рассмотрим задачу логистической регрессии. Пусть случайная величина y имеет распределение Бернулли с параметром p, $y \sim B(p, 1-p)$, тогда

$$y = \begin{cases} 1, & p; \\ 0, & 1-p. \end{cases} \tag{1}$$

Функция плотности $p(y|p)$ в таком случае имеет вид

$$p(y|p) = p^y(1-p)^{1-y}. \tag{2}$$

В логистической регрессии предполагается, что вектор ответов $\mathbf{y} = [y_1, \ldots, y_m]$ — бернуллиевский случайный вектор с независимыми компонентами $y_i \sim \mathcal{B}(p_i, 1-p_i)$ и функцией плотности

$$p(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^{m} p_i^{y_i}(1-p_i)^{1-y_i}. \tag{3}$$

Определим функцию ошибки следующим образом:

$$E(\mathbf{w}) = -\ln p(\mathbf{y}|\mathbf{w}) = -\sum_{i=1}^{m} y_i \ln p_i + (1-y_i)\ln(1-p_i). \tag{4}$$

Другими словами, функция штрафа есть логарифм плотности, или функции правдоподобия, со знаком минус. В случае логистической регрессии

$$p_i = \frac{1}{1+exp(-\mathbf{x}_i^T\mathbf{w})} = \sigma(\mathbf{x}_i^T\mathbf{w}) \equiv \sigma_i. \tag{5}$$

Воспользовавшись тождеством $\frac{d\sigma(\theta)}{d\theta} = \sigma(1-\sigma)$, вычислим градиент функции $E(\mathbf{w})$

$$\nabla E(\mathbf{w}) = -\sum_{i=1}^{m} (y_i(1-\sigma_i) - (1-y_i)\sigma_i)\,\mathbf{x}_i = \sum_{i=1}^{m}(\sigma_i - y_i)\mathbf{x}_i = X^T(\sigma - \mathbf{y}),$$

где $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m)^T$. Оценка параметров осуществляется по схеме Ньютона-Рафсона. Введем обозначение Σ — диагональная матрица с элементами $\Sigma_{ii} = \sigma_i(1-\sigma_i), i = 1, \ldots, m$. Пусть параметры оцениваются на множестве $\mathcal{W}$. В качестве начального значения $\mathbf{w}_0$ возьмем

$$\mathbf{w}_0 = \arg\min_{\mathbf{w}\in\mathcal{W}} E(\mathbf{w}).$$

Тогда итеративная оценка параметров логистической регрессии (5) имеет вид

$$\mathbf{w}^{k+1} = \mathbf{w}^k - (X^T\Sigma X)^{-1}X^T(\boldsymbol{\sigma} - \mathbf{y}) = (X^T\Sigma X)^{-1}X^T\Sigma(X\mathbf{w}^k - \Sigma^{-1}(\boldsymbol{\sigma} - \mathbf{y})). \tag{6}$$

Процедура оценки параметров повторяется, пока норма разности $\left\|\mathbf{w}^{k+1} - \mathbf{w}^k\right\|$ не станет достаточно мала.

Алгоритм классификации имеет вид:

$$a(\mathbf{x}) = \operatorname{sign}\left(\sigma(\mathbf{x}, \mathbf{w}) - \sigma_0\right), \tag{7}$$

где σ_0 — задаваемое пороговое значение функции регрессии, о выборе σ_0 будет рассказано позже. Для контроля за качеством классификации можно использовать следующий функционал:

$$Q = (1 - TPR)^2 + FPR^2,$$

где

$$TPR = \frac{1}{m}\sum_{i=1}^{m}[a(\mathbf{x}_i) = 1][y_i = 1]$$

(*true positive rate*) — доля элементов выборки, правильно классифицированных в пользу заданного класса;

$$FPR = \frac{1}{m}\sum_{i=1}^{m}[a(\mathbf{x}_i) = 1][y_i = 0]$$

(*false positive rate*) — доля ошибочно классифицированных в пользу данного класса элементов выборки. Здесь используется обозначение индикаторной функции:

$$[y = 1] = \begin{cases} 1, & y = 1; \\ 0, & y \neq 1. \end{cases} \tag{8}$$

Таким образом, алгоритм тем лучше разделяет классы, чем меньше значение функционала Q. В данной работе используется альтернативный критерий. Отложив на графике по оси абсцисс значения FPR, а по оси ординат — TPR, получим так называемую ROC-кривую, каждая точка которой соответствует некоторому значению σ_0. В алгоритме (7) используется значение σ_0, отвечающее наибольшему расстоянию точки ROC-кривой до линии $TPR = FPR$. В качестве максимизируемого функционала будем использовать площадь под кривой, AUC (area under curve). Выбрав для каждого $\mathbf{x}$ пороговое значение σ_0 равным $\sigma(\mathbf{x}, \mathbf{w})$, вычислим TPR и FPR и построим по ним ROC-кривую. Выбирая различные наборы маркеров (то есть меняя координаты элементов выборки в признаковом пространстве) будем получать различные кривые. Оптимальному случаю соответствует кривая с наибольшей площадью под графиком (AUC).

Пусть $\mathcal{A}$ — некоторое подмножество индексов маркеров, $\mathcal{A} \subseteq \mathcal{J} = \{1, \ldots, n\}$, $\hat{\mathcal{A}}$ — оптимальный набор индексов. Тогда задачу можно сформулировать как задачу максимизации:

$$\hat{\mathcal{A}} = \arg\max_{\mathcal{A} \subseteq \mathcal{I}} \mathrm{AUC}(\sigma(X_{\mathcal{A}}, \mathbf{w}_{\mathcal{A}}), \mathbf{y}), \tag{9}$$

где $X_{\mathcal{A}}$ — состоит из столбцов $\mathbf{x}_j$, $j \in \mathcal{A}$, $\mathbf{w}_{\mathcal{A}}$ — вектор параметров, рассчитанный по формуле (6). При этом разбиение выборки на обучающую и контрольную не производится, т.к. при постановке задачи экспертами наложено ограничение на сложность модели — число признаков, входящих в модель, не должно превышать четырех.
Наборы признаков, полученные с помощью алгоритма (9), будем называть оптимальным для данной пары классов, а сами признаки — *наиболее информативным.*

Подготовка данных. Особенность используемых данных состоит в наличии большого числа пропущенных значений признаков. Прежде чем приступить к решению задачи классификации, предложим некоторые пути решения этой проблемы.

1) Заполнять пустующие позиции средним в данной группе значением признака.
2) Воспользовавшись предположением о вероятностном распределении случайной величины x_{ij}, реализациями которой являются значения признаков $\boldsymbol{\chi}_j$, заполнять пропуски соответствующими этому распределению случайными величинами в интервале от минимального значения признака в выборке до максимального. В данной работе используется предположение о нормальности случайных величин. Обозначим $\mathcal{I} = \{1, \ldots, m\}$ — множество индексов объектов, и фиксируем $\mathcal{N}(\mu, \sigma^2)$ — некоторую реализацию нормально распределенной с математическим ожиданием μ и дисперсией σ^2 случайной величины. Будем заполнять пропущенные величины по следующему правилу:

$$\begin{cases} \min\limits_{i \in \mathcal{I}} x_{ij}, & \min\limits_{i \in \mathcal{I}} x_{ij} > \mathcal{N}(\mu, \sigma^2); \\ \max\limits_{i \in \mathcal{I}} x_{ij}, & \mathcal{N}(\mu, \sigma^2) > \max\limits_{i \in \mathcal{I}} x_{ij}; \\ \mathcal{N}(\mu, \sigma^2), & \min\limits_{i \in \mathcal{I}} x_{ij} \leqslant \mathcal{N}(\mu, \sigma^2) \leqslant \max\limits_{i \in \mathcal{I}} x_{ij}. \end{cases}$$

3) **Multiple imputation for missing values**. Как и в предыдущем пункте, сделаем предположение о распределении пропущенных величин, но не будем сразу заполнять пропуски реализациями этой случайной величины. ведем обозначение для $k \in \mathcal{B}* \subset \mathcal{I} = \{1, \ldots, m\}$ индексов таких, что x_{kj} — пропущенное значение. В процедурах (6), (9) оценку $\mathbf{w}$ и AUC будем проводить по K реализациям переменных $x_{kj}, k \in \mathcal{B}*$. После этого используем медиану K полученных оценок $\mathbf{w}$, AUC.

Таблица 1. Всевозможные значения параметра α

α_1	α_2	$\ldots$	α_n
1	0	$\ldots$	0
0	1	$\ldots$	0
$\ldots$	$\ldots$	$\ldots$	$\ldots$
1	1	1	1

.

Таблица 2. Возможный результат классификаций с использованием элемента $\mathbf{x}_{m+1}$

	A1	A3	B1	B2
A1	-	0	0	1
A3	1	-	1	1
B1	1	0	-	0
B2	0	0	1	-

.

Предлагается использовать последний подход. Второй способ несколько проще реализовать, однако его использование приводит к неустойчивости результата — определяемый алгоритмом набор признаков существенно зависит от текущей реализаций случайной величины, используемой для заполнения пробелов.

