МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное бюджетное государственное образовательное

Учреждение Высшего образования

**ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Шатров А.В.

**Прогнозирование экономических процессов. Основы анализа данных с использованием среды R**

Учебно-методическое пособие для

#### направлений 09.03.03 Прикладная информатика

#### 01.04.02 Прикладная математика и информатика

Киров, 2020 г

**УДК 004.9**

Шатров А.В. Прогнозирование экономических процессов. Основы анализа данных с использовании среды R - Киров, изд. ВятГУ, 2020.- 134 с.

Для использования методов социально-экономического прогнозирования необходимо знание математической статистики и основ эконометрики. Во введении в курс анализа данных мы используем опыт многолетнего преподавания отдельных дисциплин современной прикладной и вычислительной статистики. Анализ данных востребован в различных областях науки и прикладных исследований (биоинформатика, биофармацевтика, медицина, экономика). Природа данных при анализе не имеет большого значения, важно, чтобы предоставленные данные имели соответствующую структуру и объем. Эти и многие другие данные могут быть получены с платформы Kaggle[[1]](#footnote-1). Настоящее методическое пособие является начальным и ориентировано на использование табличного редактора Excel и среду программирования R. В первой части пособия приведены данные и ссылки общего характера, которые необходимы для работы в среде R.

Подп. В печ. Усл. Печ.л. Зак. Тир.

ПРИП ВятГУ, 610000, г. Киров, ул. Московская, 36

С ФГБОУ ВПО «Вятский государственный университет»

**Содержание**

|  |  |  |
| --- | --- | --- |
|  | Введение | 7 |
| 1. | Раздел 1. Обработка временных рядов и прогнозирование в табличном редакторе MS Excel | 8 |
| 1.1 | Понятие временного ряда. Основные характеристики временного ряда. | 8 |
| 1.2 | Методы сглаживания временного ряда | 18 |
| 1.3 | Прогнозирование временных рядов с учетом неоднородности и сезонности | 20 |
| 1.3.1 | Экспоненциальное сглаживание | 22 |
| 1.3.2 | Модель Хольта-Уинтерса | 24 |
| 1.3.3 | Адаптивная модель Брауна | 29 |
| 1.4 | Моделирование авторегрессии временных рядов | 34 |
| 1.4.1 | Модели авторегрессии | 34 |
| 1.4.2 | Критерии качества модели | 38 |
| 2. | Основы работы в программной среде R | 40 |
| 2.1 | История создания среды R | 40 |
| 2.2 | Установка и обновления среды R | 41 |
| 2.3 | Установка и обновление дополнительных пакетов | 41 |
| 2.4 | Работа с данными | 42 |
| 2.4.1 | Работа с матрицами | 42 |
| 2.4.2 | Массивы данных | 46 |
| 2.4.3 | Факторные данные | 47 |
| 2.4.4 | Структуры данных | 47 |
| 2.4.5 | Рабочая директория, сохранение и воспроизводство  кодов | 48 |
| 2.5 | Ресурсы среды R: условия, циклы, функции | 49 |
| 2.6 | Работа с индексами | 50 |
| 3 | Линейная регрессия | 51 |
| 3.1 | Парная линейная регрессия | 51 |
| 3.2 | Оценивание параметров регрессии | 52 |
| 3.3 | Множественная линейная регрессия | 54 |
| 3.4 | Оценивание параметров множественной  регрессии: t – критерий | 55 |
| 3.5 | Оценивание параметров множественной  регрессии: F – критерий | 59 |
| 3.6 | Проблема отбора переменных | 59 |
| 3.7 | Использование множественной линейной  регрессии для качественных переменных | 61 |
| 3.8 | Проблемы, возникающие при использовании  множественной линейной регрессии | 63 |
| 3.9 | Сравнение линейной регрессии с методом  *К* – ближайших соседей | 66 |
| 3.10 | Практическая реализация методов линейной регрессии | 67 |
| 3.10.1 | Парная линейная регрессия | 67 |
| 3.10.2 | Множественная линейная регрессия | 71 |
| 3.10.3 | Эффекты взаимодействия | 73 |
| 3.10.4 | Нелинейные преобразования предикторов | 74 |
| 3.10.5 | Качественные переменные | 76 |
| 4 | Методы классификации | 78 |
| 4.1 | Общее представление о классификации | 78 |
| 4.2 | Логистическая регрессия | 79 |
| 4.2.1 | Оценивание коэффициентов логической регрессии | 82 |
| 4.2.2 | Множественная логистическая регрессия | 84 |
| 4.3 | Дискриминантный анализ | 85 |
| 4.4 | Практические применения методов классификации | 87 |
| 4.4.1 | Данные по цене акций | 87 |
| 4.4.2 | Логистическая регрессия на примере биржевых операций | 91 |
| 4.4.3 | Линейный дискриминантный анализ (LDA) | 94 |
| 4.4.4 | Квадратичный дискриминантный анализ (QDA) | 96 |
| 4.4.5 | Метод *К –* ближайших соседей | 97 |
| 4.4.6 | Применение к данным по жилым прицепам | 98 |
| 5 | Методы создания повторных выборок | 100 |
| 5.1 | Перекрестная проверка | 101 |
| 5.1.1 | Метод проверочной выборки | 101 |
| 5.1.2 | Перекрестная выборка по отдельным наблюдениям | 103 |
| 5.1.3 | *k –* кратная перекрестная проверка | 104 |
| 5.2 | Бустреп | 106 |
| 5.2.1 | Понятие о бустреп-анализе | 106 |
| 5.2.2 | Реализация бустреп-анализа с помощью пакета boot() | 108 |
| 5.2.3 | Бустреп-анализ для единичной выборки | 110 |
| 5.2.4 | Бустреп-анализ для нескольких выборок | 112 |
| 5.3 | Практическое применение методов повторных выборок | 113 |
| 5.3.1 | Подход создания набора для валидации | 113 |
| 5.3.2 | Перекрестная проверка по принципу “leave-one-out” | 115 |
| 5.3.3 | Перекрестная проверка по принципу k-Fold | 116 |
| 5.3.4 | Bootstrop | 117 |
| 5.3.5 | Оценка точности модели логистической регрессии | 118 |
|  | Рекомендуемая литература | 121 |
|  | Приложения | 122 |

# Введение

Анализ данных (*data mining*) (АД) или вычислительная статистика является сравнительно новой научной областью. Важность АД обусловлена огромными базами данных, которые порождаются передовыми технологиями.

В настоящее время, вычислительная статистика является одним из наиболее динамичных и перспективных научных направлений. Успешные результаты в области АД базируются на глубоком понимании статистических закономерностей и на свободном владении языками научного программирования.

В отличие от теоретической статистики, вычислительная статистика ориентирована на обработку и анализ реальных данных. Соответственно, центральным вопросом является выбор оценивающего критерия, согласно которому производится сравнение различных методов и алгоритмов. В этом смысле соревнования по АД представляются очень перспективным направлением. С одной стороны, эти соревнования предусматривают участие в отдельном проекте десятков и сотен независимых команд. С другой стороны, оценка качества результатов является совершенно независимой.

*R* — это научная некоммерческая программная среда[[2]](#footnote-2), развиваемая в рамках международного проекта GNU которая включает язык программирования для автоматизированной обработки данных и большой набор разнообразных функций для анализа данных, включая графические иллюстрации. Среда *R* применяется не только для первичного анализа, но также для математического моделирования высокого уровня. Отметим, что среда *R* может быть также использована в сочетании с другими программными средами. На базе *R* разработано большое количество многофункциональных пакетов, которые могут быть использованы для специализированного анализа данных.

# Раздел 1. Обработка временных рядов и прогнозирование в табличном редакторе MS Excel

* 1. **Понятие временного ряда. Основные характеристики временного ряда.**

*Временными рядами* или *рядами динамики* называют статистические данные, полученные последовательно во времени в результате наблюдения за развитием изучаемого явления. При описании рядов динамики обязательно должны иметься два основных элемента:

* *Показатель времени t*;
* Соответствующий ему *уровень* развития изучаемого *явления Y*.

Для наглядности и удобства обработки данных при записи временных рядов пользуются таблицами и (или) построением графиков и диаграмм.

Пример временного ряда

Табл. 1.

|  |  |
| --- | --- |
| ***На 1 января*** | ***МРОТ, руб.*** |
| **2011** | **4330** |
| **2012** | **4611** |
| **2013** | **5205** |
| **2014** | **5554** |
| **2015** | **5965** |
| **2016** | **6204** |
| **2017** | **7500** |

Графическое представление временного ряда.

Полигон (вариационный ряд)

Рис. 1 График распределения показателя МРОТ по годам

Гистограмма

Рис. 2 Гистограмма распределения показателя МРОТ по годам

При этом способы графических представлений весьма разнообразны и в Excel выбираются из меню Вставка/Диаграммы/графики. Непосредственный выбор дизайна подбирается с помощью Конструктора меню.

Классификация временных рядов

Табл. 2.

|  |  |  |
| --- | --- | --- |
| **1. В зависимости от способа выражения** | | **Ряды абсолютных величин**, т.е. уровни ряда представлены в абсолютных величинах. |
| **Ряды относительных величин**, т.е. уровни ряда представлены в относительных величинах. |
| **Ряды средних величин**, т.е. уровни ряда представлены в виде средних показателей за определенные промежутки времени. |
| **2. В зависимости от момента определения разности моментов времени** | | **Моментные ряды,** т.е. уровни ряда фиксировались в конкретные моменты времени. Могут быть равноотстоящими и разнесенными во времени |
| **Интервальные равноотстоящие ряды,** т.е. уровни ряда представляют собой суммарные показатели за определенные и равные промежутки времени. |
| **Интервальные разнесенные во времени ряды,** т.е. уровни ряда представляют собой суммарные показатели за определенные и различные промежутки времени. |
| **3. В зависимости от вида соответствия** | **Детерминированные ряды**, у которых уровни ряда изменяются согласно определенной закономерности. | |
| **Стохастические (Случайные) ряды**, у которых изменение уровней не подчиняется никакой закономерности, а носит случайный характер. | |
| **4. В зависимости от наличия основной тенденции** | **Стационарные ряды**, если свойства временного ряда не изменяются во времени, т.е. математическое ожидание, дисперсия и другие характеристики остаются постоянными. | |
| **Нестационарные ряды**, если уровни и их числовые характеристики зависят от момента их измерения. | |

Основные задачи, возникающие при изучении временных рядов:

* Характеристика отдельных изменений в уровнях ряда от периода к периоду или от момента к моменту;
* Определение средних показателей за тот или иной период;
* Выявление основных закономерностей динамики исследуемого явления на отдельных этапах и в целом за рассматриваемый период;
* Выявление факторов, обусловливающих изменение изучаемого объекта во времени;
* Прогноз развития явления на будущее.

Для решения этих задач вычисляются определенные *показатели динамического ряда*, которые условно можно разделить на две группы:

* Показатели, характеризующие скорость и интенсивность развития явления;
* Средние показатели, дающие обобщающие характеристики.

При сравнении нескольких последовательных уровней возможны два варианта их сопоставления:

1) Каждый уровень динамического ряда сравнивается с одним и тем же предшествующим уровнем, принятым за основной, базисный уровень. Такое сравнение называется *сравнением с постоянной базой.*

2) Каждый уровень динамического ряда сравнивается с непосредственно ему предшествующим уровнем этого же ряда. Такое сравнение называется *сравнением с переменной базой.*

При этом различаются способы сопоставления:

1) Показатели динамики с постоянной базой *(базисные показатели)* характеризуют окончательный результат всех изменений в уровнях ряда от периода к которому относится базисный уровень, до данного (*i*-го) периода.

2) Показатели динамики с переменной базой *(цепные показатели)* характеризуют интенсивность изменения уровня от периода к периоду в пределах изучаемого промежутка времени.

Рассмотрим характерные показатели и способы их вычисления.

*Абсолютный прирост* *Δy* характеризует размер увеличения (или уменьшения) уровня ряда за определенный промежуток времени.

* Базисный абсолютный прирост
* Цепной абсолютный прирост

*Темп роста* *Тр* характеризует отношение двух уровней ряда и может выражаться в виде дроби (иногда, в этом случае, называют как коэффициент роста *К*) или в процентах, тогда *Тр=К∙100%*. Различают:

* Базисный темп роста
* Цепной темп роста

*Темп прироста* *Тп* характеризует абсолютный прирост в относительных величинах и показывает во сколько раз или на сколько процентов изменился данный уровень. Темп прироста может быть вычислен как разность между темпом роста и 100%:

*Тп=Тр-100%*

*Темп наращивания* *Тн* определяется как отношение цепного абсолютного прироста к базовому уровню :

Пример расчета показателей роста

Табл.3.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Момент времени, t | Уровень  ряда, Y | Абсолютный  прирост, ΔY | | Темп роста, Тр% | | Темп прироста, Тп% | |
| На 1  января | МРОТ,  руб. | Ц | Б | Ц | Б | Ц | Б |
| 2011 | 4330 |  | 0 |  | 100,0% |  | 0,0% |
| 2012 | 4611 | 281 | 281 | 106,5% | 106,5% | 6,5% | 6,5% |
| 2013 | 5205 | 594 | 875 | 112,9% | 120,2% | 12,9% | 20,2% |
| 2014 | 5554 | 349 | 1224 | 106,7% | 128,3% | 6,7% | 28,3% |
| 2015 | 5965 | 411 | 1635 | 107,4% | 137,8% | 7,4% | 37,8% |
| 2016 | 6204 | 239 | 1874 | 104,0% | 143,3% | 4,0% | 43,3% |
| 2017 | 7500 | 1296 | 3170 | 120,9% | 173,2% | 20,9% | 73,2% |

Средние показатели используют в качестве обобщающей характеристики динамики исследуемого явления.

Средние показатели можно разделить на две категории:

* Средние показатели уровней ряда;
* Средние показатели изменения уровней ряда.

Вычисление средних показателей уровней ряда осуществляется по следующим формулам:

Интервальный равноотстоящий ряд

Интервальный неравноотстоящий ряд

Моментальный ряд с равноотстоящими датами

Моментальный ряд с неравноотстоящими датами

Вычисление средних показателей изменения уровней ряда осуществляется по следующим формулам:

Средний абсолютный прирост

Средний темп роста

Или

Средний темп прироста

*Показатели вариации* (колебаний) уровней временного ряда аналогичны показателям вариации признака в статистических совокупностях:

Размах уровней

Среднее линейное отклонение ряда

Среднее квадратичное отклонение

Коэффициент вариации

**Определение тенденции изменения (тренда) временного ряда** (построение математической модели). Основная тенденция временного ряда, называемая *трендом*, проявляется как результат регулярного влияния большого числа факторов на величину уровней временного ряда и представляет собой детерминированную составляющую в формуле:

Первое слагаемое в данной формуле отражает детерминированную закономерность поведения уровней ряда и обычно подбирается аналитически (метод аналитического выравнивания) из известного набора функций, второе слагаемое представляет случайную ошибку в учете большого числа факторов на величину уровней временного ряда. *Метод аналитического выравнивания* заключается в замене фактических уровней *Y* теоретическими *Y\**, которые находятся по определенному уравнению, принятому за *математическую модель тренда*, т.е. *определяется функция* , которая наиболее точно соответствует детерминированной составляющей временного ряда.

В тех случаях, когда вид тренда известен (допустим при исследовании графическим методом), остается вычислить параметры функции.

Наиболее часто подбирают в качестве функции следующие представления зависимостей от времени.

Табл.4

|  |  |
| --- | --- |
| Вид функциональной зависимости | Формальное представление функции |
| Линейная зависимость |  |
| Квадратическая зависимость |  |
| Обратная зависимость |  |
| Степенная зависимость |  |
| Показательная зависимость |  |
| Экспоненциальная зависимость |  |

Следует отметить, что последние четыре функции путем соответствующих преобразований (логарифмирование) могут быть сведены к линейной зависимости. Вычисление параметров функциональной зависимости осуществляется методом наименьших квадратов (МНК), который реализует минимум функционала, равного сумме квадратов отклонений фактических уровней от выбранной функциональной зависимости.

Рассмотрим пример построения линейной модели (тренда) временного ряда

Параметры линейного тренда для данных роста МРОТ в виде функции

найдем из решения системы линейных уравнений, полученных из необходимых условий существования минимума функционала, равного сумме квадратов отклонений фактических уровней от выбранной линейной зависимости.

Коэффициенты при искомых параметрах *a,b* находятся из таблицы 5

Табл.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Момент времени, t** | **Уровень  ряда, Y** | **Условный номер ряда** |  |  |
| **На 1  января** | **МРОТ,  руб.** | **n** | **n2** | **Y\*n** |
| **2011** | **4330** | **1** | **1** | **4330** |
| **2012** | **4611** | **2** | **4** | **9222** |
| **2013** | **5205** | **3** | **9** | **15615** |
| **2014** | **5554** | **4** | **16** | **22216** |
| **2015** | **5965** | **5** | **25** | **29825** |
| **2016** | **6204** | **6** | **36** | **37224** |
| **2017** | **7500** | **7** | **49** | **52500** |
| **Сумма столбца** | **39369** | **28** | **140** | **170932** |

В результате получим систему

Решая её относительно параметров, получим:

*a =*3701,9 *b=*480,57.

Таким образом, уравнение линейного тренда имеет вид

На рис. 3 приведен график исходного ряда и линейного тренда.

Рис. 3 Графики исходных данных и линейного тренда

Как можно видеть из графиков, приближение линейным трендом весьма удовлетворительно. Критерий показывает, что при уровне значимости 0,05 ошибка составляет менее 6%. В данном случае аналитическое выравнивание, выполненное с помощью линейной зависимости вполне адекватно и может быть использовано для прогноза

Рис. 4. Прогноз по линейному тренду на три года.

**1.2. Методы сглаживания временного ряда.**

Метод скользящей средней (МСС)

Самый простой метод сглаживания (выравнивания) уровней временного ряда – метод скользящей средней (МСС). Суть его заключается в том, что каждое выбранное значение сглаженного ряда представляется как среднее арифметическое предыдущих значений. Выбор числа предшествующих значений определяется исследователем. Это количество называют «окном сглаживания». Чаще всего окно сглаживания выбирают равным 3. Формальное представление МСС задается соотношением

 (1)

Метод взвешенной скользящей средней (МВСС)

Иногда, для того, чтобы учесть влияние отдельных членов временного ряда, используют веса предшествующих значений ряда

 (2)

Выбор веса остается за исследователем.

Рассмотрим пример (для выполнения задания 1). В таблице приведены расходы на транспорт некоторой компании в млн. руб.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| y | 3 | 11 | 10 | 11 | 15 | 17 | 23 | 19 | 25 | 23 |
| *t* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Выберем окно сглаживания из 3-х значений временного ряда. Естественно, что в этом случае мы можем начинать процесс выравнивания с 4-го члена ряда. При этом первые 3 значения сохраняем.

В результате получим после расчетов по формулам (1) и (2) результат сглаживания временного ряда

Рис. 1 Сглаживание по трем предстоящим значениям с равными весами

Рис. 2 Сглаживание временного ряда в весами w1 = 0,1; w2 = 0,3; w3 = 0,6

Для закрепления материала по обработке временных рядов необходимо выполнить Лабораторную работу №1.

* 1. **Прогнозирование временных рядов с учетом неоднородности и сезонности**

Прогнозирование является статистической задачей и может осуществляться путем экстраполяции сформировавшейся к настоящему моменту времени тенденции на будущие моменты. Однако необходимо иметь в виду, что это допустимо при твердой уверенности в том, что не будет никаких принципиальных изменений в поведении временного ряда между последним наблюдаемым моментом времени и моментом, который нас интересует. Естественно, что точность оценки снижается с увеличением этого промежутка времени. При этом необходимо учитывать, что не только фактор тенденции определяет поведение временного ряда. Как определено выше, классификация временных рядов предполагает сложный характер поведения ряда в зависимости от его вида. Одними из наиболее сложных проблем изучения различных экономических процессов являются оценка, моделирование и прогнозирование финансово-экономических показателей. Для краткосрочного прогнозирования временных рядов можно использовать модель Брауна. Если нужно учесть тренд без учета сезонности, подходит модель Хольта. Однако часто исследуемые финансовые показатели имеют трендовую компоненту и подвержены сезонным колебаниям. Такие процессы удовлетворительно моделируются временными рядами, включающими в себя как тренд, так и сезонную компоненту (тренд-сезонные временные ряды). Одним из эффективных способов моделирования тренд-сезонных временных рядов, включая прогнозирование показателей экономического процесса, является модель Хольта – Уинтерса, которая является развитием модели Хольта. К достоинствам данной модели также относится ее достаточно простая реализация в различных пакетах прикладных программ, включая электронные таблицы *Excel*. Однако при использовании этой модели возникает необходимость подбора параметров модели, что может вызвать затруднения, поскольку алгоритм такого подбора неясен. Поэтому мы предлагаем простой, но достаточно эффективный алгоритм, включающий минимизацию функционала ошибки, который часто используется в теории искусственных нейронных сетей. Программная реализация данного алгоритма является достаточно простой и не должна вызывать затруднений.