Отбор признаков. Отбор признаков $\mathbf{x}_j, j \in \mathcal{A}$ осуществляется путем полного перебора. Такой подход возможен благодаря сравнительно небольшому количеству признаков. Полный перебор приводит к выбору набора признаков, лучше всего отвечающего заданному критерию. Запишем выражение для функции регрессии в виде

$$f(\mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{x}_i^T \mathbf{w}), \mathbf{x}_i^T \mathbf{w} = \alpha_1 x_{i1} \mathrm{w}_1 + \alpha_2 x_{i2} \mathrm{w}_2 + \cdots + \alpha_n x_{in} \mathrm{w}_n,$$

здесь $\alpha_j \in \{0, 1\}$ — структурный параметр. Таким образом, перебор признаков сводится к перебору значений элементов α_j вектора структурных параметров, см. (1).

Выполнение прогноза при многоклассовой классификации. По состоянию здоровья пациентов можно разделить на 5 групп.

1) $\mathbf{A}_1$. Группа пациентов, уже перенесших инфаркт.
2) $\mathbf{A}_2$. К этой группе относят пациентов, анализы которых были получены прямо в во время инфаркта. Это очень редкая группа, поэтому в данной работе она не рассматривается.
3) $\mathbf{A}_3$. Пациенты, имеющие предрасположенность к инфаркту.
4) $\mathbf{B}_1, \mathbf{B}_2$. Здоровые пациенты двух типов.

При появлении в выборке нового объекта $\mathbf{x}_{m+1}$ выполняем следующую процедуру. Решаем задачу классификации для всех пар групп. При этом для каждой пары используется оптимальный для нее набор признаков. Решив задачу логистической регрессии (7), в каждом случае получим вероятность p_{m+1} принадлежности объекта к одному из двух рассматриваемых классов. По этим результатам составим таблицу (**??**):

Здесь каждая строка есть результат сравнения некоторого класса с каждым из остальных. Например, в третьей строке содержится следующая информация: объект $\mathbf{x}_{m+1}$ более похож на объект класса B_1, чем на A_1 (в соответствующей ячейке стоит единица), но менее похож на B_1, чем на A_3 и B_2 (в ячейках нули). Присвоив классам A_1, A_3, B_1, B_2 номера 1, 2, 3, 4 соответственно, переформулируем последнее утверждение:

$$a_{23}(\mathbf{x}_{m+1}) = 0,$$

Таблица 3. Результаты отбора признаков

classes	obj. in both classes	in 1st class	best sets	AUC	Err_1	Err_2
A1 A3	31	14	[2, 11, 19, 20]	0.953	0.262	0
			[2, 13, 19, 20]	0.953		
			[2, 16, 19, 20]	0.962		
			[2, 14, 19, 20]	0.966		
			[2, 17, 19, 20]	0.970		
A1 B1	55	14	[3, 13, 18, 19]	0.829	0.254	0
			[12, 13, 15, 19]	0.829		
			[8, 18, 19, 20]	0.831		
			[13, 15, 18, 19]	0.831		
			[12, 13, 18, 19]	0.850		
A1 B1	55	14	[5, 15, 17, 19]	0.901	0.207	0
			[6, 12, 15, 19]	0.901		
			[3, 12, 15, 19]	0.903		
			[9, 12, 15, 19]	0.903		
			[12, 15, 17, 19]	0.909		
A3 B1	58	17	[5, 6, 11, 17]	0.814	0.293	0
			[2, 7, 9, 13]	0.829		
			[7, 13, 18, 20]	0.834		
			[2, 3, 5, 9]	0.835		
			[2, 5, 6, 9]	0.836		
A3 B2	43	17	[2, 3, 5, 9]	0.954	0.239	0
			[2, 3, 9, 19]	0.957		
			[2, 9, 18, 19]	0.959		
			[2, 3, 9, 17]	0.963		
			[2, 3, 9, 13]	0.970		
B1 B2	67	41	[1, 2, 3, 9]	0.821	0.563	0
			[2, 3, 9, 11]	0.823		
			[1, 2, 18, 19]	0.824		
			[2, 13, 18, 19]	0.827		
			[2, 3, 9, 18]	0.829		

где $a_{lk}(x) = \xi \in \{0, 1\}, l, k = 1, \ldots, 4$ — результат работы алгоритма (7) при выборе между классами l и k. Таким образом, мы относим объект к тому классу, для которого сумма элементов таблицы по строке наибольшая:

$$\text{class}(\mathbf{x}_{m+1}) = \arg\max_{l=1,\ldots,4} \sum_{k=1}^{4} a_{lk}(\mathbf{x}_{m+1}), \text{class}(\mathbf{x}_{m+1}) \in \{1, \ldots, 4\}.$$

Если эта сумма для двух классов совпала, результатом будет решение задачи классификации, полученное для этих двух классов.

Оценка объема выборки

В качестве обоснования достоверности классификации приводится исследование необходимого размера выборки пациентов. Рассмотрим три способа получения этой оценки.

Метод доверительных интервалов. Пусть имеется выборка независимых одинаково распределенных случайных величин $\{x_i\}\, i = 1, \ldots, m$. Подсчитанное по этой выборке

среднее арифметическое $\bar{x}$ в общем случае не совпадает с матожиданием μ рассматриваемой случайной величины. Пусть $E = \bar{x} - \mu$ — разница между максимальным измеренным средним арифметическим $\bar{x}$ и μ. При известном среднеквадратичном отклонении σ случайная величина

$$Z = \frac{\bar{x} - \mu}{\sigma\sqrt{m}} = \frac{E}{\sigma\sqrt{m}}$$

принадлежит стандартному нормальному распределению $Z \sim \mathcal{N}(0, 1)$. Тогда

$$E = z_{\alpha/2}\frac{\sigma}{\sqrt{m}},$$

где $z_{\alpha/2}$ таково, что вероятность события $\{|Z| \leqslant z_{\alpha/2}\}$ равна α. Отсюда получаем формулу для оценки размера выборки

$$m = \left(\frac{z_{\alpha/2}\sigma}{E}\right)^2. \tag{10}$$

Если $m \geqslant 30$, можно пользоваться этой формулой, заменив в ней среднеквадратичное отклонение σ на его оценку $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$. Однако в случае $m \leqslant 30$ для использования этой формулы необходимо, чтобы случайные величины x_i были распределены нормально; кроме того, среднеквадратичное отклонение σ должно быть известно.

Скользящий контроль. Выделим из исходной выборки две непересекающиеся подвыборки одинакового размера и назовем одну из них обучающей X^L, а другую — контрольной X^C. Настроим алгоритм на обучающей выборке: найдем параметры регрессии $\mathbf{w}^L$ с помощью процедуры (6) , выберем набор признаков $\mathcal{A}^L$, используя алгоритм (9).

Затем, используя полученные результаты, решим задачу классификации (7) на обучающей X^L и контрольной выборках X^C. Для каждой из выборок получим ошибки $E^L(\mathbf{w}^L)$ и на контроле $E^C(\mathbf{w}^L)$ и вычислим их отношение $S = \frac{E^L(\mathbf{w}^L)}{E^C(\mathbf{w}^L)}$. Будем добавлять в каждую выборку по элементу и проводить всю процедуру заново. Таким образом, сможем построить график зависимости величины S от объема выборки и, наблюдая за ростом этой величины, определим момент, когда она начинает меняться достаточно плавно. С этого момента считаем, что объем выборки достаточен.

Таблица 4. Число вхождений признаков в K лучших для каждой пары классов.

classes/markers	K	L	K/M	K/N	K/O	L/O	K/P	L/P	K/R
A1 A3	0	5	0	0	0	0	0	0	1
A1 B1	0	0	1	0	0	0	1	0	0
A1 B2	0	0	1	1	1	0	0	1	0
A3 B1	0	3	1	3	2	2	0	3	1
A3 B2	0	5	4	1	0	0	0	5	0
B1 B2	2	5	3	0	0	0	0	3	1

classes/markers	L/R	L/R/SA	L/T/SA	L/T/SO	U/V	U/W	U/X	U/Y	U/Z
A1 A3	0	1	1	0	1	1	0	5	5
A1 B1	2	4	0	2	0	0	4	5	1
A1 B2	4	0	0	5	0	2	0	5	0
A3 B1	0	2	0	0	0	1	1	0	1
A3 B2	0	1	0	0	0	1	1	2	0
B1 B2	0	1	0	0	0	0	3	2	0