* + 1. **Экспоненциальное сглаживание**

Метод экспоненциального сглаживания может быть использован как для сглаживания уровней временного ряда, так и для прогнозирования. Особенность этого метода заключена в том, что в процедуре выравнивания каждого наблюдения используются только значения предыдущих уровней, взятых с определенным весом. Вес каждого наблюдения уменьшается по мере его удаления от момента, для которого определяется сглаживаемое значение. Сглаженное значение наблюдения ряда на момент времени определяется по формуле:

(1)

Пусть β=1-α, тогда

(2)

Используя рекуррентное соотношение (1), (2) для моментов времени *t-1, t-2, t-k,* получим

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| год | месяц | t | y(t) | S(t) | y(t)p | SSE | alpha | beta | S(0) |
| 1981 | Jan | 1 | 114 | 128,8674 | 134,94 | 438,4836 | 0,29 | 0,71 | 134,94 |
|  | Feb | 2 | 141,3 | 132,4729 | 128,8674 | 154,5695 |  |  |  |
|  | Mar | 3 | 135,5 | 133,3507 | 132,4729 | 9,163613 |  |  |  |
|  | Apr | 4 | 156,4 | 140,035 | 133,3507 | 531,269 |  |  |  |
|  | May | 5 | 127,5 | 136,3999 | 140,035 | 157,1266 |  |  |  |
|  | Jun | 6 | 90 | 122,9439 | 136,3999 | 2152,947 |  |  |  |
|  | Jul | 7 | 143,8 | 128,9922 | 122,9439 | 434,9768 |  |  |  |
|  | Aug | 8 | 158,7 | 137,6074 | 128,9922 | 882,5552 |  |  |  |
|  | Sep | 9 | 167,3 | 146,2183 | 137,6074 | 881,6481 |  |  |  |
|  | Okt | 10 | 162,4 | 150,911 | 146,2183 | 261,848 |  |  |  |
|  | Nov | 11 | 137,5 | 147,0218 | 150,911 | 179,8544 |  |  |  |
|  | Dec | 12 | 150,1 | 147,9145 | 147,0218 | 9,475337 |  |  |  |
|  | Jan | 13 | 111,2 | 137,2673 | 147,9145 | 1347,953 |  |  |  |
|  | Feb | 14 | 163,6 | 144,9038 | 137,2673 | 693,4123 |  |  |  |
|  | Mar | 15 | 153,8 | 147,4837 | 144,9038 | 79,14296 |  |  |  |
|  | Apr | 16 | 122 | 140,0934 | 147,4837 | 649,4177 |  |  |  |
|  | May | 17 | 82,2 | 123,3043 | 140,0934 | 3351,647 |  |  |  |
|  | Jun | 18 | 116,4 | 121,3021 | 123,3043 | 47,66964 |  |  |  |
|  | Jul | 19 | 106,1 | 116,8935 | 121,3021 | 231,1029 |  |  |  |
|  | Aug | 20 | 118,8 | 117,4464 | 116,8935 | 3,634865 |  |  |  |
|  | Sep | 21 | 94,7 | 110,8499 | 117,4464 | 517,397 |  |  |  |
|  | Okt | 22 | 98,1 | 107,1524 | 110,8499 | 162,5604 |  |  |  |
|  | Nov | 23 | 127 | 112,9082 | 107,1524 | 393,9256 |  |  |  |
|  | Dec | 24 | 84,3 | 104,6118 | 112,9082 | 818,431 |  |  |  |
|  | Jan | 25 |  |  | 104,6118 | 14390,21 |  |  |  |
|  | Feb |  |  |  |  |  |  |  |  |

и т.д.

Подставляя полученные рекуррентные соотношения в (2), получим

(3)

Проанализируем соотношение (3). Сомножитель β=1-α, стоящий перед в каждом слагаемом, является относительным весом, который определяет величину вклада соответствующего уровня ряда в общую сумму. Поскольку β и α < 1, то с увеличением  *веса* значение уменьшается. Относительный вес каждого предшествующего уровня снижается по экспоненте по мере его удаления от момента, для которого вычисляется сглаженное значение, т.е. от давности наблюдения (отсюда произошло название этого метода сглаживания).

В лабораторной работе №2 работе необходимо реализовать алгоритм экспоненциального сглаживания в Excel для простейшего случая *k=*1.

Для вычисления величины *S(0)* в ячейке J2 набираем среднюю по первым пяти уровням ряда

=(D2+D3+D4+D5+D6)/5

В ячейках для вычисления сглаженных значений

=D2\*H$2+(1-H$2)\*J$2 и т.д.

Для подбора параметра α используется Поиск решения Excel при этом целевая ячейка располагается в итоговой сумме квадрата ошибки ESS, ограничения по параметру α очевидные: 0<α<1, необходимо в начале получить решение при α=0,5, затем при оптимальном значении.

* + 1. **Модель Хольта-Уинтерса**

Использование модели предполагает знание тренд-сезонных параметров. В первую очередь должен быть понятен механизм образования сезонной компоненты, так называемого периода сезонности *L.* Знание этого параметра можно получить из визуального представления ряда, либо из других источников. Трендовая составляющая определяется по части приведенных данных, как правило, включающих хотя бы 2 сезонных периода. Для определения тренда используется методика, описанная выше при аналитическом выравнивании ряда. Общий вид мультипликативной модели Хольта-Уинтерса представляется в виде произведения трендовой и сезонных компонент:

(1)

Где *t* – текущий момент времени, *k* – параметр упреждения (горизонт прогноза), *L* – период сезонности.

Для расчета по формуле Хольта –Уинтерса с линейным трендом

 (2)

Необходимо выполнить пошаговое вычисление коэффициентов

 (3)

 (4)

 (5)

Будем предполагать, что период сезонности составляет 4 временных интервала (допустим, 4 месяца). При этом вычисление сезонного коэффициента *F(t)* на первых четырех шагах потребует знания данных величин в «отрицательные» моменты времени *t=*-4; -3; -2; -1 . Для этого используются формулы аппроксимации с целью выравнивания значений ряда между фактическими и регрессионными

(6)

(7)

(8)

(9)

Рассмотрим пример:

Временной ряд представляет данные кредитования поквартально в течение 5 лет

Таблица 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 124 | 237 | 409 | 115 | 129 | 244 | 427 | 125 | 133 | 251 | 499 | 148 | 145 | 274 | 535 | 153 | 158 | 287 | 589 | 163 |
| *t* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

Начальные оценки параметров линейной модели этого ряда для первых 8 значений получены методом наименьших квадратов по данным таблицы 2

Таблица 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *t* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Σ |
|  | 124 | 237 | 409 | 115 | 129 | 244 | 427 | 125 | 1811 |
|  | 1 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 204 |
|  | 124 | 474 | 1227 | 460 | 645 | 1464 | 2989 | 1000 | 8388 |

Расчетные формулы метода наименьших квадратов

(10)

Где коэффициенты находятся из системы уравнений

= (11)

Для данного случая система имеет решение

Таким образом, уравнение линейной регрессии имеет вид

(12)

В результате по расчетным формулам (6)-(9) с учетом (12) получим

*F(-4)*=0,5\*(B3/J3+B7/J7) =0,5838

*F(-3)=* 0,5\*(B4/J4+B8/J8)= 1,0779

*F(-2)=* 0,5\*(B5/J5+B9/J9)= 1,8262

*F(-1)=* 0,5\*(B6/J6+B10/J10)= 0,511

Здесь в скобках записаны адреса ячеек в Excel таблице, соответствующие значениям уровней в формулах (6)-(9) (см. приложение к лабораторной работе №3)

Параметры сглаживания принимаем равными

При вычислении по формуле Хольта-Уинтерса полагаем *k* равным 1, *t* начинаем с 0. В результате получим таблицу 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| t | y(t) | a(t) | b(t) | F(t) | Y(t) | y(t)-Y(t) | 100%\*[y(t)-(t)]/y(t) |
| 1 | 124 | 208,27 | 6,21 | 0,59 | 125,22 | -1,22 | 0,98 |
| 2 | 237 | 216,10 | 6,69 | 1,09 | 240,14 | -3,14 | 1,33 |
| 3 | 409 | 223,14 | 6,80 | 1,83 | 419,92 | -10,92 | 2,67 |
| 4 | 115 | 228,47 | 6,36 | 0,51 | 120,01 | -5,01 | 4,35 |
| 5 | 130 | 230,40 | 5,03 | 0,57 | 139,08 | -9,08 | 6,99 |
| 6 | 244 | 232,00 | 4,00 | 1,07 | 257,06 | -13,06 | 5,35 |
| 7 | 427 | 235,20 | 3,76 | 1,82 | 437,34 | -10,34 | 2,42 |
| 8 | 125 | 241,32 | 4,47 | 0,51 | 124,47 | 0,53 | 0,42 |
| 9 | 133 | 241,46 | 3,17 | 0,56 | 140,63 | -7,63 | 5,73 |
| 10 | 251 | 241,83 | 2,33 | 1,05 | 260,45 | -9,45 | 3,76 |
| 11 | 499 | 253,11 | 5,01 | 1,91 | 470,13 | 28,87 | 5,79 |
| 12 | 148 | 267,17 | 7,73 | 0,54 | 141,12 | 6,88 | 4,65 |
| 13 | 149 | 272,19 | 6,92 | 0,55 | 156,42 | -7,42 | 4,98 |
| 14 | 274 | 273,70 | 5,30 | 1,02 | 292,79 | -18,79 | 6,86 |
| 15 | 535 | 279,27 | 5,38 | 1,91 | 544,08 | -9,08 | 1,70 |
| 16 | 153 | 284,61 | 5,37 | 0,54 | 155,92 | -2,92 | 1,91 |
| 17 | 158 | 288,76 | 5,00 | 0,55 | 162,34 | -4,34 | 2,74 |
| 18 | 287 | 290,01 | 3,88 | 1,00 | 299,88 | -12,88 | 4,49 |
| 19 | 589 | 298,04 | 5,12 | 1,95 | 580,25 | 8,75 | 1,49 |
| 20 | 163 | 303,17 | 5,12 | 0,54 | 165,75 | -2,75 | 1,68 |
| 210 | 5150 |  |  |  |  | -4,15 | 3,51 |

На графике построены фактические и расчетные значения показателя кредитования поквартально

Рис.1 Расчет тренд-сезонных колебаний цены по модели Хольта-Уинтерса

Для расчета параметров и критериев адекватности построим таблицу 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Оценка адекватности | t | Et=y(t)-Y(t) | T(точки поворотов) | (E(t)-E(t-1))^2 | Et^2 |
| Таблица 4 | 1 | -1,22 |  |  | 1,48 |
|  | 2 | -3,14 | 0 | 3,71 | 9,88 |
|  | 3 | -10,92 | 1 | 60,44 | 119,18 |
|  | 4 | -5,01 | 1 | 34,93 | 25,07 |
|  | 5 | -9,08 | 0 | 16,60 | 82,46 |
|  | 6 | -13,06 | 0 | 15,81 | 170,46 |
|  | 7 | -10,34 | 1 | 7,39 | 106,88 |
|  | 8 | 0,53 | 1 | 118,08 | 0,28 |
|  | 9 | -7,63 | 0 | 66,48 | 58,15 |
|  | 10 | -9,45 | 1 | 3,32 | 89,28 |
|  | 11 | 28,87 | 1 | 1468,24 | 833,41 |
|  | 12 | 6,88 | 0 | 483,66 | 47,29 |
|  | 13 | -7,42 | 0 | 204,38 | 55,05 |
|  | 14 | -18,79 | 1 | 129,24 | 352,99 |
|  | 15 | -9,08 | 0 | 94,21 | 82,48 |
|  | 16 | -2,92 | 1 | 37,91 | 8,55 |
|  | 17 | -4,34 | 0 | 1,99 | 18,80 |
|  | 18 | -12,88 | 1 | 73,06 | 165,98 |
|  | 19 | 8,75 | 1 | 467,95 | 76,54 |
|  | 20 | -2,75 |  | 132,12 | 7,54 |
| Сумма |  | -82,99 | 10 | 3419,53 | 2311,75 |
| Ср. квадр. Отклонение |  | 10,18 |  |  |  |

Рассмотрим адекватность и качество модели

1. Проверка случайности ряда остатков устанавливает отсутствие систематической ошибки. Для этого используется критерий числа поворотных точек в распределении ряда остатков (третий столбец табл. 4). Точка в этом распределении считается поворотной, если она меньше или больше двух соседних точек. В четвертом столбце производится подсчет этих точек *p.* Если это число удовлетворяет неравенству

То это означает, уровни ряда остатков являются случайными и систематическая ошибка отсутствует. В нашем случае *p=10, N=20,* следовательно *p> 8* (квадратные скобки означают целую часть числа)

1. Отсутствие автокорреляции (взаимная независимость ряда остатков) проверяется с помощью критерия Дарбина Уотсона

или , где

*d* = = 1,4792

Следовательно, по критерию Дарбина-Уотсона, автокорреляция отсутствует

1. Проверка соответствия ряда остатков нормальному распределению. Для этого используется критерий размаха *RS.*

среднее квадратическое отклонение ряда остатков. Если оно попадает в интервал

[3,18; 4,49] , то можно вычислить доверительную вероятность для прогноза по формуле Хольта –Уинтерса для уровня значимости α=0,05. В нашем случае этого сделать нельзя, т.к. *RS= 4,68.*

|  |  |  |  |
| --- | --- | --- | --- |
| Прогноз на четыре шага | t | t1 | Yprognoz |
|  | 21 | 1 | 169,3583354 |
|  | 22 | 2 | 314,0215673 |
|  | 23 | 3 | 621,5803729 |
|  | 24 | 4 | 174,0150094 |

Однако, по совокупности других критериев модель адекватна и позволяет сделать прогноз по формуле Хольта-Уинтерса 21-24 кварталы. Результаты прогноза в таблице 5

Таблица 5.

Для усвоения материала необходимо выполнить лабораторную работу №3

* + 1. **Адаптивная модель Брауна**

Модель Брауна относится к адаптивным моделям прогнозирования, способным изменять свою структуру и параметры, приспосабливаясь к изменению условий. Все адаптивные модели делятся на два класса: модели скользящего среднего (СС-модели) и авторегрессии (АР-модели).

Согласно схеме скользящего среднего оценкой текущего уровня (наблюдения) является взвешенное среднее всех предшествующих уровней, причем вес (множитель), который отражает информационную ценность наблюдения, тем больше, чем ближе оно находится к текущему уровню. Такие модели хорошо отражают тенденцию, но не позволяют отражать колебания, например, сезонные.

В СС-моделях сглаживание производятся с помощью параметра сглаживания, который принимает значения в интервале от 0 до 1. Параметр сглаживания принимает значение больше 0,5 для быстроизменяющихся процессов и меньше 0,5 для относительно стабильных процессов.

**Формализация модели Брауна**

Модель Брауна описывает процессы с линейной и параболической тенденцией (трендом), а также случайные процессы без тенденции. Построение линейной модели Брауна имеет следующие этапы:

1. По первым пяти точкам временного ряда с помощью метода наименьших квадратов оцениваются значения параметров линейной модели для нулевого момента времени:

(1)

2. С использованием параметров *a*0 и *a*1 , найденных на предыдущем этапе, находим прогноз на шаг вперед ( *t*= 1):

*y (*1) = *a*0(0) + *a*1(0)t = *a*0(0) + *a*1(0) . (2)

3. Находим величину отклонения фактического значения экономического показателя от расчетного (в данном случае t = 1):

= *y*(t) – *yp*(t) . (3)

4. Корректируем параметры модели по формулам:

*a*0(t) = *a*0(t-1) + *a*1(t-1) + (1 – 2)(t) , (4)

*a*1(t) = *a*1(t-1) + , (5)

где = параметр сглаживания.

5. С помощью скорректированных на предыдущем шаге параметров находим прогноз на следующий момент времени ( = 1):

*yp*() = *a*0(t) + *a*1(t) . (5)

Точечный прогноз на будущее рассчитывается по формуле

*yp*(n + ) = *a*0(n) + *a*1(n) , = 1, 2, … (6)

Здесь n – число наблюдений.

**Построение модели Брауна в Excel**

Рассмотрим пример использования модели Брауна. Данные об использовании топлива приведены в таблице 1

Таблица 1

Данные представлены в виде показателя по наблюдениям y(t)

|  |  |  |  |
| --- | --- | --- | --- |
| год | месяц | t | y(t) |
| 1981 | Jan | 1 | 114 |
|  | Feb | 2 | 141,3 |
|  | Mar | 3 | 135,5 |
|  | Apr | 4 | 156,4 |
|  | May | 5 | 127,5 |
|  | Jun | 6 | 90 |
|  | Jul | 7 | 143,8 |
|  | Aug | 8 | 158,7 |
|  | Sep | 9 | 167,3 |
|  | Okt | 10 | 162,4 |
|  | Nov | 11 | 137,5 |
|  | Dec | 12 | 150,1 |
|  | Jan | 13 | 111,2 |
|  | Feb | 14 | 163,6 |
|  | Mar | 15 | 153,8 |
|  | Apr | 16 | 122 |
|  | May | 17 | 82,2 |
|  | Jun | 18 | 116,4 |
|  | Jul | 19 | 106,1 |
|  | Aug | 20 | 118,8 |
|  | Sep | 21 | 94,7 |
|  | Okt | 22 | 98,1 |
|  | Nov | 23 | 127 |
|  | Dec | 24 | 84,3 |
|  | Jan | 25 |  |

Результаты вычисления тренда по первым пяти наблюдениям

Таблица 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t | y | t^2 | y\*t | a0 | a1 | y^ |
| 1 | 114 | 1 | 114 | 247,8786 | -4,01429 | 243,8643 |
| 2 | 141,3 | 4 | 282,6 |  |  | 239,85 |
| 3 | 135,5 | 9 | 406,5 |  |  | 235,8357 |
| 4 | 156,4 | 16 | 625,6 |  |  | 231,8214 |
| 5 | 127,5 | 25 | 637,5 |  |  | 227,8071 |
| 6 | 90 | 36 | 540 |  |  |  |
| 21 | 764,7 | 91 | 2606,2 |  |  |  |
| 3,5 | 127,45 |  |  |  |  |  |

Результаты расчета по модели Брауна (1)-(6)

Таблица 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Модель Брауна и прогноз a0 a1 | | | | | | Yp | e(abs) | e(rel) |
|  |  | 1 | 114 | 183,4789 | -68,414 | 243,8643 | -129,864 | 16864,73 |
|  |  | 2 | 141,3 | 128,0749 | -55,404 | 115,0649 | 26,23511 | 688,2811 |
|  |  | 3 | 135,5 | 103,8278 | -24,247 | 72,67089 | 62,82911 | 3947,497 |
|  |  | 4 | 156,4 | 117,6754 | 13,8476 | 79,58081 | 76,81919 | 5901,188 |
|  |  | 5 | 127,5 | 129,528 | 11,85257 | 131,523 | -4,02305 | 16,18491 |
|  |  | 6 | 90 | 115,901 | -13,6271 | 141,3806 | -51,3806 | 2639,965 |
|  |  | 7 | 143,8 | 122,8667 | 6,965734 | 102,2739 | 41,52611 | 1724,418 |
|  |  | 8 | 158,7 | 144,1479 | 21,28117 | 129,8324 | 28,86758 | 833,337 |
|  |  | 9 | 167,3 | 166,3568 | 22,20898 | 165,429 | 1,87098 | 3,500567 |
|  |  | 10 | 162,4 | 175,5902 | 9,233353 | 188,5658 | -26,1658 | 684,6503 |
|  |  | 11 | 137,5 | 161,3558 | -14,2344 | 184,8235 | -47,3235 | 2239,518 |
|  |  | 12 | 150,1 | 148,5985 | -12,7573 | 147,1214 | 2,978594 | 8,872025 |
|  |  | 13 | 111,2 | 123,6216 | -24,9769 | 135,8412 | -24,6412 | 607,1879 |
|  |  | 14 | 163,6 | 130,8561 | 7,234438 | 98,64475 | 64,95525 | 4219,185 |
|  |  | 15 | 153,8 | 145,8808 | 15,02478 | 138,0905 | 15,7095 | 246,7885 |
|  |  | 16 | 122 | 141,6123 | -4,26852 | 160,9056 | -38,9056 | 1513,647 |
|  |  | 17 | 82,2 | 109,998 | -31,6143 | 137,3438 | -55,1438 | 3040,839 |
|  |  | 18 | 116,4 | 97,23596 | -12,762 | 78,38366 | 38,01634 | 1445,242 |
|  |  | 19 | 106,1 | 95,1983 | -2,03766 | 84,47394 | 21,62606 | 467,6866 |
|  |  | 20 | 118,8 | 105,8752 | 10,6769 | 93,16064 | 25,63936 | 657,3769 |
|  |  | 21 | 94,7 | 105,7156 | -0,15956 | 116,5521 | -21,8521 | 477,514 |
|  |  | 22 | 98,1 | 101,8586 | -3,85703 | 105,5561 | -7,45608 | 55,59318 |
|  |  | 23 | 127 | 112,3819 | 10,52329 | 98,00158 | 28,99842 | 840,9082 |
|  |  | 24 | 84,3 | 103,7609 | -8,62102 | 122,9052 | -38,6052 | 1490,36 |
|  |  | 25 |  |  |  | 95,13985 |  |  |
|  |  |  |  |  |  | 86,51882 |  |  |