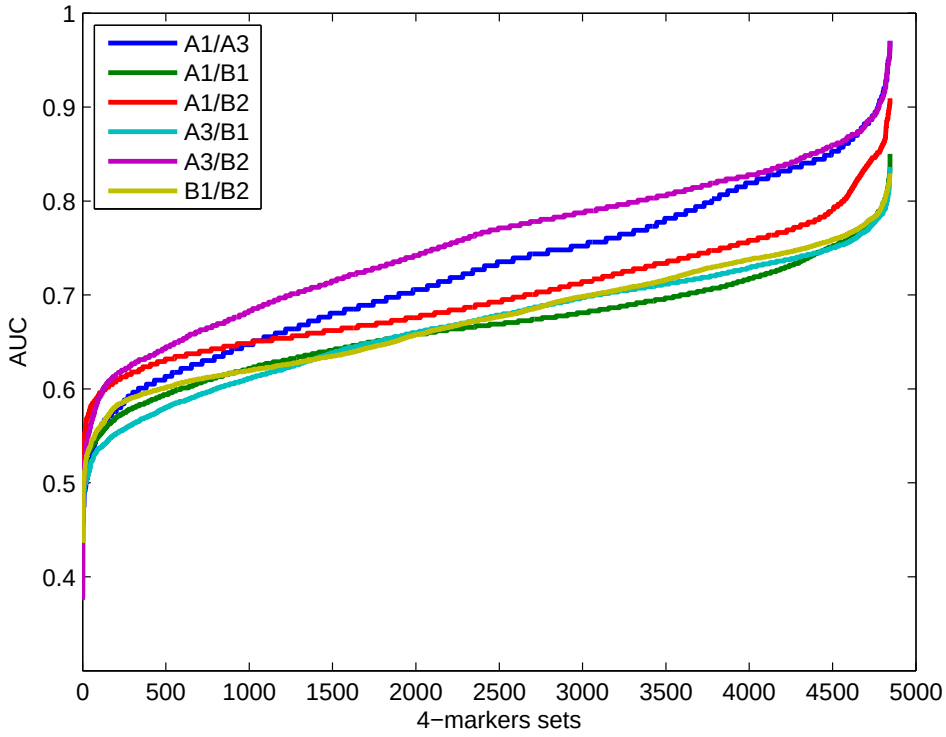


Рис. 1. Для всевозможных наборов из четырех маркеров вычислено значение AUC (для каждой пары класов). На графике отложены эти значения в порядке возрастания. Обратим внимание на то, что значение по оси абсцисс — не более чем порядковый номер значения AUC, не привязанный ни к какому определенному набору маркеров.

Расстояние Кульбака-Лейблера. В задаче восстановления регрессии предполагается, что

$$\mathbf{y} = \mathbf{f}(X, \mathbf{w}),$$

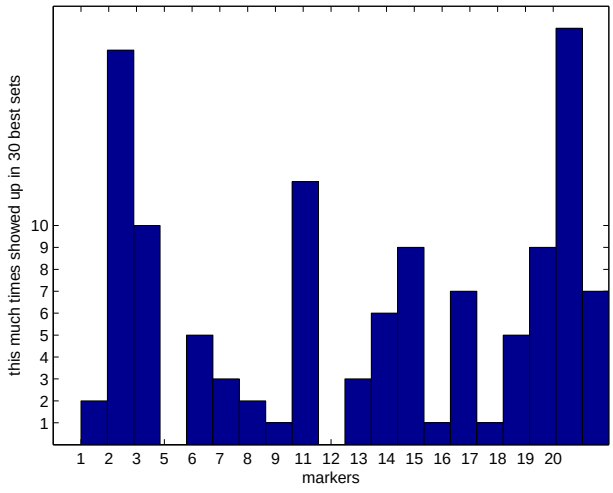


Рис. 2. Количество вхождений каждого из двадцати маркеров в набор «K лучших», $\mathcal{S}$. Например, маркер под номером 2 (L) имеет наибольшую частоту вхождений, следующий за ним — номер 19 (U/Y).

где $\mathbf{f}$ — функция с вектором параметров $\mathbf{w}$, в данной работе это логистическая функция. Покажем, что при биномиальном распределении вектора ответов $\mathbf{y}$ вектор параметров $\mathbf{w}$ распределен нормально. Пусть компоненты $\mathbf{y}$ независимы и $y_i \sim \mathcal{B}(p_i, 1-p_i)$. Рассмотрим некоторую подвыборку исходной выборки. Пусть нам удалось оценить плотность распределения признака на этой подвыборке, назовем ее $p_1(\mathbf{x})$. Удалив из подвыборки один элемент, снова произведем оценку плотности; обозначим полученную функцию $p_2(\mathbf{x})$. Тогда степень «похожести» этих функций будем определять через расстояние Кульбака-Лейблера

$$D(p_1, p_2) = \int_{-\infty}^{+\infty} p_1(x) \ln \frac{p_1(x)}{p_2(x)} dx. \tag{11}$$

Видно, что «расстояние» несимметрично, т.е. $D(p_1, p_2) \neq D(p_2, p_1)$. Если объем выборки достаточен, распределение не должно существенно меняться при малом изменении выборки. Обратное свидетельствует о слишком маленьком объеме выборки. Воспользуемся формулой Байеса для оценки апостериорных вероятностей. Пусть заданы обратные ковариационные матрицы A, B. Тогда при гипотезах о биномиальном распределении вектора ответов $\mathbf{y} \sim \mathcal{B}$ и нормальном распределении вектора параметров $\mathbf{w} \sim \mathcal{N}$, получаем

$$P(\mathbf{w}|D, A, B) = \frac{P(D|\mathbf{w}, B)P(\mathbf{w}|A)}{\int \frac{1}{2}(\mathbf{w}-\mathbf{w}_0)A^T(\mathbf{w}-\mathbf{w}_0)^T}, \tag{12}$$

где D — исследуемые данные, а в $\mathbf{w}_0$ достигается минимум функции ошибки

$$S(\mathbf{w}) = \frac{1}{2}(\mathbf{w}-\mathbf{w}_0)A^T(\mathbf{w}-\mathbf{w}_0)^T + \sum_{i=1}^{m} y_i \ln p_i + (1-y_i)\ln(1-p_i)$$

Заменяя в (11) $p_1(\mathbf{x})$ и $p_2(\mathbf{x})$ на $P(\mathbf{w}|D_1, A, B)$ и $P(\mathbf{w}|D_2, A, B)$ и, учитывая дискретность, распределений получим

$$D(P_1, P_2) = \sum P(\mathbf{w}|D_1, A, B) \ln \frac{P(\mathbf{w}|D_1, A, B)}{P(\mathbf{w}|D_2, A, B)}. \tag{13}$$

Вычислительный эксперимент: классификация и выбор признаков

В этом разделе проводился эксперимент на реальных данных, описанных в разделе «Задача классификации». Пропуски данных были заполнены средними по признакам значениями.

В таблице (3) содержатся следующие результаты: для каждой пары классов указаны K наборов маркеров, давших наибольшие значения максимизируемого критерия AUC и сами значения этого критерия, а также ошибки первого Err_1 (*false negative rate*) и второго Err_2 (*false positive rate*) рода. Здесь число K определяется визуально, по графику 1. На этом графике изображены зависимости значения AUC (в порядке возрастания), полученные при классификации на различных наборах из четырех признаков. Предлагается выбирать такое число K, при котором рост графика меняется еще достаточно сильно. В данном случае, было выбрано значение $K = 5$.

Таким образом, для j-той пары классов найдено некоторое множество оптимальных наборов $\mathcal{S}_j = \bigcup_{i=1}^{K} \{\mathcal{A}_{j_i}\}, j = 1, \ldots, 4$. Объединив признаки из всех наборов из колонки «best

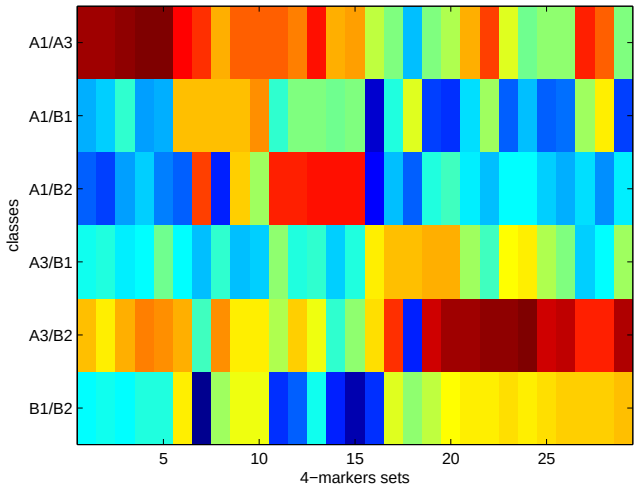


Рис. 3. Для каждого из 30 полученных наборов наиболее информативных признаков приведено его значение AUC (для каждой пары классов).