Прогноз на следующие три момента времени

|  |
| --- |
| 95,13985 |
| 86,51882 |
| 77,8978 |

Оценка качества модели производится по данным следующей таблицы

Таблица 4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Проверка качества модели | | | |  |  |  |  |  |
|  |  | y(t) | Et | Et^2 | Et-Et-1 | (Et-Et-1)^2 | Et(rel) | %Et(rel) |
|  |  | 114 | -12,52 | 156,7504 |  |  | 0,109825 | 10,98246 |
|  |  | 141,3 | 27,52321 | 757,5272 | 40,04321 | 1603,459 | 0,194786 | 19,47857 |
|  |  | 135,5 | -4,12909 | 17,04939 | -31,6523 | 1001,868 | 0,030473 | 3,047299 |
|  |  | 156,4 | 9,128267 | 83,32526 | 13,25736 | 175,7575 | 0,058365 | 5,836488 |
|  |  | 127,5 | -38,8067 | 1505,963 | -47,935 | 2297,765 | 0,304367 | 30,43666 |
|  |  | 90 | -51,1976 | 2621,198 | -12,3909 | 153,5343 | 0,568863 | 56,88626 |
|  |  | 143,8 | 60,07258 | 3608,715 | 111,2702 | 12381,06 | 0,417751 | 41,77509 |
|  |  | 158,7 | 45,244 | 2047,02 | -14,8286 | 219,8866 | 0,285091 | 28,50914 |
|  |  | 167,3 | 8,636585 | 74,59061 | -36,6074 | 1340,103 | 0,051623 | 5,162334 |
|  |  | 162,4 | -27,2361 | 741,804 | -35,8727 | 1286,848 | 0,16771 | 16,77098 |
|  |  | 137,5 | -50,9918 | 2600,164 | -23,7557 | 564,3345 | 0,370849 | 37,08495 |
|  |  | 150,1 | 0,792552 | 0,628139 | 51,78436 | 2681,62 | 0,00528 | 0,528016 |
|  |  | 111,2 | -25,6802 | 659,4742 | -26,4728 | 700,8081 | 0,230937 | 23,09373 |
|  |  | 163,6 | 65,6884 | 4314,966 | 91,36863 | 8348,226 | 0,401518 | 40,15183 |
|  |  | 153,8 | 14,92847 | 222,8592 | -50,7599 | 2576,57 | 0,097064 | 9,706417 |
|  |  | 122 | -39,5963 | 1567,869 | -54,5248 | 2972,953 | 0,32456 | 32,456 |
|  |  | 82,2 | -54,2215 | 2939,975 | -14,6252 | 213,897 | 0,659629 | 65,96294 |
|  |  | 116,4 | 40,19034 | 1615,263 | 94,41188 | 8913,602 | 0,345278 | 34,52778 |
|  |  | 106,1 | 21,60123 | 466,6132 | -18,5891 | 345,5549 | 0,203593 | 20,35931 |
|  |  | 118,8 | 24,47921 | 599,2318 | 2,877979 | 8,282765 | 0,206054 | 20,6054 |
|  |  | 94,7 | -23,3653 | 545,936 | -47,8445 | 2289,095 | 0,246729 | 24,67294 |
|  |  | 98,1 | -7,4701 | 55,80236 | 15,89518 | 252,6566 | 0,076148 | 7,614779 |
|  |  | 127 | 29,63744 | 878,3776 | 37,10753 | 1376,969 | 0,233366 | 23,33656 |
|  |  | 84,3 | -38,8225 | 1507,186 | -68,4599 | 4686,762 | 0,460528 | 46,05278 |
|  |  |  |  | 29588,29 |  |  |  | 23,27072 |

Как можно видеть, точность неудовлетворительна и составляет 23%. Низкая точность метода обусловлена ролью параметра сглаживания. В следующем разделе мы рассмотрим также и другие причины снижения точности прогнозирования, обусловленные внутренними свойствами временного ряда, в частности, нестационарностью. Для усвоения материала этого раздела необходимо выполнить лабораторную работу № 4.

* 1. **Моделирование авторегрессии временных рядов**

**1.4.1 Модели авторегрессии.**

Процесс называется авторегрессионным (авторегрессией), если уровни временного ряда являются зависимыми от предыдущих значений. При этом, если текущее значение уровня зависит только от одного предыдущего значения, то то такой процесс называется авторегрессией первого порядка, если от двух – авторегрессией второго порядка и т.д.

Будем рассматривать линейную зависимость уровней, т. е линейную авторегрессию. Модель авторегрессии будем называть линейную функцию от предыдущих значений. Например, модель авторегрессии третьего порядка можно представить в виде

(1)

При этом существенным является требование стационарности авторегрессионного процесса. Проверка стационарности временного ряда осуществляется по коэффициентам выбранной модели или с помощью числовых характеристик временного ряда. Так, например, если среднее значение уровней временного ряда равно нулю, то ряд является стационарным, однако, финансовые показатели редко имеют такое значение средних. Тогда ориентируются на вычисленную дисперсию временного ряда. Если ряд стационарный, то его дисперсия конечна, в противном случае дисперсия растет с ростом времени t.

Разновидностью авторегрессионных моделей являются модели скользящей средней, которые строятся на основании линейной комбинации прошлых ошибок между фактическими и модельными значениями ряда:

(2)

Комбинированные модели включают модель скользящей средней и проинтегрированную часть, они называются авторегрессионными проинтегрированными моделями скользящей средней (АРПСС). Например, модель, включающая 3 лага авторегресии и 2 лага скользящей средней выглядит так:

(3)

Для определения степени автокорреляции необходимо использовать и оценить значения коэффициентов автокорреляции , с *i –*тым лагом. Коэффициент автокорреляции называется частным коэффициентом автокорреляционной функции (ЧАКФ) если он измеряет связь между текущим значением переменной и предыдущими значениями этой переменной , , …., когда влияние всех предыдущих лагом устранено. Частные коэффициенты автокорреляции определяются по соотношениям

(4)

(5)

Для учета тренда в уравнение (3) добавляют трендовую составляющую и с учетом дифференцирования ряда (вычисления конечных разностей) уравнение (3) для авторегрессии первого порядка запишется в виде системы двух уравнений – для уровней ряда и для разности первого порядка

(6)

(7)

Где: – трендовая составляющая, *T =*1, 2, ….*n*; – случайная компонента. Выбор порядка авторегрессии и, соответственно, порядка разностного оператора определяется с помощью коэффициентов корреляции или по количественному значению дисперсий исходного ряда и разностных рядов. Если видно, что дисперсия не увеличивается, то это сигнализирует об оптимальном выборе порядка. Более точная оценка получается когда рассматривается отношение величины дисперсии к коэффициенту – числу сочетаний из *2k* по *k.* Например, /, т.е. относительная дисперсия ряда первых разностей близка по значению относительной дисперсии ряда вторых разностей, то следует в качестве оптимального порядка разностного оператора выбрать 1-й порядок. Такой способ определения порядка предпочтительнее, чем по значениям корреляционных коэффициентов ввиду простоты реализации.

Исследование адекватности и качества выбранной модели производится по нескольким критериям.

1. Критерий Дики –Фуллера для уровня значимости 0,05: сначала вычисляется статистический критерий
2. Критерий Дарбина-Уотсона (см. Лаб. работу 2)

**Пример**

Дан временной ряд объема продаж акций за последние 13 дней. Требуется определить прогнозные значения данного показателя на следующие два месяца с использованием авторегрессионной модели.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| y(t) | 60 | 70 | 55 | 80 | 90 | 65 | 70 | 75 | 60 | 80 | 90 | 100 | 95 |

1. Вычислим для исходного ряда разности первого и второго порядка и построим графики

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 60 | 70 | 55 | 80 | 90 | 65 | 70 | 75 | 60 | 80 | 90 | 100 | 95 |
|  | - | 10 | -15 | 25 | 10 | -25 | 5 | 5 | -10 | 20 | 10 | 10 | -5 |
|  | - | - | -25 | 40 | -15 | -35 | 30 | 0 | -15 | 30 | -10 | 0 | -15 |

Для определения порядка разностного уравнения определиv дисперсии исходного ряда и разностей

Как можно видеть, значения относительной дисперсии стабилизировались и выбор 1-го порядка разностного оператора обоснован. Учитывая наличие тренда, создадим две модели.

Модель с положительным средним значением

(8)

Модель с положительным средним значением и трендом

(9)

Рассмотрим 1-ю модель. Для определения параметров модели воспользуемся методом наименьших квадратов

(10)

Вычислив суммы, получим:

Решая данную систему, получим:

Уравнение (8) примет вид:

Аналогичным образом вычислим коэффициенты для уравнения (9)

Вычислим прогноз по первой модели

Прогноз по второй модели является более точным

**1.4.2 Критерии качества моделей**

**Проверка качества** модели по модифицированному критерию Дики-Фуллера выполняется с помощью таблицы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Коэфф. | Без учета средней и тренда | | С учетом средней | | С учетом средней и тренда | |
| значимость | 1% | 5% | 1% | 5% | 1% | 5% |
|  | -2,57 | -1,94 | -3,43 | -2,86 | -3,96 | -3,41 |
|  | -1,96 | -0,398 | -6,0 | -2,74 | -8,35 | -4,04 |
|  | -10,04 | 0 | -29,25 | -8,36 | -47,44 | -17,83 |

Критерий Дики-Фуллера

Если выполняется условие , то ряд считается стационарным, в противном случае необходимо использовать разностные методы для приведения его к стационарному.

В нашем примере для 1-й модели значение критерия *DF* равно

-2,86-2,74/11-8,36/121=-3,178

Значит, по критерию *DF* -3,178<-1,762 что означает: исходный ряд нестационарный и конечные разности 1-го порядка приводят его к стационарному.

Для 2-й модели значение критерия *DF* равно

-3,41-4,04/11-17,83/121=-3,924<-2,449

Значит для второй модели, по критерию *DF* -3,924<-2,449, следовательно,

исходный ряд нестационарный и конечные разности 1-го порядка приводят его к стационарному. Чтобы обосновать наилучший выбор, необходимо использовать дополнительные критерии.

Критерий Акаике AIС

Критерий Шварца BIC

По этим критериям выбираем ту модель, где значения критериев меньше.

Для первой модели

Для второй модели

Таким образом, модель 2, что ожидаемо, дает лучший результат

Для усвоения материала необходимо выполнить работу №5

# Программная среда R

### 2.1 История создания среды R

В августе 1993 г. двое новозеландских учёных анонсировали свою новую разработку, которую они назвали *R*. По замыслу создателей (Robert Gentleman и Ross Ihaka), это должна была быть новая реализация языка *S*, отличающаяся от *S* некоторыми деталями, например, обращением с глобальными и локальными переменными, а также работой с памятью. Фактически, они создали не полный аналог *S*, а новое «ответвление» на «дереве *S*». Сначала проект развивался довольно медленно, но когда появилась платформа на базе интернета для написания дополнений и специализированных пакетов, всё большее количество людей стало переходить с *S* на *R*. Количество книг, написанных про *R* за последние годы, выросло в несколько раз [1, 4, 5, 13, 14], а количество пакетов уже приближается к полутора тысячам.

Идея центральной системы хранения и распространения пакетов CRAN известного как Comprehensive R Archive Network (http://cran.r-project.org/) была заимствована из TEX-сообщества (CTAN, или Comprehensive TeX Archive Network; аналогичной схемой пользуется и Perl-сообщество: CPAN или Comprehensive Perl Archive Network). Все три упомянутых проекта объединяет одно: стабильная база и множество дополнений. Cообщество CRAN публикует специальный журнал [19], в котором многочисленные пользователи могут найти последние новости, касающиеся среды R (новые пакеты, модификации, конференции и т.д.). Кроме того следует отметить, что в данном пособии мы будем использовать большей частью книгу [4], текст которой доступен на английском языке на сайте [www.StatLearning.com](http://www.StatLearning.com) Также на этом сайте доступны все базы данных, которые используются при выполнении лабораторных работ. Часть примеров приводится из книги [5], выпущенной в переводе на русский язык и доступной on line.

### 2.2 Установка и обновление среды R

Для установки среды R необходимо:

1. Скачать установщик с сайта *http://cran.r-project.org/*

2) Выполнить установку согласно данным инструкциям.

В среде R отдельную команду можно набрать вручную, либо подготовить при помощи блокнота последовательность команд, а затем вставить эту последовательность в качестве функции:

> pr1<-function(n){

+ print("Hello World!")

+ }

> pr1(1)

[1] "Hello World!"

>

Любая строка в коде, начинающаяся с символа *#* означает комментарий. Узнать все доступные в текущем окружении имена можно с помощью функции *ls().* Удалить некоторые имена *x* и *x0* можно с помощью функции *rm(x, x0),* полностью очистить текущее окружение можно следующим способом *rm(list= ls()).* Выход из среды R осуществляется при помощи команды *q().*

### 2.3 Установка и обновление дополнительных пакетов

Среда R предлагает пользователям большой набор пакетов различных типов [1], позволяющих строить графики или выполнять классификационный либо регрессионный анализ. Их полный список доступен на сайте: [*http://cran.r-project.org/*](http://cran.r-project.org/)

Установку пакетов можно производить как вручную, загрузив файл с сайта, так и из консоли, выполнив определённые команды.

Для установки пакета - *install.packages(‘имя пакета’).*

Для подключения пакета к рабочей среде - *require(‘имя пакета’)* либо *library(‘имя пакета’).*

Например, следующая последовательность команд устанавливает пакет “*install*”, а затем обновляет версию R.

install.packages(installr)

require(installr)

updateR()

Следует отметить, что проект R находится в постоянном развитии. По этой причине необходимо обновлять не только среду R, но и все используемые пакеты не реже чем раз в полгода.

### Работа с данными

##### 2.4.1 Работа с матрицами

В то время как векторы представляют одномерную последовательность данных определённых типов, во многих случаях удобнее оперировать многомерными данными. Например, матрицы представляют собой двумерные векторы. Для преобразования вектора в матрицу можно присвоить вектору атрибут размерности с помощью функции *dim(x).* Данные в матрице хранятся по столбцам.

> x = 1:16

> dim(x) = c(16)

> x

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Другим способом создания матриц является использование функции *matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE)*. Параметр *data* принимает вектор, элементами которого будет проинициализирована созданная матрица. Параметры *nrow* и *ncol* определяют количество строк и столбцов соответственно, *byrow* определяет будет ли заполнена матрица элементами из *data* по строкам или по столбцам.

>x = 1:16

> y = matrix(data = x , nrow = 2, ncol = 8)

> x

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

> y

[ ,1] [ ,2] [ ,3] [ ,4] [ ,5] [ ,6] [ ,7] [ ,8]

[1 ,] 1 3 5 7 9 11 13 15

[2 ,] 2 4 6 8 10 12 14 16

> dim(y)

[1] 2 8

Узнать количество строк и столбцов матрицы можно, получив весь вектор размерностей с помощью функции dim, либо используя функции *nrow(x)* и *ncol(x).*

Доступ к элементам матрицы *A* происходит по индексу. *A[i, j]* ссылается на элемент *i*-й строки и *j*-го столбца матрицы *A*. На месте индексов *i* и *j* могут стоять векторы. Векторы *i* и *j* могут быть логическими или символьными. В последнем случае строкам и столбцам матрицы должны быть присвоены имена функций *rownames(x)* и *colnames(x).* [1, c. 446]

*A[i, ]* эквивалентно *A[i, 1:ncol(A)]*, а *A[, j]* эквивалентно *A[1:nrow(A), j]*. Возможен доступ к элементам матрицы с помощью одного индекса. Например, *A[5]* означает 5-й элемент матрицы *A.*

> m = matrix(1:6 , nrow = 2, ncol = 3)

> m

[ ,1] [ ,2] [ ,3]

[1 ,] 1 3 5

[2 ,] 2 4 6

> m [1 , 3]

[1] 5

> A = matrix(c(23 , 31 , 58 , 16) , nrow = 2)

> rownames(A) <- c("petal", "sepal")

> colnames(A) <- c("length", "width")

> A

length width

petal 23 58

sepal 31 16

> A["petal", "length"]

[1] 23

Для того, чтобы объединить несколько матриц в одну, используются функции *rbind(...)* (объединяет матрицы друг с другом снизу) и *cbind(...)* (объединяет матрицы друг с другом справа).

> A = matrix(1:4 , nrow = 2, ncol = 2)

> B = matrix(5:8 , nrow = 2, ncol = 2)

> cbind(A , B)

[ ,1] [ ,2] [ ,3] [ ,4]

[1 ,] 1 3 5 7

[2 ,] 2 4 6 8

> rbind(A , B)

[ ,1] [ ,2]

[1 ,] 1 3

[2 ,] 2 4

[3 ,] 5 7

[4 ,] 6 8

Рассмотрим еще несколько функций, часто используемых при работе с матрицами:

* t(A). Транспонирует матрицу.
* diag(A). Возвращает вектор диагональных элементов матрицы, либо создает диагональную матрицы с указанными значениями диагонали.
* det(A). Вычисляет определитель матрицы.
* solve(A, b). Решает систему линейных уравнений Ax = b с квадратной и невырожденной матрицей A. При этом solve(A) может применяться для отыскания обратной матрицы.
* eigen(A). Вычисляет собственные числа и векторы. Возвращает список, содержащий собственные числа и матрицу собственных векторов.
* chol(A). Выполняет разложение Холецкого для симметричной положительно определенной матрицы.
* qr(A). Выполняет QR-разложение матрицы. Возвращает список, содержащий полученные в результате разложения матрицы.
* qr.solve(A, b, tol = 1e-7). Решает систему линейных уравнений через QR-разложение.

**Примеры операций с матрицами**

Заданы две матрицы А и В

A=; B=

Выполнить следующие операции: 1) транспонировать обе матрицы, 2) вычислить определители, 3) объединить обе матрицы

1. Операция транспонирования:

> A= matrix(c(1, 5, 3, 3, 7, 6, 2, 8, 1),nrow = 3)

> B= matrix(c(1, 0, 0, 9, 1, 0, 20, 3, 1),nrow = 3)

> A

[,1] [,2] [,3]

[1,] 1 3 2

[2,] 5 7 8

[3,] 3 6 1

> t(A)

[,1] [,2] [,3]

[1,] 1 5 3

[2,] 3 7 6

[3,] 2 8 1

> B

[,1] [,2] [,3]

[1,] 1 9 20

[2,] 0 1 3

[3,] 0 0 1

> t(B)

[,1] [,2] [,3]

[1,] 1 0 0

[2,] 9 1 0

[3,] 20 3 1

1. Вычисление определителя

> A= matrix(c(1, 5, 3, 3, 7, 6, 2, 8, 1),nrow = 3)

> B= matrix(c(1, 0, 0, 9, 1, 0, 20, 3, 1),nrow = 3)

> det(A)

[1] 34

> det(B)

[1] 1

1. Объединение «сверху»

> rbind(A, B)

[,1] [,2] [,3]

[1,] 1 3 2

[2,] 5 7 8

[3,] 3 6 1

[4,] 1 9 20

[5,] 0 1 3

[6,] 0 0 1

>

##### Массивы данных

Ресурсы среды R позволяют обрабатывать массивы данных любой размерности. Работа с массивами почти идентична работе с матрицами. Создать массив можно либо изменив размерность вектора или матрицы с помощью функции *dim(x),* либо используя функцию *array (data = NA, dim = length(data)).*

Для проверки является ли некоторая переменная вектором (не атомарным, а общим), матрицей или массивом используются функции *is.vector(x), is.matrix(x), is.array(x).*

С помощью функции *colnames(x)* можно изменить имена столбцов фрейма, с помощью *rownames(x)* – имена строк.

> x = 1:24

> dim(x) = c(2, 3, 4)

> x

, , 1

[ ,1] [ ,2] [ ,3]

[1 ,] 1 3 5

[2 ,] 2 4 6

, , 2

[ ,1] [ ,2] [ ,3]

[1 ,] 7 9 11

[2 ,] 8 10 12

, , 3

[ ,1] [ ,2] [ ,3]

[1 ,] 13 15 17

[2 ,] 14 16 18

, , 4

[ ,1] [ ,2] [ ,3]

[1 ,] 19 21 23

[2 ,] 20 22 24

Для загрузки наборов данных из файла в структуру или таблицу может быть использована функция *read.table(file, header = FALSE, sep = "", ...),* в которой на вход необходимо задать путь к текстовому файлу с данными. Параметр *header* позволяет указать, следует ли интерпретировать первую строку файла, как имена столбцов таблицы.

##### Факторные данные

Факторы представляют собой структуру данных для хранения векторов категориальных данных (классов), т.е. величин, которые могут принимать значения из конечного и неупорядоченного множества.

Факторы создаются с помощью функции *factor(x = character(), levels, labels= levels).*

Предположим, что у нас имеется 3 класса *Yes, No, Perhaps*. И некоторая выборка из 6 объектов, каждый из которых принадлежит одному из этих классов.