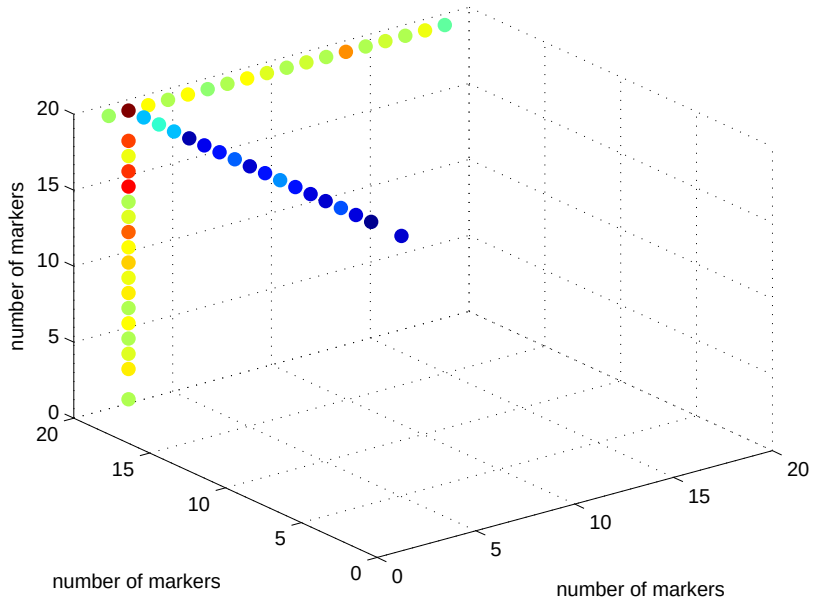


Рис. 4. Оптимальный набор маркеров (2, 19, 20), получаемый при ограничении на сложность модели, равном трем, в окрестности неоптимальных наборов. Чем теплее цвет, тем больше значение оптимизируемого функционала.

sets» таблицы (3), получим некоторое множество наиболее информативных признаков. Для каждого из них можно подсчитать количество его вхождений в наборы из «best sets» и затем построить гистограмму 2, показывающую, насколько часто каждый признак входит в K лучших наборов. Таким образом, гистограмма 2 характеризует степень качества каждого признака по отдельности. Чтобы сравнить между собой наборы признаков, построим таблицу 3. Здесь снова было рассмотрено множество $\mathcal{S}$ всех наборов, вошедших в

K лучших хотя бы для одной пары признаков: $\mathcal{S} = \bigcup_{j=1}^{4} \mathcal{S}_j$. Для всех пар классов проведем классификацию на каждом из этих наборов и сведем полученные значения критерия AUC в таблицу 3. Здесь более теплым тонам соответствуют большие значения AUC, более холодным — меньшие. Таким образом, можно наблюдать ступенчатую структуру таблицы.

Продемонстрируем «оптимальность» найденного набора (на примере A_1 и A_3 и при ограничении на сложность модели, равном трем) с помощью рисунка 4. Каждому набору из дискретной окрестности оптимального (то есть такому набору, у которого от оптимального отличается лишь один элемент) можно сопоставить величину максимизируемого функционала. На рисунке 4 эта величина выражается цветом точки — чем теплее цвет, тем больше значение.

Заключение

В работе проведен поиск наиболее информативных признаков для классификации пациентов на четыре группы с точки зрения наличия нарушений работы сердечно сосудистой системы. При этом задача многоклассовой классификации сводилась к двуклассовой классификации путем рассмотрения всевозможных пар групп. Для каждой из таких пар получен оптимальный набор признаков. Отбор признаков производился на основе полного перебора, преимуществом которого перед другими алгоритмами выбора признаков является его наглядность.

Литература

[1] Bishop C. M. *Pattern Recognition and Machine Learning*, Springer, 2006.

[2] Perez-Cruz F., *Kullback-Leibler Divergence Estimation of Continuous Distributions*, 2008.

[3] Azuaje F., Devaux Y. & Wagner D., *Computational biology for cardiovascular biomarker discovery*, http://bib.oxfordjournals.org/content/10/4/367.abstract

[4] Hosmer D. & Lemeshow S., *Applied logistic regression*, 2000.

[5] MacKay D. J. C., *Information Theory, Inference, and Learning Algorithms*, 2003.

[6] Friedman J., Hastie, Tibshirani R., *Additve logistic regression: a statistical way of boosting*, The Annals of Statistics, 28(2):337-407, 2000.

[7] Breton M. H., Stuart D. Russell, Michelle M. Kittleson, Kenneth L. Baughman, Heidecker J. B., Edward K. Kasper, Ilan S. Wittstein, Hunter C. Champion, Elayne. *Transcriptomic Biomarkers for Individual Risk Assessment in New-Onset Heart Failure.* Circulation, 2008.

[8] Bos S., *How to partition examples between cross-validation set and fraining set?* Labaratory for information representation RIKEN, Hirosawa 2-1, Wako-shi, Saitama, 351-01, Japan.

[9] Реброва О. Ю. *Статистический анализ медицинских данных. Применение прикладного пакета STATISTICA*, 2006.

[10] Fawcet T., *ROC Graphs: Notes and Practical Considerations for Researchers*, HP Laboratories, 2004.

Событийное моделирование и прогноз финансовых временных рядов*

А. А. Романенко
angriff07@gmail.com
Московский физико-технический институт, ФУПМ, каф. «Интеллектуальные системы»

Финансовые временные ряды обычно сильно зашумлены и зависят от других временных рядов (например, курс доллара или пошлины на таможне). Но насколько сильна эта зависимость, какие факторы учитывать при их прогнозировании, однозначно определить непросто. В работе для прогнозирования поведения целевого ряда используется разметка временных рядов. Предлагается алгоритм порождения признаков из размеченных временных рядов и генетический алгоритм отбора признаков на размеченных временных рядах.

Ключевые слова: *временные ряды, разметка временных рядов, логистическая регрессия, прогнозирование событий.*

Введение

В данной работе ставится задача прогнозирования динамики роста финансовых временных рядов. Это задача является экономически важной и сложной. Сложность обусловлена тем, что такие временные ряды обычно сильно зашумлены и зависят от других временных рядов (курс доллара, пошлины на таможне, и т.д.), но степень этой зависимости однозначно определить сложно.

Различают два основных подхода к прогнозированию цен: технический анализ [1] и фундаментальный [2]. Оба подхода сейчас активно применяются при прогнозировании цен на различные активы и имеют своих критиков и сторонников. Прогнозирование методами технического анализа основано на анализе временных рядов и индикаторов, прогнозирование методами фундаментального анализа — на анализе экономической ситуации и новостей.

В данной работе для прогнозирования динамики роста временных рядов предлагается использовать методы событийного моделирования к временным рядам с выделенными на них трендами [3]. Для выделения трендов к временным рядам применим технологию разметки [4, 5, 6]. Для прогнозирования используем нейронные сети [7, 8], а для поиска наилучшего набора признаков — генетический алгоритм.

В следующем разделе будет поставлена задача прогнозирования. Затем будет описан способ ее решения, а также вычислительный эксперимент на реальных данных и представлены его результаты.

Постановка и предлагаемое решение задачи

Постановка задачи. Пусть $\mathbf{f}_i(t),\ i = 1, \ldots, a$ — данные временные ряды, t —номера отсчётов. Предполагаем, что во временных рядах нет пропущенных значений. Задача состоит в том, чтобы по известным значениям временных рядов

$$\mathbf{f}_i(t),\ i = 1, \ldots, a,\ t = 1, \ldots, T$$

спрогнозировать для временного ряда $\mathbf{f}_1$ увеличится его значение в момент времени $T + 1$ или уменьшится.

Научный руководитель В. В. Стрижов

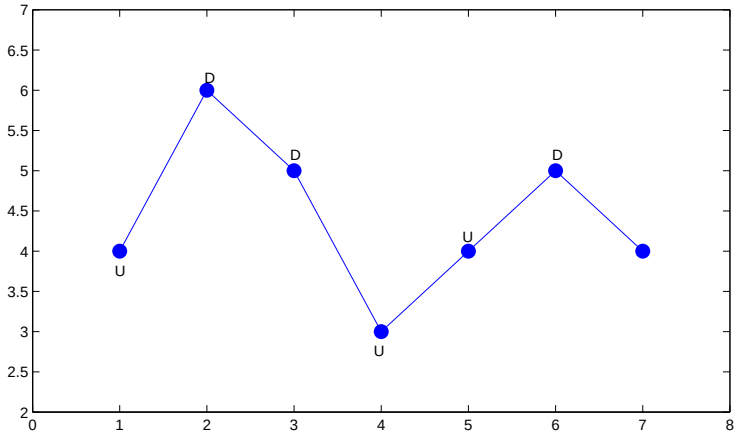


Рис. 1. Пример разметки в алфавите $\mathcal{M} = \{up, down\}$

Разметка временного ряда.

Определение 1. *Множеством меток называется конечное множество* $\mathcal{M} = \{m_1, \ldots, m_r\}$, *которое задается экспертом.*

Пример множества меток: $\mathcal{M}$= $\{up, down, plt\}$, где "*up*" — метка для обозначения точек возрастания, "*down*" — убывания, "*plt*" — метка для обозначения плато. На рис. 1 показан пример разметки временного ряда в алфавите $\mathcal{M}$= $\{up, down\}$.

Зафиксируем множество меток $\mathcal{M}$. Определим разбиение временного ряда на сегменты $\bar{\mathbf{s}} = (s_1, \ldots, s_V)$: $s_k = \{\mathbf{f}(i), \mathbf{f}(i+1), \ldots, \mathbf{f}(i+l_k)\}$, $s_{k_1} \bigcap s_{k_2} = \varnothing$ при $k_1 \neq k_2$, $\bigcup\limits_{k=1}^{V} s_k =$ $= \{\mathbf{f}(1), \ldots, \mathbf{f}(T)\}$.