> v = c("Yes", "No", "Yes", "Perhaps", "No", "Perhaps")

> f = factor(v)

> f

[1] Yes No Yes Perhaps No Perhaps

Levels : No Perhaps Yes

Для получения и изменения текстового вектора, содержащего имена уровней фактора, служит функция *levels(x).*

##### Структуры данных

Структуры данных (data frames) один из самых важных типов данных в R, позволяющий объединять данные разных типов вместе. Структура является специальной версией списка, где все элементы имеют одинаковую длину. Можно считать, что структура данных это двумерная таблица, в которой (в отличие от числовых матриц), разные столбцы могут содержать данные разных типов (но все данные в одном столбце имеют один тип).

Создать структуру данных можно с помощью функции *data.frame(...),* аргументами которой являются произвольное количество элементов (столбцов) фрейма. В качестве элементов структуры данных могут выступать векторы, факторы, матрицы, списки или другие структуры. При этом все векторы должны иметь одинаковую длину, а матрицы и фреймы одинаковое число строк. Функция *data.frame(...)* просто собирает все данные вместе. Символьные векторы конвертируются в факторы. Остальные данные собираются во фрейм такими, какие они есть, поэтому для правильной работы некоторых алгоритмов приходится конвертировать их самостоятельно.

> a = matrix(1:8 , nrow = 4, ncol = 2)

> b = c("a", "b", "c", "a")

> d = (1:4 %% 2 == 0)

> e = factor(c("soft", "hard", "soft", "medium"))

> f = data.frame(a , b , d , e )

С помощью функции *colnames(x)* можно изменить имена столбцов фрейма, с помощью *rownames(x)* – имена строк.

Последовательность команд для самостоятельного рассмотрения:

m<-50

u<-c(1:m)

ncol(u)

nrow(u)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

u<-matrix(c(1:m),1,m)

ncol(u)

nrow(u)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

k<-10

B<-matrix(rep(u,times=k),m,k)

ncol(B)

nrow(B)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

B<-t(matrix(rep(u,times=k),m,k))

ncol(B)

nrow(B)

Для загрузки наборов данных из файла в структуру или таблицу может быть использована функция *read.table(file, header = FALSE, sep = "", ...),* в которой на вход необходимо задать путь к текстовому файлу с данными. Параметр *header* позволяет указать, следует ли интерпретировать первую строку файла, как имена столбцов таблицы.

### 2.4.5 Рабочая директория, сохранение и воспроизводство кодов

Следующее окно содержит все необходимые команды:

getwd() # get current working directory

setwd("c:/contest/MSU/") # set working directory

setwd("c:/project/")

pr1 <- function(n){

source(file.path("hello.R"))

}

ls()

save(pr1, file = "hello.R")

load(file = "hello.R")

### Ресурсы среды R: условия, циклы, функции

Следующий код иллюстрирует операторы цикла *for* и операторы условия *if/else* и реализует алгоритм построения фрактального дерева (Алг. 1)

fr1 <- function(m){

m = 3

a = c(0.0,-0.85,0.2,-0.15)

b = c(0.0,0.04,-0.26,0.28)

c = c(0.0,-0.04,0.23,0.26)

d = c(0.16,0.85,0.22,0.24)

e = c(0.0,0.0,0.0,0.0)

f = c(0.0,0.15,0.15,0.02)

p = c(0.01,0.85,0.07,0.07)

n = 10000

A = matrix(0,n+1,2)

for(i in 1:n){

q = runif(1)

if(q>=0.93){k=4}

else{

if(q>=0.86){k=3}

else{

if(q>=0.01){k=2}

else{k=1}

}

}

A[i+1,1] = a[k]\*A[i,1]+b[k]\*A[i,2]+e[k]

A[i+1,2] = c[k]\*A[i,1]+d[k]\*A[i,2]+f[k]

}

win.graph()

plot(A[,1], A[,2], xlab = "x", ylab = "y", main = "Fractal TREE", type = "p", col = "green", pch = m)

}

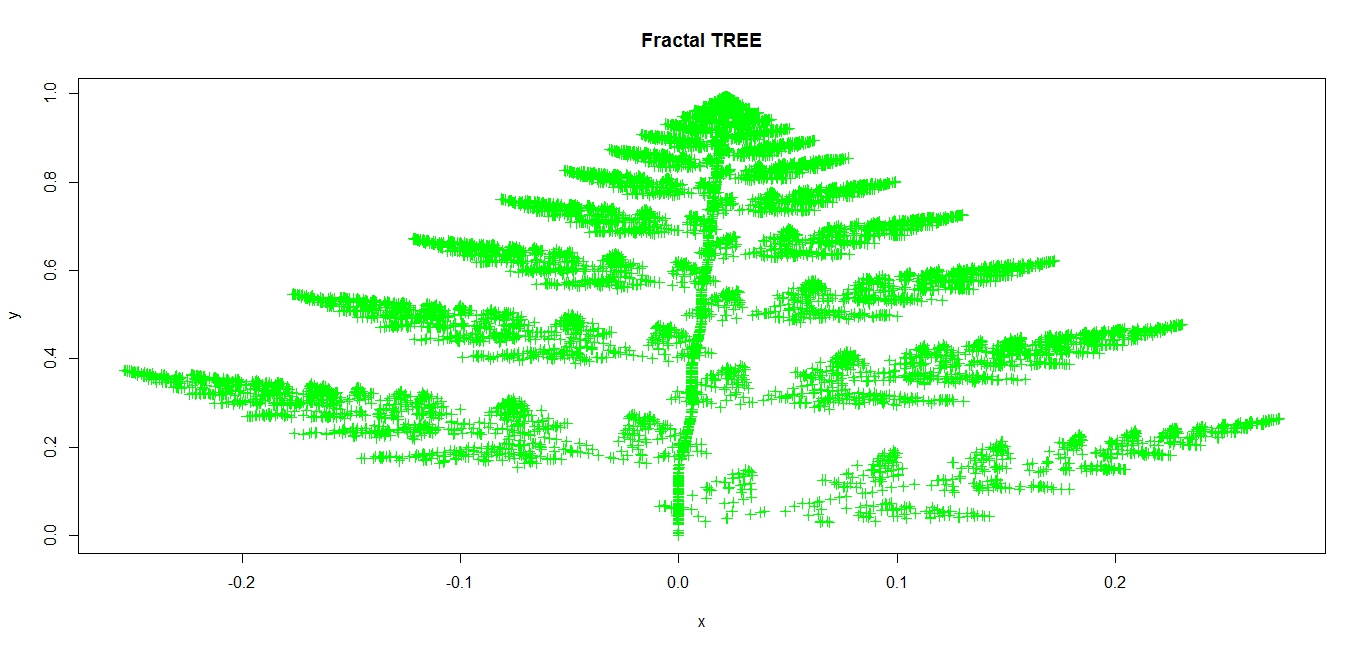


Рис1. Фрактальное дерево (построено при помощи Алг.1)

Кроме того, в приложении приведён алгоритм для вычисления другого фрактала, широко известного как Mandelbrot Set, см. Алг. 9. Следует отметить, что язык R является медленным. По этой причине необходимо использовать встроенные функции для ускорения работы программы. В приложении приведён пример (Алг. 10.), согласно которому вычисление при помощи оператора *which* осуществляется в несколько раз быстрее.

### Работа с индексами

При работе в среде R необходимо учитывать некоторые особенности при оперировании с индексами. Следующий пример иллюстрирует, что второй цикл перегружен операциями сложения и по этой причине выполняется некорректно:

tst <- function(n){

nr = 20

nc = 3

shift = 5

A = matrix(0,nr+shift,nc)

for(i in 1:nr){

A[i,1] = i

}

for(i in 1+shift:nr+shift){

A[i-shift,2] = i-shift

}

print(A)

}

**Задачи для самостоятельной работы:**

1) выполнить приведенный выше код и убедиться, что второй столбец матрицы *A* сформирован некорректно;

2) выполнить Алг. 9 из приложения и рассмотреть отдельные фрагменты фрактала Мандельброт;

3) выполнить Алг. 10 из приложения и убедиться в существующей разнице скорости вычислений.

# Раздел 3. Линейная регрессия

##### 3.1 Парная линейная регрессия.

Линейная регрессия (Linear regression model) является самой простой моделью, используемой в задачах классификации и регрессии [2]. Математическое основание линейной регрессии состоит в том, между объясняющей *X* и объясняемой *Y* переменными устанавливается линейная зависимость. В случае, когда аргумент этой зависимости является единственной переменной *X* , говорят о *парной линейной регрессии*.В англоязычной литературе для этого используется термин *Simple Linear Regression*. Парную линейную регрессию формально мы можем представить, как

(3.1)

В математической статистике для (2.1) используется также термин *регрессия Y на X (*или *Y по X).* В уравнении коэффициенты и являются постоянными и определяются по известной выборке с помощью метода наименьших квадратов. В практике статистического анализа регрессия используется для предсказания статистического признака *Y*. В наших обозначениях предсказанное значение определяется по *X=x*

(3.2)

**3.2 Оценивание параметров регрессии**

Задача линейной регрессии – нахождение параметров или коэффициентов и . Для их определения обычно используется метод наименьших квадратов. Суть его заключается в том, чтобы уравнение прямой линии (3.1) проходило максимально близко к заданным точкам выборки

В соответствие с методом наименьших квадратов сформируем функционал

(3.3)

Решение задачи минимизации функционала (3.3) дает выражения для определения параметров линейной регрессии

, (3.4)

Здесь средние значения, вычисленные по выборке.

Для оценки точности и адекватности модели линейной регрессии (3.2) с коэффициентами (3.4) обычно используются стандартная ошибка остатков *RSE*

(3.5)

Величина (3.5) считается мерой несоответствия модели (3.2) исходным данным. Если величина *RSE* относительно невелика (одного порядка с данными или меньше данных), то можно заключить, что отклонение модельных данных от фактических незначительно и модель (3.2) хорошо описывает данные выборки. В противном случае модель не соответствует данным.

*RSE* является абсолютной величиной меры соответствия с данными и по ней трудно судить о пригодности модели, ведь данные могут быть большими размерными величинами. Для лучшей меры соответствия используют альтернативную величину – коэффициент детерминации *R2*

(3.6)

Где общая сумма квадратов отклонений от среднего значения. *TSS* является мерой общей дисперсии объясняемой переменной *Y* и её можно использовать как меру изменчивости переменной *Y.* Следовательно, величина коэффициента детерминации является относительной и выражает долю количества дисперсии, получающуюся после выполнения предсказания по линейной регрессии (3.2). В качестве меры близости линейной регрессии и исходных данных принимается значение коэффициента детерминации *R2,* принадлежащее интервалу от 0 до1. Если коэффициент детерминации *R2* близок к 1, то приближение данных моделью линейной регрессии считается хорошим, если коэффициент детерминации *R2* ближе к 0, то использование линейной регрессии необоснованно. Можно показать, что коэффициент детерминации *R2* идентичен коэффициенту корреляции в парной линейной регрессии.

**Пример использования функции lm() для парной линейной регрессии**

Рассмотрим базу данных Auto [4]. Воспользуемся функцией lm() для подгонки парной линейной регрессии с mpg в качестве отклика (объясняемой переменной) и horsepower в качестве предиктора (объясняющей переменной).

Подключим пакет ISLR и базу данных Auto:

> library(ISLR)

> attach(Auto)

Воспользуемся функцией lm() для подгонки парной линейной регрессии с mpg в качестве отклика (объясняемой переменной) и horsepower в качестве предиктора (объясняющей переменной).

>lm.fit=lm(mpg~horsepower, data=Auto)

Построим график зависимости отклика от предиктора, используя для этого функцию plot().

> plot(lm.fit)

Получим следующий график:

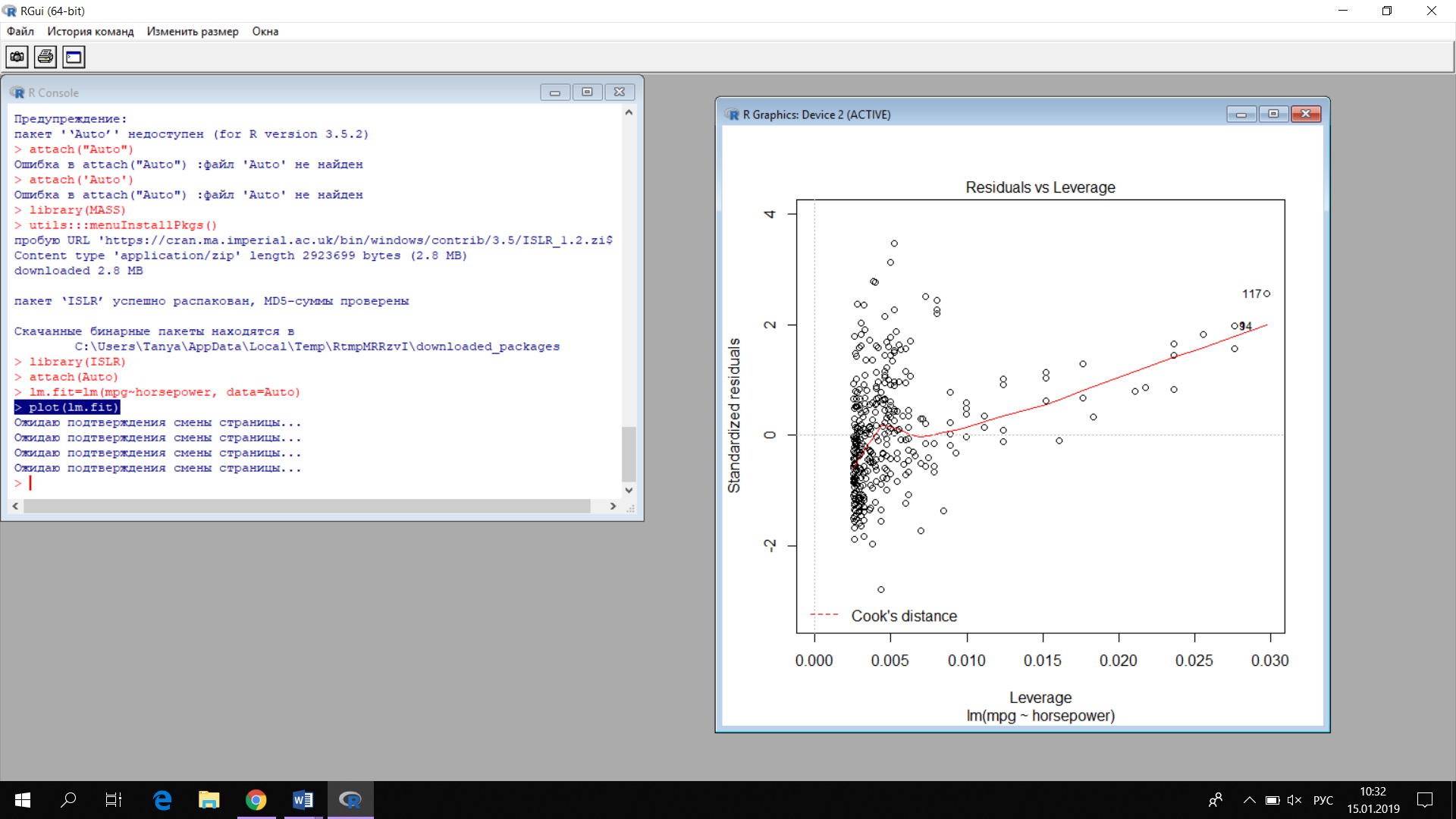


Рис. 2 диаграмма рассеяния mpg~horsepower

**3.3 Множественная линейная регрессия**

На практике выборки могут содержать не одну, а множество объясняющих переменных или факторов, от изменения которых зависит поведение объясняемой переменной. В этом случае эффективной моделью может быть множественная регрессия. Простейшей моделью в этом случае является линейная зависимость в виде функции нескольких переменных вида

(3.7)

Например, в [4] рассматривается предсказания выручки в зависимости от расходов на рекламу на телевидении (TV), радио (radio) и прессе (newspaper). В этом случае уравнение регрессии имеет вид

(3.8)

Проверка качества модели (3.8) производится так же как и в случае парной регрессии, при этом минимизация функционала *RSS*

приводит к решению системы из *n*+1 линейных уравнений.

**3.4 Оценивание параметров множественной регрессии: – критерий**

В рассмотренном выше примере предсказания выручки в зависимости от расходов на рекламу на телевидении (TV), радио (radio) и прессе (newspaper), приведены результаты, сведенные в таблицу. В данной таблице статистика ( – критерий) вычисляется по формуле

и показывает, на сколько стандартных отклонений соответствующий коэффициент отличается от 0 в соответствие с нулевой гипотезой – отсутствие связи между объясняющими переменными *TV, radio, newspaper* и объясняемой переменной *sales.*

Таблица 3.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Переменные | Коэффициенты | Ст. ошибка | статистика | *p* -значение |
| Свободный член | 2.929 | 0.3119 | 9.42 | < 0.0001 |
| TV | 0.046 | 0.0014 | 32.81 | < 0.0001 |
| Radio | 0.189 | 0.0086 | 21.89 | < 0.0001 |
| newspaper | -0.001 | 0.0059 | -0.18 | < 0.8599 |

Таблица 3.1 Коэффициенты множественной регрессии (2.8) предсказания продаж по расходам на рекламу на телевидении, радио и прессе, рассчитанные по методу наименьших квадратов по данным Advertising [4, [www.statlearning.com](http://www.statlearning.com)].

Формально принятие нулевой гипотезы означает равенство нулю всех коэффициентов в модели (3.8). По данным таблицы 3.1 величины фактических значений – критерия свидетельствуют об отклонении (непринятии) нулевой гипотезы, т.к. значения эти относительно велики и превышают критические. Последний столбец содержит значения вероятности того, что ненулевые значения коэффициентов являются случайными. В следующей таблице приведены коэффициенты корреляционной матрицы, которые подтверждают результаты таблицы 3.1.

Таблица 3.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Переменные | sales | TV | Radio | newspaper |
| sales | 1.0000 | 0.7822 | 0.5762 | 0.2283 |
| TV | 0.7822 | 1.0000 | 0.0548 | 0.0567 |
| Radio | 0.5762 | 0.0548 | 1.000 | 0.3541 |
| newspaper | 0.2283 | 0.0567 | 0.3541 | 1.0000 |

Согласно этой таблице, рост выручки (продаж) становится выше от расходов на рекламу на телевидении и радио, в то время как расходы на рекламу в прессе не оказывает влияния на рост продаж, более того, расходы на рекламу в прессе по существу кредитуются из расходов на телевидение и радио.

**Пример использования функции lm() и summary() для множественной линейной регрессии**

Множественная линейная регрессия на примере Auto и переменных cylinders, horsepower, year, acceleration.

|  |
| --- |
| > library (MASS)  > attach (Auto)  > lm.fit = lm (mpg ~ cylinders + horsepower + year + acceleration)  > summary (lm.fit)  Call:  lm(formula = mpg ~ cylinders + horsepower + year + acceleration)  Residuals:  Min 1Q Median 3Q Max  -7.8887 -2.2340 -0.3894 1.9247 14.6041  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) -12.85814 5.12426 -2.509 0.012510 \*  cylinders4 7.04731 1.83086 3.849 0.000139 \*\*\*  cylinders5 5.14055 2.77562 1.852 0.064790 .  cylinders6 0.65058 1.85788 0.350 0.726402  cylinders8 1.28526 1.93745 0.663 0.507489  horsepower -0.10129 0.01134 -8.932 < 2e-16 \*\*\*  year 0.65117 0.05438 11.976 < 2e-16 \*\*\*  acceleration -0.43032 0.09372 -4.592 5.98e-06 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error:3.583 on 383 degrees of freedom  Multiple R-squared: 0.7933,Adjusted R-squared: 0.7896  F-statistic: 210.1 on 7 and 383 DF, p-value: < 2.2e-16  > par (mfrow = c (2, 2))  > plot (lm.fit) |

В результате получим графики остатков

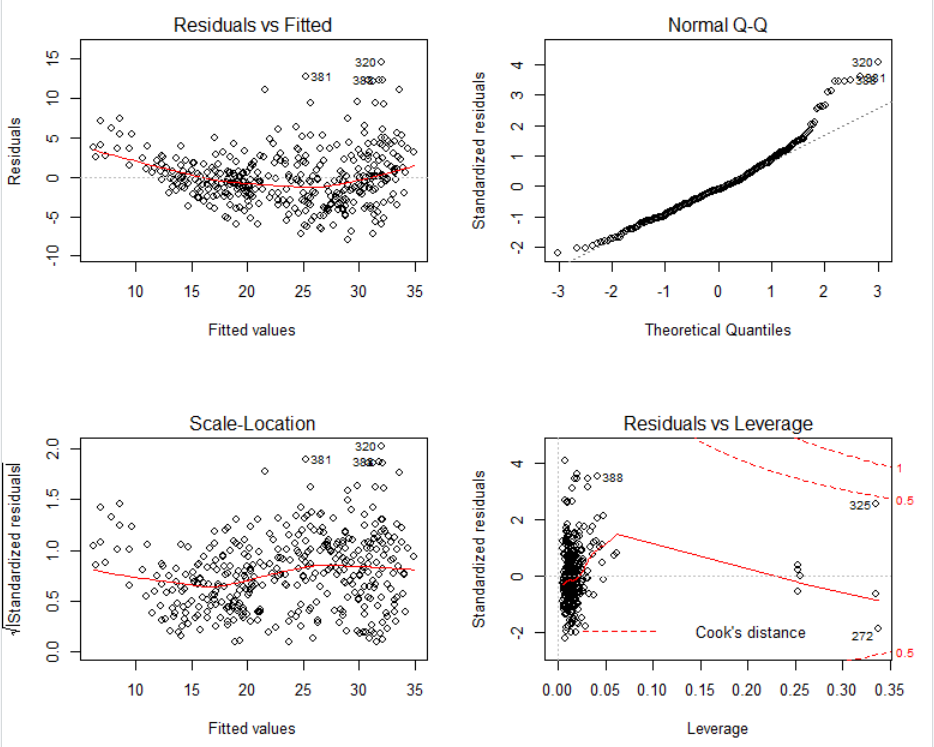


Рис. 4. Диаграммы рассеяния остатков и стандартизованные ошибки

Матрица корреляций для переменных cylinders, horsepower, year, acceleration.

|  |
| --- |
| > names (Auto)  [1] "mpg" "cylinders" "displacement" "horsepower"  [5] "weight" "acceleration" "year" "origin"  [9] "name"  cor (Auto [c (1, 2, 4, 6, 7)])  mpg cylinders horsepower acceleration year  mpg 1.0 -0.7776431 -0.7781647 0.4221878 0.5799602  cyl-0.7776431 1.0 0.8433154 -0.5022810 -0.3415949  hors-0.7781647 0.8433154 1.0 -0.6888569 -0.4152394  acc 0.4221878 -0.5022810 -0.6888569 1.0 0.2865470  year 0.5799602 -0.3415949 -0.4152394 0.2865470 1.0 |

**3.5 Оценивание параметров множественной регрессии: *F*– критерий**

Как и в случае парной линейной регрессии, рассмотрим нулевую гипотезу, но для проверки её воспользуемся *F*– критерием Фишера. Нулевая гипотеза

Альтернативная гипотеза

или по крайней мере, хотя бы один из коэффициентов не равен нулю. Проверка гипотезы выполняется путем вычисления *F*– критерия

(3.9)

Здесь *n, p –* соответственно число элементов в выборке и число объясняющих переменных, *TSS, RSS* определяются по аналогии с парной регрессией. Если нулевая гипотеза выполняется, то значение *F*– критерия будет близким к 1. Расчеты по методу наименьших квадратов с определением критерия Фишера приведены в таблице 3.3

Таблица 3.3

|  |  |
| --- | --- |
| Мера | значение |
| Стандартная ошибка остатков | 1.69 |
| *R2* | 0.897 |
| *F*– критерий | 570 |

Данные таблицы свидетельствуют об отклонении нулевой гипотезы, что подтверждает выводы предыдущего раздела.