Определение 2. *Разметкой временного ряда* $\mathbf{f}(t)$, $t = 1, \ldots, T$ *назовем пару* $(\bar{\mathbf{s}}, \bar{\mathbf{m}})$: $\bar{\mathbf{m}} =$ $= (m_1, \ldots, m_U)$, $m_i \in \mathcal{M}$.

Предлагается произвести разметку всех временных рядов в алфавите $\mathcal{M}$= $\{1, -1\}$. Метка "1" ставиться участку временного ряда, на котором его значения растут; метка "$-$ -1" ставится, если значения не увеличиваются. Вообще говоря, длина сегмента разметки как внутри одного ряда, так и в разных рядах может менятся. Но мы будем считать, что разметка *синхронная*, то есть длины сегментов и их начала для всех временных рядов совпадают. В таком случае можно рассматривать не изначально данные временные ряды, а последовательности из 1 и -1. Тогда задача прогнозирования сводится к прогнозированию появления в первой последовательности 1 или -1.

Порождение признаков и прогноз временного ряда. После процедуры разметки получим a последовательностей $\mathbf{f}_i$ одинаковой длины T из 1 и -1.

Для порождения признаков используем идеи из [9, 10]. Выберем натуральное число b, назовем его *глубиной логирования*. Каждому $\mathbf{f}_1(k+1)$ поставим в соответствие матрицу размера $a \times b$:

$$\begin{pmatrix} \mathbf{f}_1(k-b+1) & \ldots & \mathbf{f}_1(k-1) & \mathbf{f}_1(k) \\ \mathbf{f}_2(k-b+1) & \ldots & \mathbf{f}_2(k-1) & \mathbf{f}_2(k) \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{f}_{a-1}(k-b+1) & \ldots & \mathbf{f}_{a-1}(k-1) & \mathbf{f}_{a-1}(k) \\ \mathbf{f}_a(k-b+1) & \ldots & \mathbf{f}_a(k-1) & \mathbf{f}_a(k) \end{pmatrix}$$

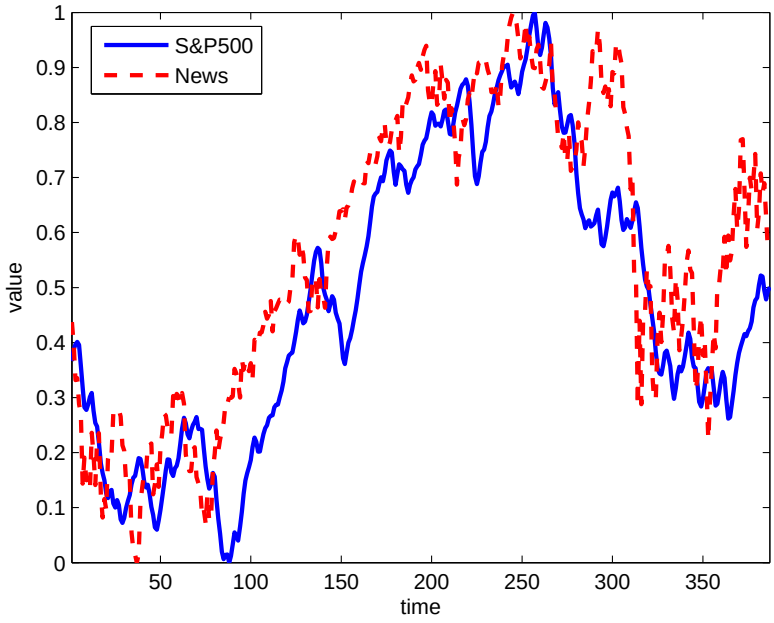


Рис. 2. Используемые временные ряды

Теперь векторизуем ее и получим вектор $\boldsymbol{x}_k$:

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{f}_1(k-b+1) \\ \dots \\ \mathbf{f}_1(k) \\ \vdots \\ \mathbf{f}_a(k-b+1) \\ \dots \\ \mathbf{f}_a(k) \end{pmatrix}. \tag{1}$$

В итоге получим множество прецедентов

$$({\mathbf{x}_k}^T, y_k), \text{ где } y_k = \mathbf{f}_1(k+1), \ k = b, \dots, T-1,$$

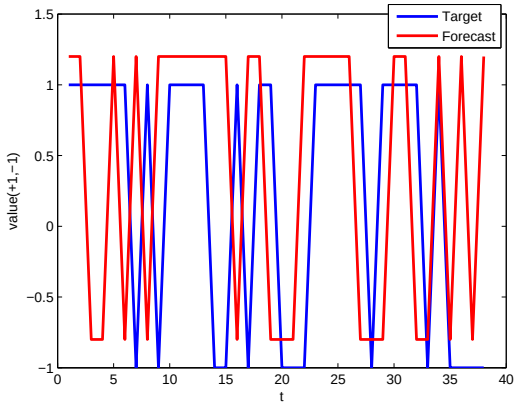
$\mathbf{x}_k$ — признаковое описание k-го объекта, $y_k \in \{+1, -1\}$ — класс, к которому он относится. Остается выбрать модель алгоритма и метод обучения, чтобы решить задачу классификации на два класса. В работе в качестве алгоритма кластеризации используются нейронные сети. Для улучшения качества классификации можно воспользоваться методами отбора признаков. Предлагается использовать генетический алгоритм отбора признаков.

Инстересно исследовать зависимость результата прогнозирования от глубины логирования b. Это исследование проведено на практике, и его результаты представлены ниже.

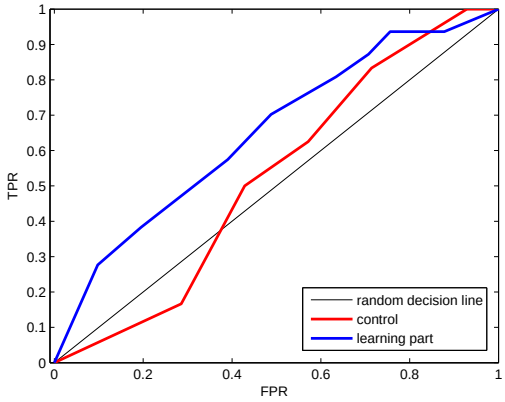
Генетический алгоритм отбора признаков. Пусть α и β — бинарные строки длины $a \times b$:

$$\alpha = (\alpha_1, \dots, \alpha_{ab}),$$
$$\beta = (\beta_1, \dots, \beta_{ab}).$$

Тогда $\alpha_i = 1$ означает, что при прогнозе учитывается i-ый признак; $\alpha_i = 0$ — не учитывается.

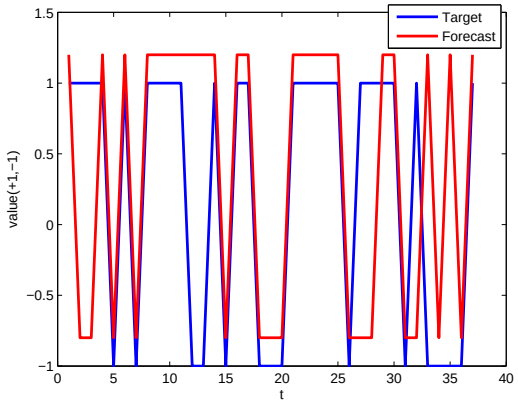


(a) Прогноз

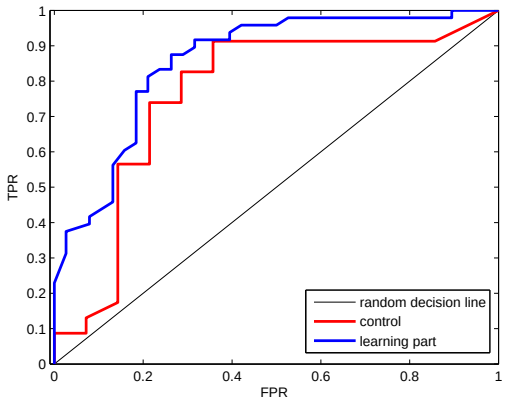


(b) ROC-кривая на обучении и контроле

Рис. 3. Результаты при $b = 3$: $Error = 44\%$, $AUC_1 = 0{,}64$, $AUC_2 = 0{,}52$

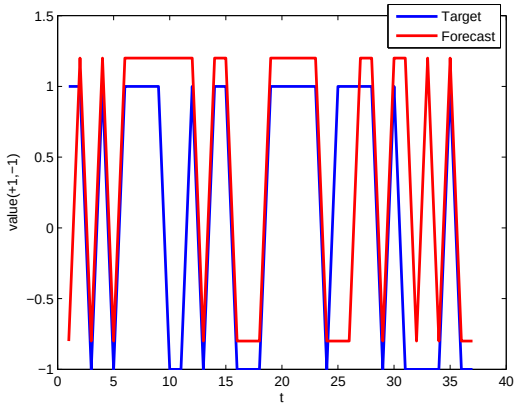


(a) Прогноз

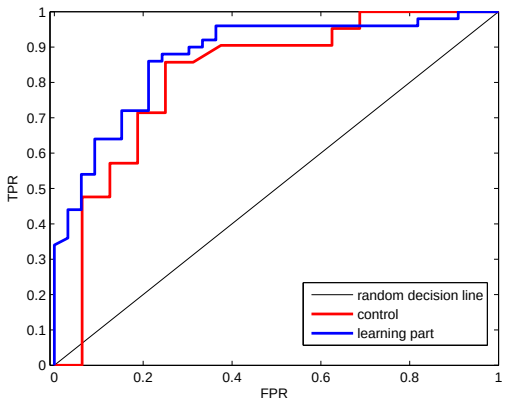


(b) ROC-кривая на обучении и контроле

Рис. 4. Результаты при $b = 6$: $Error = 24\%$, $AUC_1 = 0{,}86$, $AUC_2 = 0{,}76$



(a) Прогноз



(b) ROC-кривая на обучении и контроле

Рис. 5. Результаты при $b = 9$: $Error = 18\%$, $AUC_1 = 0{,}87$, $AUC_2 = 0{,}81$

Для поиска оптимального набора признаков будем порождать α и β с помощью следующего генетического алгоритма. Ему на вход подаются следующие параметры: I_{max} — ограничение на количество итераций, $Error_{max}$ — допустимый процент ошибок на обучающей выборке, n — число признаков, q — число мутаций при порождении новой пары признаков. Алгоритм останавливается, когда на полученном наборе признаков достигнут порог ошибки или будет сделано изначально заданное количество итераций I_{max}.

ПРОЦЕДУРА `Поиск оптимального набора признаков`

Вход: $I_{max}, Error_{max}, n, q$ —ограничение на количество итераций, требуемая точность, число признаков,число мутаций

сгенерировать случайным образом бинарные строки α и β длины n :

$\alpha = rand(1, n)$;

$\beta = rand(1, n)$;

инициализация счетчика

$i = 1$;

$BEST\_Error = 1$;

для $i = 1, \ldots, I_{max}$

обучить нейронную сеть по признакам i, где $\alpha_i = 1$:

$Error = train(X_\alpha, y)$;

если $Error < BEST\_Error$ **то**

$BEST\_SET = \alpha$;

$BEST\_Error = Error$;

обучить нейронную сеть по признакам i, где $\beta_i = 1$:

$Error = train(X_\beta, y)$;

если $Error < BEST\_Error$ **то**

$BEST\_SET = \beta$;

$BEST\_Error = Error$;

если $BEST\_Error < Error_{max}$ **то**

выход

СКРЕЩИВАНИЕ

запомнить старые α и β и выбрать случайное целое $k \in \{1, \ldots, n-1\}$:

$k = random(n)$;

$\alpha_{old} = \alpha$;

$\beta_{old} = \beta$;

$\alpha = [\alpha_{old}(1:k), \beta_{old}(k+1:end)]$;

$\beta = [\beta_{old}(1:k), \alpha_{old}(k+1:end)]$;

МУТАЦИЯ

изменить значения в α и β на q произвольных позициях:

для j=1,. . . ,q

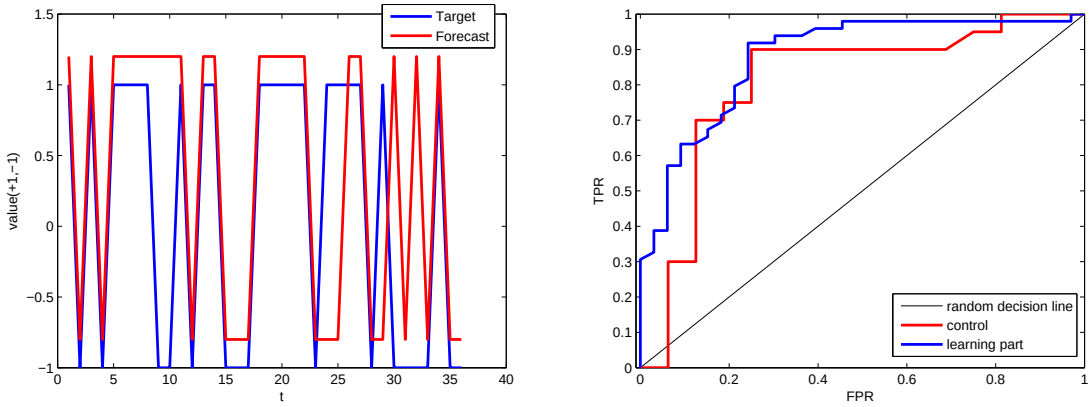
$k = random(n)$;

$\alpha_k = |\alpha - 1|$;

$k = random(n)$;

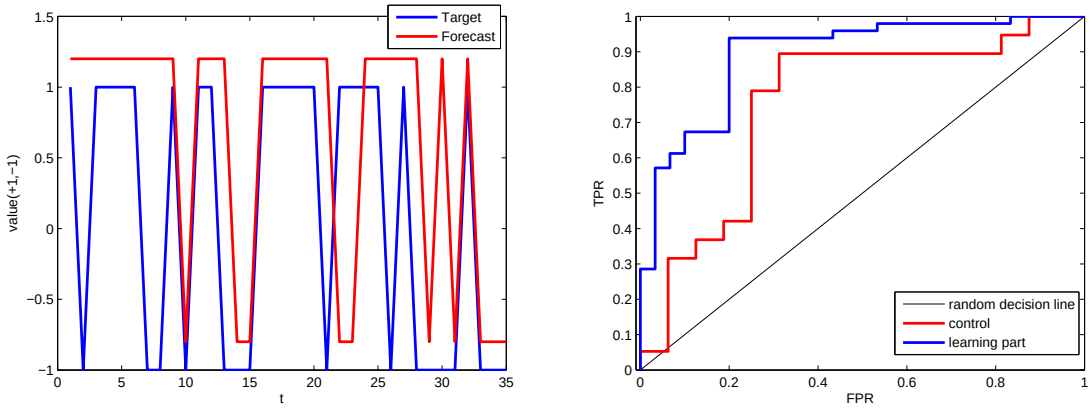
$\beta_k = |\beta - 1|$;

вернуть $BEST\_SET$



(a) Прогноз (b) ROC-кривая на обучении и контроле

Рис. 6. Результаты при $b = 11$: $Error = 19\%$, $AUC_1 = 0{,}88$, $AUC_2 = 0{,}80$



(a) Прогноз (b) ROC-кривая на обучении и контроле

Рис. 7. Результаты при $b = 15$: $Error = 28\%, AUC_1 = 0{,}89$, $AUC_2 = 0{,}75$

Вычислительный эксперимент

Описание временных рядов. Спрогнозируем динамику роста индекса S&P-500 по его истории и по временному ряду, характеризующему поток новостей по заданной тематике. Количество временных отсчетов $T = 394$. Так как в работе нас не интересуют абсолютные значения временного ряда, то можно нормировать временные ряды. Для временного ряда $\mathbf{f}(t)$ нормированный ряд будет вычисляться так:

$$\hat{\mathbf{f}}(t) = \frac{\mathbf{f}(t) - \mathbf{f}_{min}}{\mathbf{f}_{max} - \mathbf{f}_{min}}, \ t = 1, \ldots, T,$$

где $\mathbf{f}_{min}$ и $\mathbf{f}_{max}$ — минимальное и максимальное значения временного ряда $\mathbf{f}$. На рис. 2 показаны используемые нормированные временные ряды.

Результаты вычислительного эксперимента. Ниже представлены полученный в результате прогноза ряд (красным цветом) прогнозируемый ряд (синим цветом) для некоторых значений глубины логирования и ROC-кривые для полученного прогноза на обучающей выборке и на контроле. Для удобства прогноз на графике изображен немного

выше прогнозируемого ряда. В качестве обучающей выборки были выбраны 70% всех прецедентов, в качестве контрольной — оставшаяся часть. Для каждого значения глубины логирования приведены значения площади под ROC-кривыми на обучении AUC_1 и на контроле AUC_2.

Из графиков видно, что сначала при росте значения глубины логирования b качество прогноза улучшается (увеличивается площадь под ROC-кривой AUC_2 на контрольной выборке): рис. 3 – 4. При определенном значении глубины логирования ошибка AUC_2 достигает своего максимума (рис. 5). При дальнейшем росте b значение AUC_2 падает (рис. 6 – 7). Это означает, что происходит переобучение: несмотря на то, что площадь под ROC-кривой на обучающей выборке AUC_1 растет с ростом b, значение AUC_2 падает, а $Error$ увеличивается.

Таким образом, получили, что для данных временных рядов наилучшее значение глубины логирования $b = 9$, при этом ошибка на контрольной выборке равна $Error = 18\%$, площадь под ROC-кривой равна $AUC = 0{,}81$.

Код и данные для проведения эксперимента находятся по адресу [11].