**3.6 Проблема отбора переменных**

Выбор информативных переменных в предполагаемых моделях множественной регрессии может быть осуществлена многими способами. Для суждения о наилучшей модели по числу используемых объясняющих переменных в многомерной статистике используют разные критерии: – критерий Мэллоу, информационный критерий Акаике (AIC), байесовский информационный критерий (BIC), коэффициент детерминации *R2.* Применение этих критериев в идеале представляет перебор всех моделей с проверкой соответствующих критериев. Однако в реальности это приводит к «проклятию размерности». Допустим, что мы отобрали *р* объясняющих переменных, соответственно нам придется перебрать 2n моделей. Ясно, что это можно реализовать только в случае небольших значений *р.* Необходимо использовать другой эффективный способ отбора нужной модели. Для решения этой проблемы используют три подхода:

* *Отбор с включением.* В начале рассматривается нулевая модель, включающая только свободный член. Далее прогоняют *р* парных (простых) моделей и из них выбирают ту переменную, добавляя её к нулевой модели, при которой из всех парных моделей**, у** которой наблюдаетсянаименьшее значение*RSS.* Затем продолжая прогоны с оставшимися *р-1* моделями парной регрессии, вновь находят переменную с наименьшим значением*RSS* и добавляют её к модели с двумя объясняющими переменными и т.д.
* *Отбор с исключением.*В этом случае начинают рассматривать модель множественной регрессии со всеми *р* переменными и исключают ту переменную, у которой наблюдается наибольшее *р-* значение. Исключив её, то же самое производится с оставшейся *р-1* моделью и т.д.
* *Комбинированный отбор.*В этом методе используютоба сформулированных правила. Начиная с нулевой модели, добавляют в соответствии с первым правилом новые переменные, при этом следят за соответствующими вероятностными оценками, т.е. *р-* значениями новых и старых переменных, исключая те, у которых *р-* значение превышает некоторый, установленный заранее порог.

Отбор по правилу 2 неприменим при условии *p>n,* в то время как правило 1 применимо во всех случаях.

**3.7 Использование множественной линейной регрессии для**

**качественных переменных**

Качественными переменными называют факторы, которые не имеют количественных значений, например, статус больного (начальная стадия заболевания, острая стадия, хроническая стадия и т.п.). Часто приходится иметь дело с выборками, в которых присутствуют как количественные переменные, так и качественные. В книге [4] рассматривается задача прогнозирования платежеспособности клиентов банка Default (база данных Credit на платформе *Kaggle*[[3]](#footnote-3) ). Данные “Credit” использовались при проведении соревнования “Give me some credit” на платформе *Kaggle*. В рамках этого соревнования было необходимо определить кредитоспособность клиента в ближайшие два года. Данные предоставляются на 250000 клиентов и имеют следующую структуру.

Таблица 3.4. Структура данных “Credit”

|  |  |  |
| --- | --- | --- |
| **Имя переменной** | **Описание** | **Тип** |
| SeriousDlqin2yrs (целевая переменная) | Имеет ли клиент 90 или более дней просрочки по выплатам | Да/Нет |
| RevolvingUtilizationOfUnsecuredLines | Общий остаток по кредитным картам за исключением недвижимого имущества и автокредитования, делённый на сумму кредитных лимитов | percentage |
| Age | Возраст клиента в годах | integer |
| NumberOfTime30-59DaysPastDueNotWorse | Количество раз, когда клиент просрочивал на 30-59 дней в последние 2 года | integer |
| DebtRatio | Месячные выплаты долгов, алиментов, прожиточного минимума делённые на месячный валовый доход | percentage |
| MonthlyIncome | Месячный доход | integer |
| NumberOfOpenCreditLinesAndLoans | Число открытых кредитов (рассрочка на покупку автомобиля или ипотека) и кредитные линии (например, кредитные карты) | integer |
| NumberOfTimes90DaysLate | Количество раз, когда клиент просрочивал выплаты на 90 дней и более | integer |
| NumberRealEstateLoansOrLines | Количество ипотечных кредитов и кредитов на объекты недвижимости, в том числе собственный капитал кредитных линий | integer |
| NumberOfTime60-89DaysPastDueNotWorse | Количество раз, когда клиент просрочивал на 60-89 дней в последние 2 года | integer |
| NumberOfDependents | Количество иждивенцев в семье (супруг, дети и т.д.) | integer |

Все данные (250000 наблюдений) разделены на 2 части: 150000 тысяч для тренировки и 100000 для тестирования. В данном курсе первый набор разбит десять раз случайным образом на три части (триплет):

* Тренировочный набор – значения целевой переменной известны. Необходим для обучения модели.
* Валидационный набор – значения целевой также переменной известны. Необходим для предварительной проверки качества модели.
* Тестирующий набор – значения целевой переменной необходимо предсказать, используя полученную ранее модель.

Прежде всего, необходимо загрузить данные в среду R:

A <- read.table("F:/kaggle/vrs10/train\_vrs10.txt", header=TRUE)

V <- read.table("F:/kaggle/vrs10/valid\_vrs10.txt", header=TRUE)

T <- read.table("F:/kaggle/vrs10/test\_vrs10.txt", header=TRUE)

В R она входит в базовый пакет и не требует дополнительных загрузок.

Построить модель можно следующим образом:

*object <- lm(tg~.,data=A)*

*tg~.* – формула, показывающая, что ищется зависимость целевой переменной (или метки) от всех имеющихся признаков. Отметим, что формулу можно модифицировать, дополнив конкретными переменными. Например – *tg ~ v1+v*

*data = A* – выбор набора данных для тренировки модели.

После выполнения кода можно посмотреть подробную информацию о полученной модели: summary*(object).*

Основным элементом *object* является вектор регрессионных коэффициентов. Используя эти коэффициенты, можно вычислить решающую функцию (1):

 (3.10)

Решение о положительности клиента принимается согласно следующему правилу:

 (3.11)

где Δ – параметр усечения.

В данном случае выражение (2.11) представляет простейшее представление качественной переменной, выраженное в двоичных (булевских) переменных. В книге [4] для них используется также термин *индикаторные переменные*.

**3.8 Проблемы, возникающие при использовании линейных**

**регрессионных моделей**

При использовании линейных регрессионных моделей возникает ряд проблем. Основные из них:

1. Нелинейность связей между объясняемыми и объясняемыми переменными;
2. Корреляция остатков (автокорреляция);
3. Непостоянная дисперсия остатков;
4. Выбросы;
5. Коллинеарность.

Эти проблемы хорошо изучены в курсах многомерной статистики и эконометрики. Рассмотрим их по отдельности.

*Нелинейность.* Линейная регрессионная модель представляет упрощенное моделирование связей между переменными, являясь, по сути, первым приближением в отражении реальных связей. Соответственно, более точные модели связей будут заведомо нелинейными Удобным инструментом обнаружения нелинейности является геометрическая интерпретация выборок. Для этой цели используют корреляционное поле изучаемой выборки, а также графики остатков. В случаях множественной регрессии построить корреляционное поле не представляется возможным из-за многомерности пространства. Графики остатков вычисляются по формуле

Для каждой объясняемой переменной *xi*. Графики остатков имеют много общего с корреляционным полем. Если линейная модель адекватно описывает связи, то значения остатков группируются в окрестности нулевой линии, в противном случае линейная регрессионная модель не является адекватной и необходимо искать возможные нелинейные зависимости.

*Корреляция остатков.* Важным допущением моделей множественной (в том числе парной) линейной регрессии является предположение, что остатки являются взаимно-независимыми, т.е некоррелированными. В идеальном случае они распределены по нормальному закону с математическим ожиданием, равным нулю. Проблема коррелируемости остатков часто возникает при исследовании временных рядов. Там это явление называют автокорреляцией, причина которой связана с нестационарностью и взаимозависимостью членов временного ряда. Строго говоря, для таких временных рядов не выполняются условия Гаусса и применять, в частности, для нахождения коэффициентов линейной зависимости метод наименьших квадратов нельзя. Если коррелированность остатков проявляется заметно, то необходимо отказываться от линейной регрессии.

*Непостоянная дисперсия остатков.* Эта проблема связана с предыдущей и проистекает из нарушения условий Гаусса. В теории временных рядов это явление называют гетероскедастичностью. В этих случаях также необходимо отказываться от линейной модели и переходить с помощью преобразования переменных к квазилинейной модели. Следует заметить, что это явление также обнаруживается в графическом представлении ряда остатков.

*Выбросы.* Выбросом называют такие наблюдения, значения которых резко отличаются от соседних или других наблюдений. Природа выбросов очевидна – выборки не всегда являются репрезентативными и получение их сопряжено с неточностью наблюдений, наличием неконтролируемых факторов. Выбросы также хорошо наблюдаемы на графиках корреляционного поля и графиках остатков. Простейшим способом избавиться от выбросов является сглаживание данных, однако надо всегда помнить, что это может быть неслучайным событием и возможно является следствием неучтенных факторов.

*Коллинеарность.* Под коллинеарностью понимают ситуацию, когда две или более объясняющих переменных оказываются зависимыми друг от друга (чаще всего линейно-зависимыми). Наличие коллинеарности можно сравнить с ситуацией, когда мы решаем систему линейных уравнений с линейно-зависимыми строками (столбцами) в матрице коэффициентов. Определитель такой матрицы равен нулю и система в таком случае не имеет единственного решения. На практике чаще сталкиваются с так называемыми «плохо обусловленными матрицами» определитель которых близок к нулю. Во всех этих случаях использование метода наименьших квадратов приводит к большим искажениям решений при поиске коэффициентов регрессии. Простой способ обнаружения коллинеарности состоит в исследовании корреляционной матрицы. Если коррелируют две взаимно-связанные переменные, то их легко обнаружить по большим (по абсолютной величине) значениям элементов корреляционной матрицы. Труднее это сделать, когда коррелируют три и более переменных. Такое явление называют мультиколлинеарностью. При столкновении с проблемой коллинеарности самый простой способ от неё избавиться состоит в том, чтобы удалить проблематичные переменные из регрессии. Второй способ состоит в том, чтобы объединить коррелированные переменные в один комплекс, например, взять вместо них среднее значение коррелированных переменных, если это возможно.

**3.9 Сравнение линейной регрессии с методом ближайших соседей (*KNN)***

Метод множественной линейной регрессии относится к классу параметрических методов, в которых заранее определяется вид аппроксимирующей функции. В случае линейной регрессии это подбор параметров, являющихся коэффициентами линейной зависимости. Это легко выполнимая операция как в случае подбора коэффициентов, так и в случае проверки качества модели с помощью определенных критериев. Однако все параметрические методы обладают одним недостатком: в силу определенности выбора аппроксимирующей функции мы не можем её изменить. Если выбранная функциональная зависимость не соответствует реальной выборке, то нет никакой возможности улучшить предсказание в рамках выбранной модели.

В отличие от параметрических методов непараметрические методы не связаны однозначностью предположений относительно функциональной связи и, следовательно, предоставляют альтернативный и более гибкий подход для выполнения регрессионного анализа. Наиболее простым способом применения непараметрических методов является использование метода ближайших соседей *KNN (K – nearest neighbors)*. Процедура использования KNN – регрессии заключается в следующем:

1. Для некоторого заданного значения K и прогнозируемого наблюдения *х0* KNN – регрессия сначала находит К обучающих наблюдений (обозначаются как *N0*), которые ближе всего располагаются к *х0*
2. Выбрав эти наблюдения, KNN – регрессия оценивает , усредняя значения отклика всех обучающих наблюдений из множества *N0*:

Как можно видеть из формулы для аппроксимирующей функции , она представляет собой кусочно-непрерывную ступенчатую функцию, при этом увеличения значения К приводит её к более сглаженной функции.

Очевидно, что применение параметрических методов оправдано и дает лучшие результаты при прогнозировании, если выбранная функциональная зависимость оказывается близкой к реальной выборке. В остальных случаях следует предпочесть выбор метода ближайших соседей *KNN,* во всяком случае для оценки предполагаемой нелинейной зависимости. С другой стороны, в пользу использования параметрических методов, в частности линейной регрессии, говорит тот факт, что параметрические методы при обоснованном выборе аппроксимирующей функции реализуются алгоритмически удобнее и проще.

* 1. **Практическая реализация методов линейной регрессии**

**в пакете R**

* + 1. **Парная линейная регрессия**

Функция library() используется для загрузки библиотек или групп функций и наборов данных, которые не включены в базовый дистрибутив пакета R. Обычные функции, выполняющие расчет регрессии по методу наименьших квадратов и другие процедуры, в том числе проверки качества модели, входят в состав стандартного дистрибутива R. Для использования наборов данных (баз данных для обработки) в данном случае используются базы данных MASS и пакет ISLR из книги [4]:

|  |
| --- |
| > library (MASS)  > library (ISLR) |

Применение метода простой (парной) линейной регрессии показывается на примере базы данных Boston, которая включает медианные данные medv по стоимости дома для 506 расположений (окрестностей) города Бостон. В предложенной модели делается предсказание на основе 13 предикторов (факторов): “crim”, “zn”, “indus”, “chas”, “nox:, “rm”, “age”, “dis”, ”rad” , “tax”, “ptratio”, “black”, “lstat”.

|  |
| --- |
| > fix(Boston )  > names(Boston )  [1] "crim" "zn" "indus" "chas" "nox" "rm" "age"  [8] "dis" "rad" "tax" "ptratio " "black" "lstat" "medv" |

Функция lm() служит для подгонки простой линейной регрессии, где medv будет откликом (объясняемой переменной), а lstat – предиктором (объясняющей переменной). Основной синтаксис выглядит как lm(y*\_*x,data) , где y это зависимая переменная, а х – предиктор, data – набор данных, в котором содержатся эти переменные.

|  |
| --- |
| > lm.fit =lm(medv*\_*lstat)  Error in eval(expr , envir , enclos ) : Object "medv" not found |

При вызове команды появляется сообщение об ошибке, так как в алгоритме не прописан адрес для переменных medv и lstat. Следующая команда определяет их размещение и вызов

|  |
| --- |
| > lm.fit =lm(medv*\_*lstat ,data=Boston )  > attach (Boston )  > lm.fit =lm(medv*\_*lstat) |

При вводе оператора lm.fit будет выведена основная информация по модели.для получения более полной информации можно воспользоваться оператором summary (lm.fit)/ это даст возможность получить оценки р-значений и стандартные ошибки коэффициентов, а также коэффициент детерминации R2 и F- критерий Фишера для модели.

|  |
| --- |
| > lm.fit  Call:  lm(formula = medv *\_* lstat)  Coefficients:  (Intercept ) lstat  34.55 -0.95  > summary (lm.fit)  Call:  lm(formula = medv *\_* lstat)  Residuals :  Min 1Q Median 3Q Max  -15.17 -3.99 -1.32 2.03 24.50  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept ) 34.5538 0.5626 61.4 <2e-16 \*\*\*  lstat -0.9500 0.0387 -24.5 <2e-16 \*\*\*  ---  Signif . codes: 0 \*\*\* 0.001 \*\* 0.01 \* 0.05 . 0.1 1  Residual standard error : 6.22 on 504 degrees of freedom  Multiple R-squared : 0.544 , Adjusted R-squared : 0.543  F-statistic : 602 on 1 and 504 DF , p-value: <2e-16 |

Можно воспользоваться функцией names(), чтобы выяснить, что еще хранится в объекте lm.fit

|  |
| --- |
| > names(lm.fit )  [1] " coefficients" "residuals " "effects "  [4] "rank" "fitted .values " "assign "  [7] "qr" "df.residual " "xlevels "  [10] "call" "terms" "model"  > coef(lm.fit)  (Intercept ) lstat  34.55 -0.95 |

Для вычисления доверительных интервалов оценок коэффициентов можно использовать команду confint()

|  |
| --- |
| > confint (lm.fit)  2.5 % 97.5 %  (Intercept ) 33.45 35.659  lstat -1.03 -0.874 |

Функцию predict() можно использовать для вычисления доверительных интервалов и диапазона предсказаний при прогнозировании величины medv по заданному значению предиктора lstat

|  |
| --- |
| > predict (lm.fit ,data.frame(lstat=c(5 ,10 ,15) ),  interval =" confidence ")  fit lwr upr  1 29.80 29.01 30.60  2 25.05 24.47 25.63  3 20.30 19.73 20.87  > predict (lm.fit ,data.frame(lstat=c(5 ,10 ,15) ),  interval =" prediction ")  fit lwr upr  1 29.80 17.566 42.04  2 25.05 12.828 37.28  3 20.30 8.078 32.53 |

Визуализация результатов производится с помощью операторов plot() и abline()

|  |
| --- |
| > plot(lstat, medv)  > abline (lm.fit) |

Использование дополнительных функций при визуализации можно видеть в следующем фрагменте кода

|  |
| --- |
| > abline (lm.fit ,lwd =3)  > abline (lm.fit ,lwd =3, col ="red ")  > plot(lstat ,medv ,col ="red ")  > plot(lstat ,medv ,pch =20)  > plot(lstat ,medv ,pch ="+")  > plot (1:20 ,1:20, pch =1:20) |

Здесь используются команды lwd=3, которая увеличивает толщину линии регрессии в 3 раза, команда col определяет цвет линии, команда pch определяет стиль и различные символы для отображения точек. Другие сервисы при визуализации результатов приводятся ниже

|  |
| --- |
| > par(mfrow =c(2,2))  > plot(lm.fit) |

Команда par(mfrow =c(2,2)) разделяет графическое окно на сетку из 2×2 панелей

Функция rstudent() выдает остатки, распределенные по закону Стьюдента

|  |
| --- |
| > plot(predict (lm.fit), residuals (lm.fit))  > plot(predict (lm.fit), rstudent (lm.fit)) |

Значения показателя разбалансировки можно получить с помощью команды hatvalues()

|  |
| --- |
| > plot(hatvalues (lm.fit ))  > which.max (hatvalues (lm.fit)) |

Функция which.max() определяет, какой индексный номер имеет наибольший элемент некоторого вектора.

* + 1. **Множественная линейная регрессия**

Здесь также используется оператор lm() . Синтаксис кода при использовании 3-х предикторов имеет вид

|  |
| --- |
| > lm.fit =lm(medv*∼*lstat+age ,data=Boston )  > summary (lm.fit)  Call:  lm(formula = medv *∼* lstat + age , data = Boston )  Residuals :  Min 1Q Median 3Q Max  -15.98 -3.98 -1.28 1.97 23.16  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept ) 33.2228 0.7308 45.46 <2e-16 \*\*\*  Lstat -1.0321 0.0482 -21.42 <2e-16 \*\*\*  Age 0.0345 0.0122 2.83 0.0049 \*\*  ---  Signif . codes: 0 \*\*\* 0.001 \*\* 0.01 \* 0.05 . 0.1 1  Residual standard error : 6.17 on 503 degrees of freedom  Multiple R-squared : 0.551 , Adjusted R-squared : 0.549  F-statistic : 309 on 2 and 503 DF , p-value: <2e-16 |

Набор данных Boston содержит 13 переменных для сокращения операции перечисления всех переменных используется прием сокращенного варианта ввода

|  |
| --- |
| > lm.fit =lm(medv*∼*.,data=Boston )  > summary (lm.fit)  Call:  lm(formula = medv *∼* ., data = Boston )  Residuals :  Min 1Q Median 3Q Max  -15.594 -2.730 -0.518 1.777 26.199  Coefficients:  Estimate Std . Error t value Pr(>|t|)  (Intercept ) 3.646e+01 5.103 e+00 7.144 3.28e -12 \*\*\*  crim -1.080 e-01 3.286e-02 -3.287 0.001087 \*\*  zn 4.642e-02 1.373e-02 3.382 0.000778 \*\*\*  indus 2.056e-02 6.150e-02 0.334 0.738288  chas 2.687e+00 8.616e-01 3.118 0.001925 \*\*  nox -1.777 e+01 3.820 e+00 -4.651 4.25e -06 \*\*\*  rm 3.810e+00 4.179e-01 9.116 < 2e -16 \*\*\*  age 6.922e-04 1.321e-02 0.052 0.958229  dis -1.476 e+00 1.995e-01 -7.398 6.01e -13 \*\*\*  rad 3.060e-01 6.635e-02 4.613 5.07e -06 \*\*\*  tax -1.233 e-02 3.761e-03 -3.280 0.001112 \*\*  ptratio -9.527 e-01 1.308e-01 -7.283 1.31e -12 \*\*\*  black 9.312e-03 2.686e-03 3.467 0.000573 \*\*\*  lstat -5.248 e-01 5.072e-02 -10.347 < 2e -16 \*\*\*  ---  Signif . codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error : 4.745 on 492 degrees of freedom  Multiple R-Squared : 0.7406 , Adjusted R-squared : 0.7338  F-statistic : 108.1 on 13 and 492 DF , p-value: < 2.2e -16 |

Можно получить доступ к отдельным компонентам объекта с результатами анализа по их имени (для этого нужно набрать команду summary.lm –получим список доступных компонентов). Так команда summary(lm.fit)$r.sq дает значение R2, команда summary(lm.fit)$sigma даст значение RSE (среднее квадратическое отклонение). Функция vif() из пакета car применяется для вычисления значений фактора инфляции дисперсии. Пакет car необходимо подключить с помощью команды install.packfges().

|  |
| --- |
| > library (car)  > vif(lm.fit)  crim zn indus chas nox rm age  1.79 2.30 3.99 1.07 4.39 1.93 3.10  dis rad tax ptratio black lstat  3.96 7.48 9.01 1.80 1.35 2.94 |

Если обратить внимание на тот факт, что р-значение при переменной age имеет высокое значение, то это может означать, что эту переменную можно исключить. Для этого можно воспользоваться кодом

|  |
| --- |
| > lm.fit1=lm(medv*∼*.-age ,data=Boston )  > summary (lm.fit1)  ... |

Альтернативным вариантом может быть использование команды update()

|  |
| --- |
| > lm.fit1=update (lm.fit , *∼*.-age) |

* + 1. **Эффекты взаимодействия**

Включение эффектов взаимодействия в линейную модель может быть выполнен при помощи команд типа lstat: black что означает добавление в анализ взаимодействия между предикторами lstat и black. Синтаксис lstat\*age включает взаимодействие между предикторами lstat, age и взаимодействие lstat×age.

|  |
| --- |
| > summary (lm(medv*∼*lstat \*age ,data=Boston ))  Call:  lm(formula = medv *∼* lstat \* age , data = Boston )  Residuals :  Min 1Q Median 3Q Max  -15.81 -4.04 -1.33 2.08 27.55  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept ) 36.088536 1.469835 24.55 < 2e-16 \*\*\*  lstat -1.392117 0.167456 -8.31 8.8e-16 \*\*\*  age -0.000721 0.019879 -0.04 0.971  lstat:age 0.004156 0.001852 2.24 0.025 \*  ---  Signif . codes: 0 ’\*\*\*’ 0.001 ’\*\*’ 0.01 ’\*’ 0.05 ’.’ 0.1 ’ ’ 1  Residual standard error : 6.15 on 502 degrees of freedom  Multiple R-squared : 0.556 , Adjusted R-squared : 0.553 |

* + 1. **Нелинейные преобразования предикторов**

Функция lm() может принимать нелинейно преобразованные переменные . Например, имея предиктор Х, мы можем создать предиктор Х2 с помощью операции I(X^2). Теперь есть возможность построить регрессию по Х и по Х2, принимая в качестве предикторов lstat и lstat2

|  |
| --- |
| > lm.fit2=lm(medv*∼*lstat +I(lstat ^2))  > summary (lm.fit2)  Call:  lm(formula = medv *∼* lstat + I(lstat ^2))  Residuals :  Min 1Q Median 3Q Max  -15.28 -3.83 -0.53 2.31 25.41  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept ) 42.86201 0.87208 49.1 <2e-16 \*\*\*  lstat -2.33282 0.12380 -18.8 <2e-16 \*\*\*  I(lstat ^2) 0.04355 0.00375 11.6 <2e-16 \*\*\*  ---  Signif . codes: 0 ’\*\*\*’ 0.001 ’\*\*’ 0.01 ’\*’ 0.05 ’.’ 0.1 ’ ’ 1  Residual standard error : 5.52 on 503 degrees of freedom  Multiple R-squared : 0.641 , Adjusted R-squared : 0.639  F-statistic : 449 on 2 and 503 DF , p-value: <2e-16 |

Близкое к нулю р-значение при переменной lstat2 дает основание к включению в модель, которая становится от этого более качественной.

Можно воспользоваться функцией anova() для дальнейшего количественного описания степени преимущества квадратичной модели (в дальнейшем линейную модель обозначим Model1, а квадратичную – Model2).

|  |
| --- |
| > lm.fit =lm(medv*∼*lstat)  > anova(lm.fit ,lm.fit2)  Analysis of Variance Table  Model 1: medv *∼* lstat  Model 2: medv *∼* lstat + I(lstat ^2)  Res.Df RSS Df Sum of Sq F Pr(>F)  1 504 19472  2 503 15347 1 4125 135 <2e -16 \*\*\*  ---  Signif . codes: 0 ’\*\*\*’ 0.001 ’\*\*’ 0.01 ’\*’ 0.05 ’.’ 0.1 ’ ’ 1 |

Функция anova() выполняет проверку определенной гипотезы относительно двух моделей. Нулевая гипотеза состоит в том, что обе модели описывают данные одинаково, альтернативная гипотеза – в том, что Model2 описывает данные лучше, т.е. не совпадает с Model1. F- критерий при этом равен 135, а р-значение практически равно нулю. Это неудивительно, т.к. и ранее обнаруживалась нелинейная зависимость в парной регрессии между medv и lstat. Если выполнить команду

|  |
| --- |
| > par(mfrow=c(2,2))  > plot(lm.fit2) |

То можно увидеть, что при добавлении предиктора lstat2 какая-либо различимая закономерность в расположении остатков исчезает. Можно увеличить степень нелинейности модели, добавив в неё предиктор lstat3 – кубический член в объясняющие переменные. Причем это можно сделать проще. Воспользовавшись при вызове lm() функцией poly(). Например, при добавлении членов полиноминальной зависимости вплоть до 5-й степени

|  |
| --- |
| > lm.fit5=lm(medv*∼*poly(lstat ,5))  > summary (lm.fit5)  Call:  lm(formula = medv *∼* poly(lstat , 5))  Residuals :  Min 1Q Median 3Q Max  -13.543 -3.104 -0.705 2.084 27.115  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept ) 22.533 0.232 97.20 < 2e-16 \*\*\*  poly(lstat , 5)1 -152.460 5.215 -29.24 < 2e-16 \*\*\*  poly(lstat , 5)2 64.227 5.215 12.32 < 2e-16 \*\*\*  poly(lstat , 5)3 -27.051 5.215 -5.19 3.1e-07 \*\*\*  poly(lstat , 5)4 25.452 5.215 4.88 1.4e-06 \*\*\*  poly(lstat , 5)5 -19.252 5.215 -3.69 0.00025 \*\*\*  ---  Signif . codes: 0 ’\*\*\*’ 0.001 ’\*\*’ 0.01 ’\*’ 0.05 ’.’ 0.1 ’ ’ 1  Residual standard error : 5.21 on 500 degrees of freedom  Multiple R-squared : 0.682 , Adjusted R-squared : 0.679  F-statistic : 214 on 5 and 500 DF , p-value: <2e-16 |

Полученный результат свидетельствует, что включение в число объясняющих переменных дополнительных полиноминальных членов вплоть до 5-й степени приводит к улучшению модели.

* + 1. **Качественные переменные**

Использование качественных переменных продемонстрируем на примере данных Carseats из библиотеки ISLR [4]. Рассмотрим задачу предсказания продаж Sales детских автомобильных сидений для 400 региональных мест продажи на основе следующего ряда из 11 предикторов

|  |
| --- |
| > fix( Carseats )  > names(Carseats )  [1] "Sales " "CompPrice " "Income " "Advertising "  [5] " Population " "Price" "ShelveLoc " "Age"  [9] " Education " "Urban" "US" |

Переменные в этом перечне имеют качественный смысл, так например, предиктор ShelveLoc (расположение стеллажа в магазине, где продаются сидения) принимает три возможных значения: Bad, Medium, Good. Встречая подобные переменные, R автоматически создает индикаторные переменные. Ниже приводится пример множественной регрессии с качественными переменными

|  |
| --- |
| > lm.fit =lm(Sales*∼*.+ Income :Advertising +Price :Age ,data=Carseats )  > summary (lm.fit)  Call:  lm(formula = Sales *∼* . + Income : Advertising + Price:Age , data =  Carseats )  Residuals :  Min 1Q Median 3Q Max  -2.921 -0.750 0.018 0.675 3.341  Coefficients:  Estimate Std . Error t value Pr(>|t|)  (Intercept ) 6.575565 1.008747 6.52 2.2e -10 \*\*\*  CompPrice 0.092937 0.004118 22.57 < 2e -16 \*\*\*  Income 0.010894 0.002604 4.18 3.6e -05 \*\*\*  Advertising 0.070246 0.022609 3.11 0.00203 \*\*  Population 0.000159 0.000368 0.43 0.66533  Price -0.100806 0.007440 -13.55 < 2e -16 \*\*\*  ShelveLocGood 4.848676 0.152838 31.72 < 2e -16 \*\*\*  ShelveLocMedium 1.953262 0.125768 15.53 < 2e -16 \*\*\*  Age -0.057947 0.015951 -3.63 0.00032 \*\*\*  Education -0.020852 0.019613 -1.06 0.28836  UrbanYes 0.140160 0.112402 1.25 0.21317  USYes -0.157557 0.148923 -1.06 0.29073  Income :Advertising 0.000751 0.000278 2.70 0.00729 \*\*  Price:Age 0.000107 0.000133 0.80 0.42381  ---  Signif . codes: 0 ’\*\*\*’ 0.001 ’\*\*’ 0.01 ’\*’ 0.05 ’.’ 0.1 ’ ’ 1  Residual standard error : 1.01 on 386 degrees of freedom  Multiple R-squared : 0.876 , Adjusted R-squared : 0.872  F-statistic : 210 on 13 and 386 DF, p-value : <2e-16 |

Функция contrasts() возвращает матрицу, которая показывает, как именно R кодирует индикаторные переменные

|  |
| --- |
| > attach (Carseats )  > contrasts (ShelveLoc )  Good Medium  Bad 0 0  Good 1 0  Medium 0 1 |

Программа R создала индикаторную переменную ShelveLocGood, которая принимает значение 1, если расположение стеллажа хорошее и 0 в остальных случаях и переменную ShelveLocMedium, которая равна 1, если качество расположения стеллажа среднее и 0 в остальных случаях.

1. **Методы классификации**

**4.1 Общее представление о классификации**

Проблемы классификации встречаются часто. Вот некоторые из них:

1. Человек поступает в отделение экстренной медицинской помощи с набором симптомов, которые могут быть связаны с одним из трех заболеваний. Какое из трех заболеваний имеет место быть?
2. Сервис банковского обслуживания через Интернет должен иметь возможность определить, является ли та или иная онлайн-транзакция мошеннической, учитывая IP адрес пользователя, историю предыдущих операций и т.д.
3. На основе данных по структуре ДНК у ряда пациентов, имеющих определенное заболевание, и у пациентов, не подверженных рискам получить это заболевание, биолог хотел бы выяснить, какие мутации ДНК вредны (приводят к этим заболеваниям), а какие – нет.

Подобно тому как это было с регрессией, у нас имеется набор обучающих наблюдений , который мы можем использовать для построения классификатора. При этом следует отметить, что линейная регрессия в большинстве случаев плохо подходит для работы с качественными переменными. Обратимся к первому примеру из перечисленных проблем классификации. Итак, пациент имеет признаки заболевания, которые диагностируются как инсульт, передозировка наркотиками и эпилептический припадок. Их можно закодировать в виде количественных переменных следующим образом:

Применив такое кодирование, можно в дальнейшем воспользоваться методом наименьших квадратов и построить регрессионную модель, на основе которой осуществить предсказание об актуальном диагнозе. Но ведь количественные переменные предполагают порядок сравнения, т.е. 2 больше 1, а 3 больше 2. С другой стороны, почему передозировка больше, чем инсульт? Вообще говоря, можно было по-другому индексировать диагнозы и тогда смысл количественных переменных был бы другим. Если бы количественные переменные хотя бы качественно соответствовали признакам сравнения, например, мягкий, умеренный, твердый, тогда их градация была бы естественной. Проще выглядит ситуация с бинарными переменными типа «да» или «нет». Тогда обоснованно применение так называемых двоичных переменных (булевских) 1, если «да», 0, если «нет». В этом случае возможно использование регрессии. Для более общих ситуаций необходимо использовать другие подходы. Ниже рассмотрим их подробнее.

**4.2 Логистическая регрессия**

Существуют методы, которые являются обобщением линейных регрессионных моделей, особенно в тех случаях, когда методы регрессий неэффективны. К обобщенным линейным моделям относится множество распространенных методов анализа данных. В этом разделе мы кратко рас-

смотрим некоторые теоретические аспекты, лежащие в основе этих методов. Допустим, вы хотите смоделировать связь между зависимой переменной *Y* и набором из *p* независимых переменных *X*1… *Xp*. В стандартной линейной модели вы предполагали, что *Y* имеет нормальное распределение и тип связи описывается таким уравнением:

(4.1)

Это значит, что условное среднее зависимой переменной представляет собой линейную комбинацию значений независимых переменных. Параметры, определяющие ожидаемое изменение переменной *Y* на единицу изменения *Xj*, обозначены как β*j*, а β0 – это ожидаемое значение переменной *Y*, когда все независимые переменные равны нулю. Вы можете предсказать среднее значение распределения значений *Y* для наблюдений с заданным набором значений *X*, сообщая определенные веса переменным *X* и суммируя их.

Обратите внимание, что мы не делали никаких предположений относительно распределения значений независимых переменных *Xj*. Они, в отличие от *Y*, не обязательно должны иметь нормальное распределение. На самом деле они часто бывают категориальными (например, при дисперсионном анализе). Кроме того, допускаются нелинейные комбинации независимых переменных. Нередко в уравнение входят такие независимые переменные как *X2* или

*X1×* *X2*. Что здесь важно, так это линейность комбинаций параметров (β0, β1,

При создании обобщенных линейных моделей мы подбираем модели в виде

(4.2)

где *g*(*Y*) – это функция условного среднего (называемая связующей функцией). Кроме того, вы отказываетесь от предположения о нормальном распределении значений *Y*. Вместо этого вы подразумеваете, что *Y* подчиняется распределению из семейства экспоненциальных распределений. Вы задаете связующую функцию и вероятностное распределение, а параметры оцениваются при помощи итеративного алгоритма максимального правдоподобия.

Обобщенные линейные модели обычно подгоняются в R при помощи

функции glm() (хотя доступны и другие функции). Формат применения этой функции сходен с таковым для функции lm(), но имеет дополнительные параметры. Основной формат функции таков:

glm(formula, family=*family*(link=*function*), data=)

Типы распределений вероятностей (*family*) и соответствующие связующие функции (*function*) по умолчанию приведены в табл. 4.1.

Таблица 4.1

|  |  |
| --- | --- |
| **Семейство распределений** | **Связующая функция по умолчанию** |
| binominal | (link=”logit”) |
| gaussian | (link=”identity”) |
| gamma | (link=”inverse”) |
| Inverse gaussian | (link=”1/mu^2”) |
| poisson | (link=”log”) |
| quasi | (link=”identity”, variance=”constant”) |
| quasibinominal | (link=”logit”) |
| quasipoisson | (link=”log”) |

Функция glm() позволяет подгонять разные распространенные модели, включая логистическую и пуассоновскую регрессии. Это можно продемонстрировать на двух первых моделях следующим образом.

Предположим, что у вас есть одна зависимая переменная (Y) и три

независимых (X1, X2, X3), которые находятся в одной таблице данных

(mydata). Логистическая регрессия подходит для дихотомических зависимых переменных (0, 1). Модель предполагает, что *Y* имеет биномиальное распределение и что можно подогнать линейную модель следующего вида:

(4.3)

Вывод соотношения (3.3) поясним на примере обобщения парной линейной регрессии. В условиях двоичных переменных, закодированных как 0 и 1, будем рассматривать вероятность получения одного из этих значений, обозначая их как *р*(Х) для этого применяем логистическую переменную, которая дает значения на интервале от 0 до 1.

(4.4)

После небольших преобразований получим

(4.5)

После логарифмирования выражения (4.5) получим выражение (4.3) справедливое для парной регрессии

(4.6)

Левая часть выражения (4.6) называется логарифмом риска или логитом. Таким образом логит регрессионной модели (4.1) является линейным относительно предиктора *Х.*

* + 1. **Оценивание коэффициентов логистической регрессии**

Коэффициенты и  неизвестны и подлежат определению с последующим оцениванием их значений. Для этого используется обучающая выборка данных. Ранее для этого использовался метод наименьших квадратов. В принципе возможно использование метода наименьших квадратов и в этом случае как решение задачи для нелинейной функции (4.6). Однако предпочтительней в данном случае является использование более общего метода максимального правдоподобия. Суть его заключается в следующем: используя выражение (3.4), мы подбираем такие коэффициенты и , чтобы вероятность *р*(Х) максимально соответствовала условиям решения конкретной задачи. Например, в [4] рассматривается задача предсказания дефолта кредитора на основе данных Default. В данном примере вероятность *р*(Х) должна соответствовать наблюдаемому статусу клиента. Другими словами, при найденных коэффициентах и  вероятность дефолта для неплательщиков кредита должна быть близка к 1, а вероятность дефолта исправных плательщиков должна стремиться к нулю. Это можно реализовать с помощью так называемой функции правдоподобия

(4.7)

На самом деле метод максимального правдоподобия, основанный на максимизации функционала (4.7) является обобщением метода наименьших квадратов и тем самым осуществляет алгоритмически задачу обобщения линейной регрессии. Проверка гипотезы *H*0: осуществляется с помощью вычисления стандартных ошибок и *z –* критерия Лапласа, играющего ту же роль, что *t –* критерий Стьюдента для линейной регрессии.

Ниже приводится результат проверки логистической парной регрессии на примере данных Default [4] по предиктору balance. Здесь нулевая гипотеза предполагает, что вероятность дефолта кредитора не зависит от баланса его кредитной карты. В таблице 4.2 приводятся коэффициенты логистической регрессии, предсказывающей вероятность дефолта на основе размера долга на кредитной карте.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оцениваемые величины | Коэффициент | Ст. ошибка | *Z -* критерий | *Р -* значение |
| Свободный член | -10.6513 | 0.3612 | -29.5 | <0.0001 |
| balance | 0.0055 | 0.0002 | 24.9 | <0.0001 |

Можно видеть, что увеличение долга на одну единицу приводит к увеличению риска дефолта на 0.0055 единиц. Эти результаты мы можем использовать для предсказания, т.е. вычислить вероятность дефолта клиента. Так, если клиент имеет долг на кредитной карте размером 1000$, то вероятность его дефолта рассчитывается по формуле (4.4)

Если долг составляет 2000$, то вероятность существенно возрастает

* + 1. **Множественная логистическая регрессия**

Вернемся к общей формуле множественной логистической регрессии

Из этой формулы следует

(4.8)

Как и в случае парной логистической регрессии применим метод максимального правдоподобия. В таблице приведены результаты расчета по этому методу для трех предикторов: balance, income, student[yes].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оцениваемые величины | Коэффициент | Ст. ошибка | *Z -* критерий | *Р -* значение |
| Свободный член | -10.8690 | 0.4923 | -22.08 | <0.0001 |
| balance | 0.0057 | 0.0002 | 24.74 | <0.0001 |
| Income | 0.0030 | 0.0082 | 0.37 | 0.7115 |
| Student[yes] | -0.6468 | 0.2362 | -2.74 | 0.0062 |

Предиктор student входит как двоичная переменная со значениями 1, если это студент и 0, если не студент. Сравним результат, полученный по множественной логистической регрессии с парной логистической регрессией, где в качестве предиктора используется качественный предиктор student[yes].