Литература

[1] Achelis S. B. *Technical analysis from A to Z* / New York:McGraw Hill, 2001.

[2] Ritchie J. C. *Fundamental Analysis: A Back-To-The Basics Investment Guide to Selecting Quality Stocks* / Irwin Professional Pub, 1996.

[3] Кононенко Д. С. *Прогнозирование событий* // Машинное обучение и анализ данных, 2010, Т. 1, № 1, С. 113–115.

[4] Колесникова С. И. *Особенности применения эталонных моделей для разметки временного ряда при распознавании состояний сложного объекта.* //Управление, вычислительная техника и информатика, 2011, Т. 1, № 1, С. 31–36.

[5] Чехович Ю. В. *Элементы алгебраической теории синтеза обучаемых алгоритмов выделения трендов.* // Диссертация на соискание степени магистра, ФУПМ МФТИ(ГУ), 2003.

[6] Чехович Ю. В. *Об обучаемых алгоритмах выделения трендов.* // Искусственный интеллект, 2002.

[7] Воронцов К. В. *Лекции по линейным алгоритмам классификации*, www.machinelearning.ru/, 2011

[8] Головко В. А. *Нейронные сети: обучение, организация и применение* /ИПРЖР, 2001.

[9] Филипенков Н. В. *О задачах анализа пучков временных рядов с изменяющимися закономерностями.* // Диссертация на соискание степени магистра, ВМК МГУ, 2006.

[10] Филипенков Н. В. *Об алгоритмах прогнозирования процессов с плавно меняющимися закономерностями.* // Диссертация на соискание степени кандидата наук, ВЦ РАН им. А. А. Дородницына, 2010.

[11] Исходные временные ряды, http://bit.ly/uCf4XV, 2011.

Обзор некоторых статистических моделей естественных языков*

Е. А. Будников
unicorn1992@bk.ru
Московский физико-технический институт, ФУПМ, каф. «Интеллектуальные системы»

В работе производится обзор и сравнение следующих моделей натурального языка: n-граммы, n-граммы на классах, дисконтная модель. В первой части работы будет проведён обзор основной литературы по данной тематике, во второй части будут введены основные понятия и описаны сами методы.

Ключевые слова: языковая модель, униграмма, биграмма, дисконтная модель.

Введение

В задачах, связанных с распознаванием языков, мы часто сталкиваемся с проблемой распознавыния строки слов, прошедших через зашумлённый канал. Чтобы эффективней решать эту проблему, необходимо уметь оценивать априорную вероятность появления тех или иных слов. Метод n-грамм описывается в [1, 2, 3, 4]. Метод n-грамм на классах (*Class-Based n-gram Models*) подробно описан в [5, 1]. Про дисконтную модель (*discounting*) можно почитать в [3, 1, 4].

Пусть $W = w_1 w_2 \dots w_k$ — строка, которую подают на вход зашумлённого канала. Роль такого канала могут исполнять радиоэфир или человек, который переводит строку на другой язык. На выходе получим сигнал Y. По этому выходу необходимо восстановить исходную строку. Вообще говоря, многие строки W могут на выходе приводить к сигналу Y, но чтобы минимизировать вероятность ошибки, необходимо взять такую строку $\hat{\mathrm{W}}$, апостеорная вероятность которой $\Pr(W|Y)$ максимальна. При фиксированном выходе Y эта задача эквивалентна максимизации совместной плотности строки W и выхода Y $\Pr(W, Y)$. Но при этом по формуле Байеса получим:

$$\Pr(W, Y) = \Pr(Y|W) \cdot \Pr(W). \tag{1}$$

Получили разбиение большой задачи на две подзадачи. Мы будем заниматься нахождением второго множителя. Будем обозначать $w_i^j = w_i w_{i+1} \dots w_j$. При таких обозначениях $W \equiv w_1^k$.

$$\Pr(w_1^k) = \Pr(w_k|w_1^{k-1}) \cdot \Pr(w_{k-1}|w_1^{k-2}) \cdot \ \dots \ \cdot \Pr(w_2|w_1) \cdot \Pr(w_1) \tag{2}$$

Определение 1. *Моделью естественного языка назовём функцию*

$$f : R^P \times R^N \to R^k,$$

где R^P — пространство параметров (оценок вероятностей), R^N — пространство акустических входов, R^k — пространство прогнозов

Качество модели будем оценивать по значению *перплексии* [2].

Научные руководители: В. Я. Чучупал, В. В. Стрижов

Определение 2. *Перплексией назовём следующую величину:*

$$PP = \Pr(w_1 w_2 \dots w_k)^{\frac{1}{k}}.$$

Чем меньше перплексия, тем лучше модель.

Модель n-грамм

Одной из основных проблем, возникающих при решении задачи, является огромное количество параметров, поэтому методы во многом направлены на то, чтобы уменьшить число параметров.

В методе n-грам мы считаем две предыстории одинаковыми, если они оканчиваются на одинаковые $n-1$ слов. Другими словами,

Определение 3. *Модель естественного языка называется моделью на n-граммах, если для параметров модели выполнено условие:*

$$\Pr(w_k|w_1^{k-1}) = \Pr(w_k|w_{k-n+1}^{k-1}). \tag{3}$$

Если словарь содержит V слов, то 1-граммы (или *униграммы*) порождают модель, имеющую $V-1$ независимых параметров: V параметров $\Pr(w_i)$ связаны равенством

$$\sum_{i=1}^{V} \Pr(\tilde{w}_i) = 1, \tag{4}$$

где $\tilde{w}_i$ — слова из словаря. 2-граммы (или *биграммы*) порождают V^2-1 независимых параметров: $V(V-1)$, имеющих форму $\Pr(w_2|w_1)$, и $V-1$, имеющих форму $\Pr(w)$. По индукции легко показать, что модель n-грамм содержит V^n-1 параметров. Действительно, $V^{n-1}(V-1)$ параметров, имеющих форму $\Pr(w_n|w_1^{n-1})$, и $V^{n-1}-1$ параметров, имеющих форму $\Pr(w_{n-1}|w_1^{n-2})$(по предположению индукции). Всего V^n-1.

Настраивать параметры модели будем по тексту T, который называется *обучающим текстом*, в процессе, который называется *обучением*. Пусть $C(\mathbf{w})$ — число, означающее, сколько раз строка $\mathbf{w}$ встретилась в обучающем тексте. Тогда в случае *униграмм* максимум правдоподобия для параметра $\Pr(w)$ достигается при $\Pr(w) = \frac{C(w)}{T}$. Для случая n-грамм имеет место быть такой результат:

$$\Pr(w_n|w_1^{n-1}) = \frac{C(w_1^{n-1}w_n)}{\sum_w C(w_1^{n-1}w)}. \tag{5}$$

Модель n-грамм на классах

Чем больше значение n, тем точнее модель. Но в условиях ограниченной обучающей выборки текстов с ростом n доверие к полученной модели должно падать. Необходимо умешьшать число параметров, стараясь не терять значительно точности. Например, можно оценивать вероятность появления не отдельного слова, а некоторой группы слов. Один из способов сделать — использовать n-граммы на классах.

Совершенно ясно, что некоторые слова могут иметь похожие распределения вероятностей. Например, понятно, что слова «Пятница» и «Среда» имеют похожие распределения. Но не одинаковые. Вряд ли мы услышим где-то в офисе радостное восклицание «Слава Богу, наконец-то среда!» или станем беспокоиться о среде, выпавшей на тринадцатое число. Но тем не менее, объединение слов в классы представляется очень удачной идеей.

Пусть существует некоторая функция $\pi : \Omega \to G$, где Ω — множество слов, словарь, а G — множество классов слов. Тогда обозначим $Pr(w|g)$ встречаемость слова w в классе g, а $\Pr(g_n|g_1^{n-1})$ — вероятность встретить слово из класса g_n после последовательности слов, имеющих форму $g_1 g_2 \dots g_{n-1}$.

Тогда для биграмм имеют место следующие соотношения:

$$\Pr(w_i|w_{i-1}) = \Pr(w_i, \pi(w_i)|w_{i-1}, \pi(w_{i-1})) = \tag{6}$$

$$= \Pr(w_i|\pi(w_i), \pi(w_{i-1}), w_{i-1}) \cdot \Pr(\pi(w_i)|\pi(w_{i-1}), w_{i-1}).$$

Для общего случая n-грамм введём

Определение 4. *Модель n-грамм назовём моделью n-грамм на классах, если выполняется гипотеза:* $\Pr(w_k|w_1^{k-1}) = \Pr(w_k|g)\Pr(g_k|g_1^{k-1})$,*где* $k = 1, \dots, n$.

Опишем теперь один алгоритм построения функции π на примере биграмм. Пусть $T = (t_1, t_2, \dots, t_T)$ — обучающая выборка, причём все слова содержатся в словаре V. Функция правдоподобия тогда равна

$$L(T) = \Pr(T) = \prod_{x,y \in V} \Pr(x|y)^{C(x,y)}, \tag{7}$$

где x,y — слова из словаря, причём y предшествует x, а $C(x, y)$ показывает, сколько раз последовательность слов «yx» встретилась в обучающей выборке T.