Здесь мы видим обратный результат: коэффициент при переменной Student[yes] положительный, а в трехмерной регрессии отрицательный. Причина в том, что переменные student[yes] и balance коррелированы, это значит, что студенты в целом имеют большие долги на карте, т.е. этим обуславливается высокая вероятность их дефолта. Подсчитаем вероятности дефолта студентов с долгом 1500$ и доходом (income) 40000$

Оценка дефолта для не студента с такими же показателями долга и дохода

Если качественные переменные принимают значения из класса индексных величин большей размерности, чем в классе двоичных переменных, как в примере с определением статуса больного, то необходимо учитывать, что соответствующие вероятности описывают полную группу событий. Такой сценарий представляет вероятности событий ; ; =1--.

* 1. **Дискриминантный анализ**

Логистическая регрессия применительно к двоичным переменным реализуется проще и поэтому применяется в основном для этого класса переменных. Однако для более высоких классов по числу индексных переменных более оправдан так называемый Байесовский подход, так те варианты, которые были рассмотрены в конце предыдущего раздела для вычисления вероятностей представляли собой не что иное как условные вероятности. На этом основан так называемый дискриминантный анализ. Предположим, что мы хотим описать некоторые наблюдения, отнесенные к одному из *К* классов индексных переменных, при этом *К>=2.* Другими словами, переменная *Y* может принимать *К* четко различающихся и неупорядоченных значений. Пусть обозначает общую априорную тому классу, функция есть плотность вероятности для некоторого наблюдения из *k –* того класса. Тогда теорема Байеса утверждает, что

(4.9)

Как правило, нахождение априорной вероятности не составляет большого труда, а определение плотности распределения , так называемой плотности апостериорной вероятности, сопряжено с некоторыми трудностями.

Представим пока, что у нас имеется только один предиктор. Рассмотрим для него реализацию линейного дискриминантного анализа. Предположим, что плотность распределения совпадает с нормальной

(4.10)

Подставив (3.10) в (3.9), получим

(4.11)

Выражение (3.11) называют байесовским классификатором, именно он определяет апостериорную вероятность. Метод линейного дискриминантного анализа (*LDA*) аппроксимирует байесовский классификатор, используя оценки

; (4.12)

Следует отметить, что после логарифмирования выражения (3.11) мы получим так называемую дискриминантную функцию

(4.13)

Можно показать, что байесовский классификатор относит наблюдение к классу, у которого дискриминантная функция максимальна.

Теперь перейдем к случаю, когда число предикторов больше одного. Это расширение связано с многомерными гауссовскими (нормальными) распределениями. Обобщение на многомерные распределения приводит к формальному использованию общего представления плотности распределения в случае многомерных случайных величин. Пусть у нас *р-*мерная случайная величина имеет нормальное распределение. Тогда плотность вероятности многомерного распределения запишется в виде

(4.14)

Где означает ковариационную матрицу многомерной случайной величины , имеющей нормальное распределение. Из выражения (4.14) можно получить дискриминантную функцию аналогичным образом, как и в случае . В линейном дискриминантном анализе ковариационная матрица, определяемая оператором постоянна и именно через неё находится плотность вероятности и , следовательно дискриминантная функция.

(4.15)

Альтернативным подходом является алгоритм квадратичного дискриминантного анализа *QDA* . В квадратичном дискриминантном анализе предполагается непостоянство оператора . В *QDA* предполагается, что матрица ковариаций относится к классу *k.* Термин квадратичный определяется видом дискриминантной функции. В этом случае вид дискриминантной функции преобразуется к квадратичной форме

(4.16)

* 1. **Практические применения методов классификации**

**4.4.1 Данные по цене акции**

Здесь мы используем данные Smarket, которые являются частью библиотеки ISLR приведены в [4].

|  |
| --- |
| > pairs(Smarket)  > library(ISLR)  > names(Smarket)  [1] "Year" "Lag1" "Lag2" "Lag3" "Lag4" "Lag5" "Volume"  [8] "Today" "Direction"  > dim(Smarket)  [1] 1250 9  > summary(Smarket)  Year Lag1 Lag2 Lag3  Min. :2001 Min. :-4.922000 Min. :-4.922000 Min. :-4.922000  1st Qu.:2002 1st Qu.:-0.639500 1st Qu.:-0.639500 1st Qu.:-0.640000  Median :2003 Median : 0.039000 Median : 0.039000 Median : 0.038500  Mean :2003 Mean : 0.003834 Mean : 0.003919 Mean : 0.001716  3rd Qu.:2004 3rd Qu.: 0.596750 3rd Qu.: 0.596750 3rd Qu.: 0.596750  Max. :2005 Max. : 5.733000 Max. : 5.733000 Max. : 5.733000  Lag4 Lag5 Volume Today Direction  Min. :-4.922000 Min. :-4.92200 Min. :0.3561 Min. :-4.922000 Down:602  1st Qu.:-0.640000 1st Qu.:-0.64000 1st Qu.:1.2574 1st Qu.:-0.639500 Up :648  Median : 0.038500 Median : 0.03850 Median :1.4229 Median : 0.038500  Mean : 0.001636 Mean : 0.00561 Mean :1.4783 Mean : 0.003138  3rd Qu.: 0.596750 3rd Qu.: 0.59700 3rd Qu.:1.6417 3rd Qu.: 0.596750  Max. : 5.733000 Max. : 5.73300 Max. :3.1525 Max. : 5.733000    > pairs(Smarket) |

Приведенный код создает корреляционные матрицы со значениями всех парных корреляций

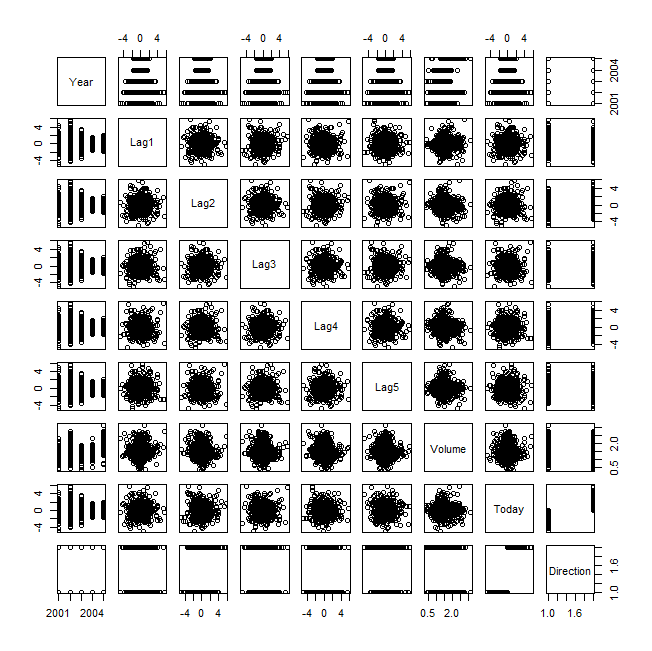


Рис. 4

Функция cor () создает матрицу, со значениями всех парных корреляции между предикторами в наборе данных.

|  |
| --- |
| > cor(Smarket )  Error in cor(Smarket) : 'x' must be numeric  > cor(Smarket [,-9])  Year Lag1 Lag2 Lag3 Lag4  Year 1.00000000 0.029699649 0.030596422 0.033194581 0.035688718  Lag1 0.02969965 1.000000000 -0.026294328 -0.010803402 -0.002985911  Lag2 0.03059642 -0.026294328 1.000000000 -0.025896670 -0.010853533  Lag3 0.03319458 -0.010803402 -0.025896670 1.000000000 -0.024051036  Lag4 0.03568872 -0.002985911 -0.010853533 -0.024051036 1.000000000  Lag5 0.02978799 -0.005674606 -0.003557949 -0.018808338 -0.027083641  Volume 0.53900647 0.040909908 -0.043383215 -0.041823686 -0.048414246  Today 0.03009523 -0.026155045 -0.010250033 -0.002447647 -0.006899527  Lag5 Volume Today  Year 0.029787995 0.53900647 0.030095229  Lag1 -0.005674606 0.04090991 -0.026155045  Lag2 -0.003557949 -0.04338321 -0.010250033  Lag3 -0.018808338 -0.04182369 -0.002447647  Lag4 -0.027083641 -0.04841425 -0.006899527  Lag5 1.000000000 -0.02200231 -0.034860083  Volume -0.022002315 1.00000000 0.014591823  Today -0.034860083 0.01459182 1.000000000 |

Коэффициенты корреляции между лаг- переменными и доходностью теперь близки к нулю. Изобразив данные графически, мы увидим, что Volume увеличивается во времени.

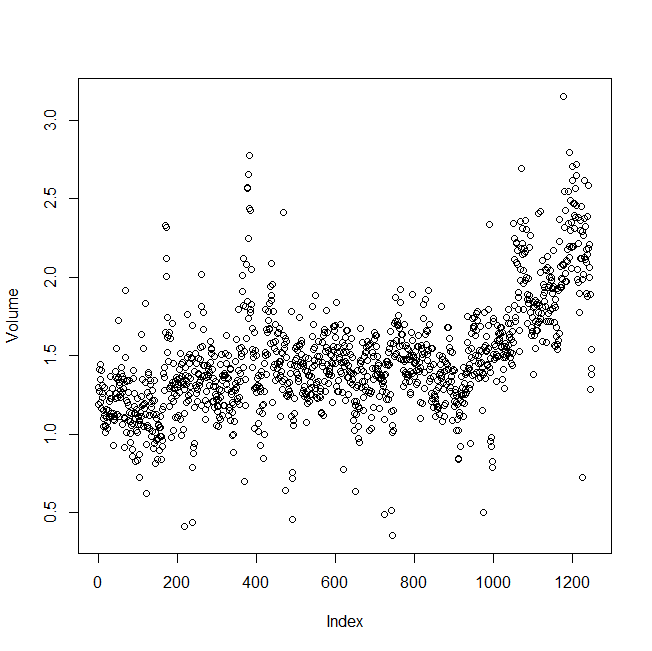


Рис. 5 Корреляционное поле Volume – Index

**4.4.2 Логистическая регрессия на примере биржевых операций**

Функция glm () соответствует обобщенным линейным моделям glm (), классу моделей, который включает логическую регрессию.

|  |
| --- |
| > glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume ,data=Smarket ,family=binomial )  > summary (glm.fit)  Call:  glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +  Volume, family = binomial, data = Smarket)  Deviance Residuals:  Min 1Q Median 3Q Max  -1.446 -1.203 1.065 1.145 1.326  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) -0.126000 0.240736 -0.523 0.601  Lag1 -0.073074 0.050167 -1.457 0.145  Lag2 -0.042301 0.050086 -0.845 0.398  Lag3 0.011085 0.049939 0.222 0.824  Lag4 0.009359 0.049974 0.187 0.851  Lag5 0.010313 0.049511 0.208 0.835  Volume 0.135441 0.158360 0.855 0.392  (Dispersion parameter for binomial family taken to be 1)  Null deviance: 1731.2 on 1249 degrees of freedom  Residual deviance: 1727.6 on 1243 degrees of freedom  AIC: 1741.6  Number of Fisher Scoring iterations: 3  используем функцию coef () для доступа к модели.  > coef(glm.fit)  (Intercept) Lag1 Lag2 Lag3 Lag4 Lag5  -0.126000257 -0.073073746 -0.042301344 0.011085108 0.009358938 0.010313068  Volume  0.135440659 |

Для детального доступа модели используем функцию summary()

|  |
| --- |
| > summary (glm.fit)$coef  Estimate Std. Error z value Pr(>|z|)  (Intercept) -0.126000257 0.24073574 -0.5233966 0.6006983  Lag1 -0.073073746 0.05016739 -1.4565986 0.1452272  Lag2 -0.042301344 0.05008605 -0.8445733 0.3983491  Lag3 0.011085108 0.04993854 0.2219750 0.8243333  Lag4 0.009358938 0.04997413 0.1872757 0.8514445  Lag5 0.010313068 0.04951146 0.2082966 0.8349974  Volume 0.135440659 0.15835970 0.8552723 0.3924004  > summary (glm.fit)$coef[,4]  (Intercept) Lag1 Lag2 Lag3 Lag4 Lag5  0.6006983 0.1452272 0.3983491 0.8243333 0.8514445 0.8349974  Volume  0.3924004  Воспользуемся функцией predict для предсказания того, что доходность индекса пойдет вверх  > glm.probs=predict (glm.fit ,type="response")  > glm.probs [1:10]  1 2 3 4 5 6 7 8  0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565 0.4926509 0.5092292  9 10  0.5176135 0.4888378  > contrasts (Direction )  Up  Down 0  Up 1  > glm.pred=rep("Down" ,1250)  > glm.pred[glm.probs >.5]=" Up"  > table(glm.pred ,Direction )  Direction  glm.pred Down Up  Up 457 507  Down 145 141  > glm.pred[glm.probs >.5]="Up"  > table(glm.pred ,Direction )  Direction  glm.pred Down Up  Down 145 141  Up 457 507  > (507+145) /1250  [1] 0.5216  > mean(glm.pred==Direction )  [1] 0.5216  > train=(Year<2005)  > Smarket.2005= Smarket [!train ,]  > dim(Smarket.2005)  [1] 252 9  > Direction.2005= Direction [!train]  > glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume ,data=Smarket, family=binomial, subset=train)  > glm.probs=predict (glm.fit ,Smarket.2005, type="response")  > glm.pred=rep("Down",252)  > glm.pred[glm.probs >.5]="Up"  > table(glm.pred ,Direction.2005)  Direction.2005  glm.pred Down Up  Down 77 97  Up 34 44  > mean(glm.pred==Direction.2005)  [1] 0.4801587  > mean(glm.pred!=Direction.2005)  [1] 0.5198413  > glm.fit=glm(Direction~Lag1+Lag2 ,data=Smarket ,family=binomial ,subset=train)  > glm.probs=predict (glm.fit ,Smarket.2005, type="response")  > glm.pred=rep("Down",252)  > glm.pred[glm.probs >.5]="Up"  > table(glm.pred ,Direction.2005)  Direction.2005  glm.pred Down Up  Down 35 35  Up 76 106  > mean(glm.pred==Direction.2005)  [1] 0.5595238  > 106/(106+76)  [1] 0.5824176 |

Предположим, что мы хотим предсказать доходность, связанную с

значениями Лаг1 и Лаг2. В частности, мы хотим спрогнозировать направление на день, когда Lag1 и Lag2 равны 1,2 и 1,1, соответственно, и в день, когда они равны 1.5 и -0,8. Мы делаем это с помощью функции predict

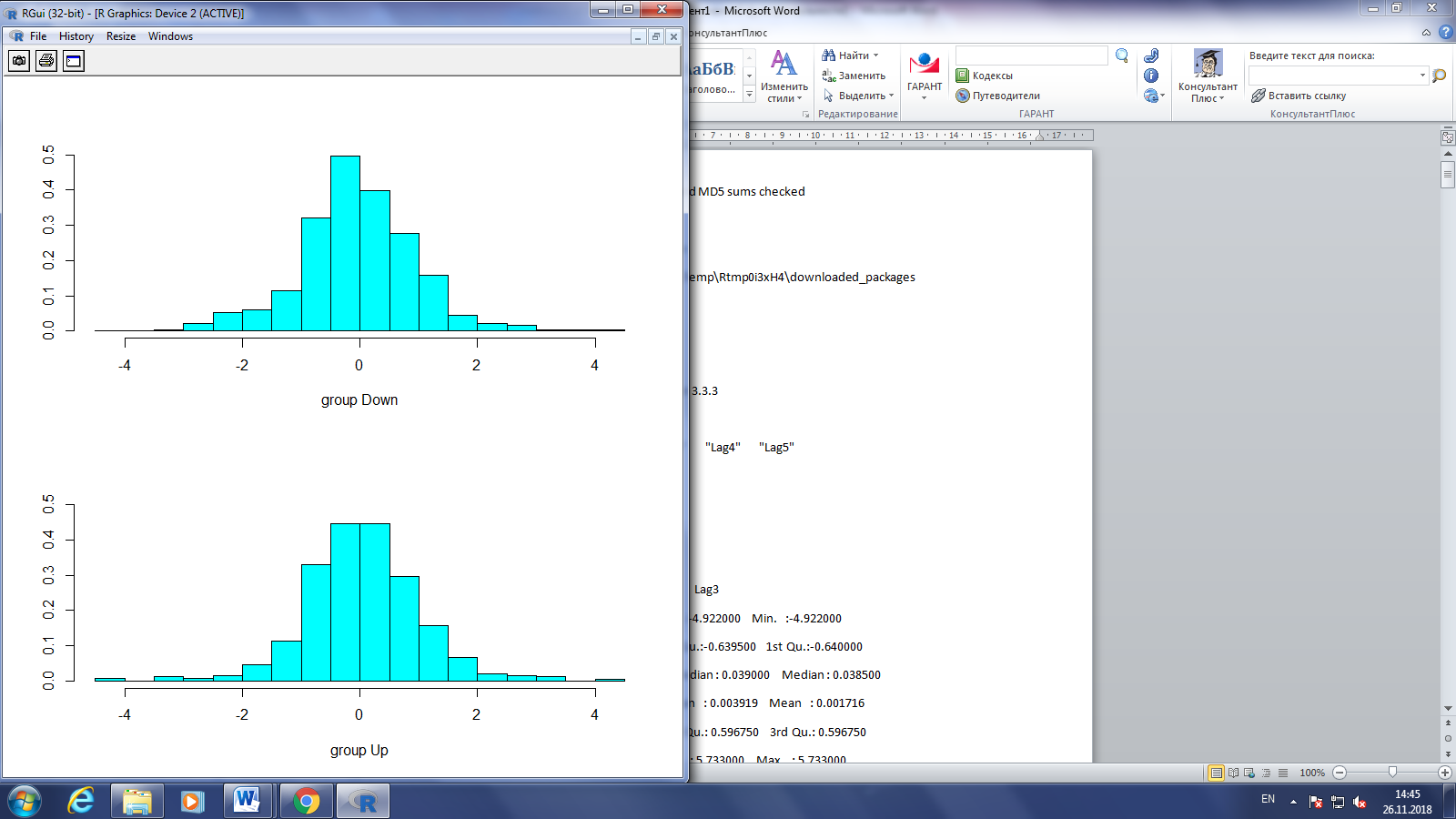
|  |
| --- |
| > predict (glm.fit ,newdata =data.frame(Lag1=c(1.2 ,1.5),Lag2=c(1.1,-0.8) ),type="response")  1 2  0.4791462 0.4960939 |

**4.4.3 Линейный дискриминантный анализ (LDA)**

Выполним LDA на наборе данных Smarket из базы данных ISLR [4]. Для этого воспользуемся функцией lda.fit из библиотеки MASS.

|  |
| --- |
| > library(MASS)  > lda.fit=lda(Direction~Lag1+Lag2 ,data=Smarket ,subset=train)  > lda.fit  Call:  lda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)  Prior probabilities of groups:  Down Up  0.491984 0.508016  Group means:  Lag1 Lag2  Down 0.04279022 0.03389409  Up -0.03954635 -0.03132544  Coefficients of linear discriminants:  LD1  Lag1 -0.6420190  Lag2 -0.5135293  > plot(lda.fit) |

На выходе получим два графика гистограмм по дням для результатов сгруппированных по признакам «Down» - повышение и «Up» - понижение.



Функция predict () возвращает список с тремя элементами.

|  |
| --- |
| > lda.pred=predict (lda.fit , Smarket.2005)  > names(lda.pred)  [1] "class" "posterior" "x"  > lda.class=lda.pred$class  > table(lda.class ,Direction.2005)  Direction.2005  lda.class Down Up  Down 35 35  Up 76 106  > mean(lda.class==Direction.2005)  [1] 0.5595238  > sum(lda.pred$posterior[,1]>=.5)  [1] 70  > sum(lda.pred$posterior[,1]<.5)  [1] 182  > lda.pred$posterior [1:20,1]  999 1000 1001 1002 1003 1004 1005 1006  0.4901792 0.4792185 0.4668185 0.4740011 0.4927877 0.4938562 0.4951016 0.4872861  1007 1008 1009 1010 1011 1012 1013 1014  0.4907013 0.4844026 0.4906963 0.5119988 0.4895152 0.4706761 0.4744593 0.4799583  1015 1016 1017 1018  0.4935775 0.5030894 0.4978806 0.4886331  > lda.class [1:20]  [1] Up Up Up Up Up Up Up Up Up Up Up Down Up Up Up  [16] Up Up Down Up Up  Levels: Down Up  > sum(lda.pred$posterior[,1]>.9)  [1] 0 |

Получаем, что ни один день в 2005 году не достигнет этого порога! Фактически, наибольшая вероятность снижения за весь 2005 год составила 52,02 %.