Для удобства будем использовать логарифм функции правдоподобия вместо самой функции:

$$\log L(T) = \sum_{x,y \in V} C(x, y) \cdot \log \Pr(x|y). \tag{8}$$

Из данного выше определения модели n-грамм на классах заключаем, что максимум правдоподобия для биграмм достигается при

$$\Pr(w_i|w_{i-1}) = \frac{C(w_i)}{C(\pi(w_i))} \cdot \frac{C(\pi(w_i), \pi(w_{i-1}))}{C(\pi(w_{i-1}))}, \tag{9}$$

где $C(w_i)$ — число раз, которые слово w_i встретилось в обучающей выборке, а $C(\pi(w))$ — число раз, которые слова из класса $\pi(w)$ встретились в выборке, аналогично $C(\pi(w_x), \pi(w_y))$ — число пар вида «$\pi(w_y)\pi(w_x)$», встретившиеся в выборке.

Подставим теперь это выражение в функцию правдоподобия и преобразуем:

$$\log L(T) = \sum_{x,y \in V} C(x, y) \cdot \log \left(\frac{C(x)}{C(\pi(x))} \cdot \frac{C(\pi(x), \pi(y))}{C(\pi(y))} \right) \tag{10}$$

$$= \sum_{x,y \in V} C(x, y) \cdot \log \left(\frac{C(x)}{C(\pi(x))} \right) + \sum_{x,y \in V} C(x, y) \cdot \log \left(\frac{C(\pi(x), \pi(y))}{C(\pi(y))} \right)$$

$$= \sum_{x \in V} C(x) \cdot \log \left(\frac{C(x)}{C(\pi(x))} \right) + \sum_{g,h \in G} C(g, h) \cdot \log \left(\frac{C(g, h)}{C(h)} \right)$$

$$= \sum_{x \in V} C(x) \cdot \log C(x) - \sum_{x \in V} C(x) \cdot \log C(\pi(x))$$

$$+\sum_{g,h\in G} C(g,h)\cdot\log C(g,h)-\sum_{g,h\in G} C(g,h)\cdot\log C(h)$$

$$=\sum_{x\in V} C(x)\cdot\log C(x)+\sum_{g,h\in G} C(g,h)\cdot\log C(g,h)$$

$$-2\sum_{g\in G} C(g)\cdot\log C(g),$$

где (g,h) — некоторая последовательность классов «hg».

Теперь вы заметим, что первое слагаемое не зависит от выбора функции π. Поэтому его рассматривать необязательно, когда мы будем оптимизировать π. Будем максимизировать функцию

$$F_\pi=\sum_{g,h\in G} C(g,h)\cdot\log C(g,h)-2\sum_{g\in G} C(g)\cdot\log C(g). \tag{11}$$

Приведём теперь алгоритм оптимизации функции π. Перед запуском алгоритма определяется число классов.

1: **для всех** $w\in\Omega$
2: $G(w)=1$ // *инициализация*
3: **для** $i=1\ldots n$
4: **повторять**
5: **для всех** $c\in G$
6: Переместить слово w в класс c, запомнив его предыдущий класс
7: Вычислить изменения F_π для этого перемещения в c. Переместить слово w назад в его предыдущий класс
8: Переместить слово w в класс, кторый больше всего увеличивает F_π, или никуда не перемещать, если увеличения ни на каком перемещении не происходит
9: **пока** s

Дисконтная модель (*discounting*)

Рассмотрим событие S, которое встретилось s раз, а общее количество наблюдений A. Тогда оценка вероятности S по принципу наибольшего правдоподобия будет равна

$$\Pr(S)=\frac{s}{A}. \tag{12}$$

Но тогда, в соответствии с этим принципом, событиям, которые не были встречены среди обучающего текста T, будут приписаны нулевые вероятности, а значит, будучи встреченными на тесте, они никогда не будут распознаны. Чтобы справиться с этой проблемой, можно поступить следующим способом. В оценке вероятности события вместо числа s брать

$$s'=d_s\cdot s, \tag{13}$$

где d_s — множитель, зависящий от числа раз, которые событие встретилось в обучающем тексте. Тогда получим дисконтную оценку вероятности события A:

$$\Pr{}_{discount}(S)=\frac{s'}{A}=\frac{d_s\cdot s}{A}. \tag{14}$$

Различные дисконтные методы различаются стратегией выбора d_s.

Обозначим c_s число всех событий которые встретились в процессе обучения ровно s раз. Тогда общее число наблюдений $A = \sum_{s \geqslant 1} c_s \cdot s$. Получается, что таким образом мы перераспределили оценки вероятности между событиями и оставили на все не встретившиеся в обучении слова $1 - \frac{1}{A} \sum_{s \geqslant 1} d_s \cdot c_s \cdot s$. Если c_0 — число таких событий, то оценка вероятности каждого из них равна

$$\frac{1}{c_0} \left(1 - \frac{1}{A} \sum_{s \geqslant 1} d_s \cdot c_s \cdot s \right). \tag{15}$$

Дисконтная модель Гуда-Тьюринга (Good-Turing). В статье[6] предлагается следующая стратегия выбора множителя:

$$d_s = (s+1) \frac{c_{s+1}}{s \cdot c_s}. \tag{16}$$

Эта стратегия называется оценкой Гуда-Тьюринга. Несмотря на очевидную простоту стратегии, у неё есть существенный недостаток: она проваливается в случае, если $c_a = 0$ для некоторого a и существует $b > a$, такой, что $c_b \neq 0$. Решение этой проблемы было предложено в [7]. Пусть есть некое, достаточно большое число k, такое что все оценки вероятностей событий, встретившихся в процессе обучения более k раз, признаем надёжными. При этом d_s будет выглядеть так:

$$d_s = \begin{cases} \frac{(s+1)\frac{c_{s+1}}{s \cdot c_s} - (k+1)\frac{c_{k+1}}{c_1}}{1 - (k+1)\frac{c_{k+1}}{c_1}} & , \ 1 \leqslant s \leqslant k \\ 1 & , \ s > k \end{cases} \tag{17}$$

Этот метод тоже нестабильный, так как возможны ситуации, когда $d_s < 0$.

Модель абсолютного уменьшения (Absolute discounting). Одной из альтернатив модели Гуда-Тьюринга является модель абсолютного уменьшения[8]. В этой модели происходит уменьшение числа a для каждого события на фиксированное число m.

$$d_s = \frac{s - m}{s}. \tag{18}$$

Для того чтобы уменьшение суммарной вероятности было таким же, как в модели Гуда-Тьюринга, необходимо:

$$m = \frac{c_1}{\sum_{s \geqslant 1} c_s}. \tag{19}$$

Заключение

Работа не описывает весь перечень методов, использующихся при моделировании естественных языков. Представленные в работе алгоритмы описывают по сути один подход, скорее дополняют друг друга, нежели конкурируют. К достоинствам метода n-грамм стоит отнести простоту реализации и интуитивную понятность подхода. Метод n-грамм на классах развивают идею уменьшения количества параметров модели, при этом, конечно, теряя в качестве прогнозирования. Дисконтная модель исправляет существенный недостаток моделей на n-граммах — нулевые значения параметров, которые приводят к невозможности прогнозирования последовательностей, которые могут в жизни, но не встретились в обучении. Модель Гуда-Тьюринга, как уже отмечалось выше, обладает простой реализацией

и прозрачной интерпретацией, но, к сожалению, является при этом неустойчивой. Модель абсолютного уменьшения исправляет этот недостаток.

Литература

[1] Huang X., Acero A., Hon H.-W. *Spoken Language Processing, A Guide to Theory, Algorithm, and System Development* /Prentice Hall PTR, 2001.

[2] Jelinek F. *Statistical Methods for Speech Recognition.* //The MIT Press, Cambridge, Massachusetts, 1997.

[3] Gotoh Y., Renals S. *Statistical language modelling.* //In Steve Renals and Gregory Grefenstette, editors, ELSNET Summer School, volume 2705 of Lecture Notes in Computer Science, pp. 78 –105, Springer, 2000.

[4] Young S., Bloothooft G., editors *Corpus-Based Methods in Language and Speech Processing* /Kluwer Academic Publishers, Dordrecht, 1997.

[5] Brown P. F., Della Pietra V. J., deSouza P. V., Mercer R. L. *Class-based n-gram models of natural language.* //Proceedings of the IBM Natural Language ITL, pp. 283 –298, Paris, France, March 1990.

[6] Good I. J. *The population frequencies of species and the estimation of population parameters.* //Biometrika, vol. 40(3, 4):pp. 237 –264, 1953.

[7] Katz S. M. *Estimation of probabilities from sparse data for the language model component of a speech recognizer.* //IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 35(3):pp. 400 – 401, March 1987.

[8] Ney H., Essen U., Kneser R. *On structuring probabilistic dependencies in stochastic language modelling.* //Computer Speech and Language, vol. 8:pp. 1 –38, 1994.