**4.4.4 Квадратичный дискриминантный анализ (QDA)**

Метод QDA реализуется с помощью функции qda.fit

|  |
| --- |
| > qda.fit=qda(Direction~Lag1+Lag2 ,data=Smarket ,subset=train)  > qda.fit  Call:  qda(Direction ~ Lag1 + Lag2, data = Smarket, subset = train)  Prior probabilities of groups:  Down Up  0.491984 0.508016  Group means:  Lag1 Lag2  Down 0.04279022 0.03389409  Up -0.03954635 -0.03132544  > qda.class=predict (qda.fit ,Smarket.2005) $class  > qda.class=predict (qda.fit ,Smarket .2005) $class  > table(qda.class ,Direction.2005)  Direction.2005  qda.class Down Up  Down 30 20  Up 81 121  > mean(qda.class==Direction.2005)  [1] 0.5992063 |

Интересно, что прогнозы QDA точны почти в 60 % случаев,

несмотря на то, что данные за 2005 год не были использованы для соответствия модели. Этот уровень точности довольно впечатляет для данных фондового рынка, который, как известно, довольно трудно точно смоделировать. Это говорит о том, что квадратичная форма QDA может захватить истинное отношение более точно, чем линейные формы LDA и логистической регрессии. Однако, мы рекомендуем оценку эффективности этого метода на большом тестовом наборе перед ставкой, именно такой подход будет последовательно обыгрывать рынок

**4.4.5 Метод К-ближайших соседей**

Метод ближайших соседей реализуется с помощью функции knn() из библиотеки class. В отличие от других функций, рассмотренных выше, функция knn() формирует предсказание без промежуточных шагов, т.е. без двухэтапной подгонки.

|  |
| --- |
| > library(class)  > train.X=cbind(Lag1 ,Lag2)[train ,]  > test.X=cbind(Lag1 ,Lag2)[!train ,]  > train.Direction =Direction [train]  > set.seed(1)  > knn.pred=knn(train.X,test.X,train.Direction ,k=1)  > table(knn.pred ,Direction.2005)  Direction.2005  knn.pred Down U  Down 43 58  Up 68 83  > (83+43)/252  [1] 0.5  > knn.pred=knn(train.X,test.X,train.Direction ,k=3)  > table(knn.pred ,Direction.2005)  Direction.2005  knn.pred Down Up  Down 48 54  Up 63 87  > mean(knn.pred==Direction .2005)  Error: unexpected numeric constant in "mean(knn.pred==Direction .2005"  > mean(knn.pred==Direction.2005)  [1] 0.5357143 |

Результаты несколько улучшились. Но дальше увеличение K никаких дальнейших улучшений не дает. Похоже, что для этих данных QDA

обеспечивает наилучшие результаты методов, которые мы рассмотрели до сих пор.

**4.4.6 Применение к данным по жилым прицепам**

|  |
| --- |
| > dim(Caravan )  [1] 5822 86  > attach(Caravan)  > summary(Purchase )  No Yes  5474 348  > 348/5822  [1] 0.05977327  > standardized.X= scale(Caravan [,-86])  > var(Caravan [ ,1])  [1] 165.0378  > var(Caravan [ ,2])  [1] 0.1647078  > var(standardized.X[,1])  [1] 1  > var(standardized.X[,2])  [1] 1  > test=1:1000  > train.X= standardized.X[-test ,]  > test.X= standardized.X[test ,]  > train.Y=Purchase [-test]  > test.Y=Purchase [test]  > set.seed(1)  > knn.pred=knn(train.X,test.X,train.Y,k=1)  > mean(test.Y!=knn.pred)  [1] 0.118  > mean(test.Y!="No")  [1] 0.059  > table(knn.pred ,test.Y)  test.Y  knn.pred No Yes  No 873 50  Yes 68 9  > 9/(68+9)  [1] 0.1168831  > knn.pred=knn(train.X,test.X,train.Y,k=3)  > table(knn.pred ,test.Y)  test.Y  knn.pred No Yes  No 920 54  Yes 21 5  > 5/26  [1] 0.1923077  > knn.pred=knn(train.X,test.X,train.Y,k=5)  > table(knn.pred ,test.Y)  test.Y  knn.pred No Yes  No 930 55  Yes 11 4  > 4/15  [1] 0.2666667 |

Для сравнения, мы также можем подогнать модель логистической регрессии к данным. Если мы используем 0.5 в качестве предсказанной вероятности отсечения для классификатора, то у нас есть проблема: прогнозируется, что только семь тестовых наблюдений обеспечивают страхование покупок. Что еще хуже, мы ошибаемся по поводу всего этого! Однако, мы не обязаны использовать отсечение 0,5. Если вместо этого мы прогнозируем покупку каждый раз, когда прогнозируемая вероятность покупки превышает 0,25, мы получаем много лучшие результаты: мы прогнозируем, что 33 человека приобретут страховку, и мы правы относительно примерно 33 % этих людей. Это более чем в пять раз лучше, чем случайное угадывание!

|  |
| --- |
| > glm.fit=glm(Purchase~.,data=Caravan,family=binomial, subset=-test)  Warning message:  glm.fit: fitted probabilities numerically 0 or 1 occurred  > glm.probs=predict (glm.fit ,Caravan [test ,], type="response ")  Error in match.arg(type) :  'arg' should be one of “link”, “response”, “terms”  > glm.probs=predict (glm.fit ,Caravan [test ,], type="response")  > glm.pred=rep("No",1000)  > glm.pred[glm.probs >.5]=" Yes"  > glm.pred[glm.probs >.5]="Yes"  > table(glm.pred ,test.Y)  test.Y  glm.pred No Yes  No 934 59  Yes 7 0  > glm.pred=rep("No",1000)  > glm.pred[glm.probs >.25]=" Yes"  > table(glm.pred ,test.Y)  test.Y  glm.pred No Yes  Yes 22 11  No 919 48  > glm.pred=rep("No",1000)  > glm.pred[glm.probs >.25]="Yes"  > table(glm.pred ,test.Y)  test.Y  glm.pred No Yes  No 919 48  Yes 22 11  > 11/(22+11)  [1] 0.3333333 |

1. **Методы создания повторных выборок**

Методы создания повторных выборок (resampling methods) подразумевают технологию отбора обучающих выборок с целью улучшения качества подобранных моделей в работе с разными подмножествами. В данной главе будут рассмотрены два основных подхода к улучшению предсказаний на основе подобранных моделей, основанных на основе повторных выборок: метод перекрестных проверок и бустреп. Первый служит для оценки проверяемых моделей (model assessment), а второй чаще как отбор для оценки уровня гибкости модели (model selection).

**5.1 Перекрестная проверка**

Ранее обсуждались вопросы различия частоты ошибок на контрольной (проверочной) выборке и на обучающей выборке. Частота ошибок на контрольной выборке находится как средняя ошибка, которая возникает в результате применения статистического обучения для предсказания отклика на новых наблюдениях, которые отсутствовали в обучающей выборке. При этом использование того или иного метода статистического обучения становится оправданным, если результат обеспечивает низкую ошибку на новых данных. Однако не всегда контрольная выборка может обеспечить адекватную проверку выбранного метода. В реальности формирование контрольной выборки ограничено возможностями наличия репрезентативных наборов данных. Что же касается обучающей выборки, то ошибки в ней рассчитать проще, применяя некоторую модель непосредственно к наблюдениям, на которых она была обучена.

В отсутствие большой и репрезентативной контрольной выборки на практике используется процедура оценивания построенной модели на обучающей выборке, кот можно изменить предикторы в пределах обучающей выборки. Все эти действия предназначены для оценки влияния изменений на качество модели. Эти приемы составляют суть перекрестных проверок выбранной модели для предсказания.

**5.1.1 Метод проверочной выборки**

Метод проверочной выборки (the validation set approach) представляет собой алгоритм случайного разбиения имеющегося набора данных на части: обучающую и проверочную. Модель строится на обучающей выборке, а затем применяется для предсказания отклика на проверочной выборке. Такие действия чаще называют валидацией модели или кросс-валидацией.

В предыдущем разделе мы изучали методы выбора входящих в уравнение регрессии переменных. Если ваша главная задача – описание, работа заканчивается выбором регрессионной модели и ее интерпретацией. Однако если ваша цель – это предсказание, вы имеете все основания спросить: ≪Насколько хорошо полученное уравнение работает в реальном мире?≫

По определению, регрессионные методы позволяют рассчитать оптимальные для имеющегося набора данных параметры. При использовании МНК-регрессии параметры модели подбираются так, чтобы минимизировать сумму квадратов ошибок предсказаний (остатков) и, напротив, максимизировать долю объясненной дисперсии зависимой переменной (коэффициент детерминации). Поскольку уравнение оптимизировано для имеющегося набора данных, оно не даст таких же хороших результатов для других данных. Рассмотрим пример из [5], в котором необходимо предсказать число затрачиваемых человеком калорий во время занятий фитнесом по продолжительности выполнения им упражнений, его возрасту, полу и индексу массы тела. Если вы подберете для этих данных уравнение

МНК-регрессии, то получите значения параметров модели, которые максимизируют коэффициент детерминации этого *конкретного* набора данных. Однако возникает вполне естественное желание использовать это уравнение для предсказания числа затраченных людьми калорий в общем, а не только теми, кто участвовал в исходном исследовании. Можем ли мы утверждать, что уравнение не будет (или будет) работать так же хорошо для нового набора наблюдений, как это можно проверить? Кросс-валидация –

это полезный метод для оценки применимости модели к генеральной совокупности. При кросс-валидации часть данных используется как обучающая выборка, а часть – как тестовая. Регрессионное уравнение подгоняется для обучающей выборки, а затем применяется для проверочной. Поскольку проверочная выборка не использовалась для подбора параметров модели, применимость модели к этой выборке – хорошая оценка применимости полученных параметров модели к новым данным. В книге [4] кросс-валидация применяется к данным базы Auto, где исходная обучающая выборка из 392 наблюдений была разбита на две равные части по 196 наблюдений в каждой части. По мере выбора различных способов разбиения исходной выборки были построены графики средне квадратической ошибки MSE сначала для одного разбиения в зависимости от степени полинома, использованного в качестве регрессионной модели, а затем с учетом различных 10 способов разбиения. Результаты показывают, что оба этих фактора значительно влияют как на величину MSE, так и её поведение.

**5.1.2 Перекрестная выборка по отдельным наблюдениям**

Перекрестная выборка по отдельным наблюдениям (leave–one-out cross-validation, LOOCV) вытекает из предыдущего метода проверочной выборки. В предыдущем разделе порядок разбиения не имеет четкого алгоритма. Этот произвол преодолевается в методе LOOCV. Перекрестная проверка по отдельным наблюдениям имеет упорядоченную процедуру организации разбиений. Вместо создания двух выборок сравнимого размера в этом методе одно наблюдение принимается в качестве контрольного, а остальные - составляют обучающую выборку. Затем, после получения оценки MSE1, в качестве в качестве контрольного наблюдения выбирают пару , оставшиеся образуют обучающую выборку. Таким образом, вычисляя MSEi, можно продолжить процедуру перекрестной проверки. Вычисленные значения средне квадратических ошибок для каждого шага перекрестной проверки используются для нахождения среднего значения LOOCV- оценки MSE:

(5.1)

Помимо упорядоченности алгоритма, данный метод кросс-валидации имеет гораздо меньшее смещение ошибок, за счет многократного повторения однотипных операций мы получаем простой алгоритм перекрестных проверок. Для линейной и полиноминальной регрессии, рассчитываемых по методу наименьших квадратов получается очень простой и экономный алгоритм, который выполняется по одной рекуррентной формуле

(5.2)

Где прогнозируемое значение отклика, –показатель разбалансировки, который для простой регрессии выглядит как

индекс \* отмечает различные способы формирования выборок.

**5.1.3 *k –*кратная перекрестная проверка**

Альтернативой методу LOOCV является алгоритм *k –*кратной перекрестной проверки (k-fold cross-validation). При *k*-кратной кросс-валидации выборка разделяется на *k* подвыборок. Каждая из них играет роль тестовой выборки, а объединенные данные оставшихся *k* – 1 подвыборок используются как обучающая группа. Применимость *k* уравнений к *k* тестовым выборкам фиксируется и затем усредняется. Если *k* = *n* (общему числу наблюдений),

то такой подход называется оценкой по методу ≪складного ножа≫ (последовательного исключения значений выборки с возвратом –jackknifing).

Выполнить *k*-кратную кросс-валидацию можно при помощи функции crossval() из пакета bootstrap. Следующий программный код содержит функцию shrinkage() для перекрестной *k*-кратной проверки коэффициента детерминации.

|  |
| --- |
| shrinkage <- function(fi t, k=10){  require(bootstrap)  theta.fi t <- function(x,y){lsfi t(x,y)}  theta.predict <- function(fi t,x){cbind(1,x)%\*%fi t$coef}  x <- fi t$model[,2:ncol(fi t$model)]  y <- fi t$model[,1]  results <- crossval(x, y, theta.fi t, theta.predict, ngroup=k)  r2 <- cor(y, fi t$fi tted.values)^2  r2cv <- cor(y, results$cv.fi t)^2  cat(“Original R-square =”, r2, “\n”)  cat(k, “Fold Cross-Validated R-square =”, r2cv, “\n”)  cat(“Change =”, r2-r2cv, “\n”)  } |

При помощи этого программного кода вы создаете нужные функции, формируете матрицу значений независимых и зависимых переменных, вычисляете обычный коэффициент детерминации и проводите расчет этого коэффициента методом кросс-валидации (методы бутстреп-анализа детально разобраны далее, в разделе 5.2). Далее функция shrinkage() использована для 10-кратной кросс-валидации со всеми четырьмя независимыми переменными для набора данных прогнозирования уровня преступности в разных штатах states [5]:

|  |
| --- |
| > fi t <- lm(Murder ~ Population + Income + Illiteracy + Frost, data=states)  > shrinkage(fi t)  Original R-square=0.567  10 Fold Cross-Validated R-square=0.4481  Change=0.1188 |

Видно, что коэффициент детерминации, рассчитанный для выборки (0.567), чересчур оптимистичен. Лучшая оценка доли изменчивости уровня преступности, объясненной нашей моделью на новых данных, – это коэффициент детерминации, полученный методом перекрестной оценки (0.448). Учтите, что наблюдения разделяются на *k* групп случайно, так что у вас будут получаться немного разные результаты при каждом использовании функции shrinkage(). Кросс-валидацию можно использовать при выборе переменных, отдавая предпочтение модели, которая лучше применяется к генеральной совокупности. Например, модель с двумя независимыми переменными (Population и Illiteracy) характеризуется меньшим снижением коэффициента детерминации при кросс-валидации (0.03), по сравнению с 0.12 для модели, включающей все переменные:

|  |
| --- |
| > fi t2 <- lm(Murder~Population+Illiteracy,data=states)  > shrinkage(fit2)  Original R-square=0.5668327  10 Fold Cross-Validated R-square=0.5346871  Change=0.03214554 |

Это обстоятельство может послужить аргументом в пользу модели с двумя переменными. При прочих равных условиях уравнение регрессии, полученное на основании большей обучающей выборки и лучше применимое для генеральной совокупности, при перекрестной проверке будет оценено выше. В этом случае коэффициент детерминации будет уменьшаться не так сильно, а предсказания получатся более точными.

* 1. **Бустреп**

**5.2.1 Понятие о бустреп-анализе**

Бустреп – метод применяемый для количественной оценки качества модели в условиях неопределенности оценок статистических параметров и методов статистического обучения. Бутстреп-анализ (bootstrapping) создает эмпирическое распределение тестовой статистики или набора статистик путем создания многих случайных выборок с возвратом из исходной выборки. Это позволяет вычислять доверительные интервалы и проверять статистические гипотезы без опоры на определенное теоретическое распре-деление данных. Проще всего объяснить логику бутстреп-анализа на примере. Допустим, вы хотите рассчитать 95%-ный доверительный интервал для выборочного среднего. У вас есть 10 наблюдений, выборочное среднее равно 40, а выборочное стандартное отклонение равно 5. Если вы готовы допустить нормальное распределение выборочного среднего, то (1 – α/2)

%-ный доверительный интервал может быть вычислен с использованием следующего выражения:

где t – это верхнее (1 – α/2) критическое значение для распределе-ния t с n – 1 степенями свободы. В случае 95%-го доверительного ин-тервала вы получите 40 – 2.262(5/3.163) < μ < 40 + 2.262(5/3.162), или 36.424 < μ < 43.577. Вы будете ожидать, что вычисленные таким образом значения доверительного интервала будут окружать реальное среднее значение для генеральной совокупности. Однако что, если вы не готовы предположить нормальное распределение выборочных средних? Тогда вместо приведенных выше

вычислений вы можете использовать бутстреп-анализ:

1. Случайно выбрать 10 наблюдений из выборки с возвратом

значений после каждого выбора. Некоторые наблюдения могут быть выбраны больше одного раза, а некоторые – могут остаться вовсе невыбранными.

2. Вычислить среднее для полученной выборки из 10 значений.

3. Повторить шаги 1 и 2 тысячу раз.

4. Отсортировать тысячу выборочных средних по возрастанию.

5. Найти выборочные средние, которые представляют собой 2.5 и 97.5 процентили. В данном случае 25-е число с начала и с конца. Это границы вашего 95%-го доверительного интервала. Похоже, что в данном случае выборочные средние распределены нормально, поэтому вы немного выиграете от применения бутстреп-анализа. Однако есть много ситуаций, когда такой подход дает преимущество. Что, если вы хотели вычислить доверительные интервалы для выборочной медианы или для разницы между двумя выборочными медианами? Для данного случая нет простой «классической» формулы, так что следует воспользоваться бутстреп-анализом. Если распределение данных неизвестно, если выбросы представляют собой проблему, если размеры выборок малы или если не существует пара-метрического метода, бустреп-анализ часто является полезным способом вычисления доверительных интервалов и проверки гипотез.

**5.2.2 Реализация бустреп-анализа с помощью пакета boot**

Пакет boot предоставляет обширные возможности для бутстреп-анализа и связанных с ним методов создания повторных выборок. Вы можете применить бутстреп-анализ к одной статистике (например, к медиане) или к вектору из статистик (например, к набору регрессионных коэффициентов). Не забудьте скачать и установить пакет boot перед его первым использованием: install.packages(“boot”)

Бутстреп-анализ покажется сложным, но после того, как вы рассмотрите примеры, все должно стать понятным. В общем случае бутстреп-анализ состоит из трех этапов:

1. Напишите функцию, которая вычисляет нужную статистику или нужные статистики. Если имеется одна статистика (например, медиана), функция должна возвращать число. Если есть набор статистик (например, набор регрессионных коэффициентов), функция должна возвращать вектор.

2. Примените функцию boot() к вашей функции, чтобы создать бутстреп-повторности вашей статистики или ваших статистик.

3. Используйте функцию boot.ci(), чтобы вычислить доверительные интервалы для статистики или статистик, созданных на этапе 2.

Теперь перейдем к частностям. Основная функция для бустреп-анализа – это boot(). Формат ее применения таков:

bootobject <- boot(data=, statistic=, R=, ...)

Все параметры функции описаны в табл.5.1.

Таблица 5.1

|  |  |
| --- | --- |
| Параметры | Описание |
| data | Вектор, матрица или таблица данных |
| statistic | Функция, которая создает k статистик, которые будут подвергнуты бутстреп-анализу (k=1, если есть только одна статистика) |
| R | Число бутстреп-выборок |

Функция boot() запускает статистическую функцию R раз. Каждый раз она генерирует набор целых чисел, выбранных случайно с возвратом из диапазона 1:nrow(data). Эти числа используются для создания новой выборки. Статистики вычисляются для новой выборки, а результаты записываются в bootobject. Структура этого объекта описана в табл. 5.2

Таблица 5.2

|  |  |
| --- | --- |
| Элемент | Описание |
| t0 | Наблюдаемые значения k статистик для исходных данных |
| t | Матрица R×k, где каждый ряд – это одна из бутстреп-реализаций k статистик |

Эти элементы можно вызвать как bootobject$t0 и bootobject$t. После создания новых выборок бутстреп-анализом можно изучить результаты при помощи функций print() и plot(). Если результаты выглядят разумно, можно вычислить доверительные интервалы для статистик(и) при помощи функции boot.ci(). Формат применения этой функции таков:

boot.ci(bootobject, conf=, type= )

Параметры описаны в табл. 5.3

Таблица 5.3

|  |  |
| --- | --- |
| Параметры | Описание |
| bootobject | Объект, создаваемый функцией boot() |
| conf | Требуемый доверительный интервал (по умолчанию conf=0.95) |
| type | Тип вычисляемого доверительного интервала. Возможны следующие значения: “norm”, “basic”, “stud”, “perc”, “bca” и “all” (по умолчанию type=”all”) |

Параметр type задает способ вычисления границ доверительного интервала. Метод процентилей (perc) был разобран в примере с выборочным средним. Метод bca рассчитывает интервал, который делает простые поправки на смещение.

**5.2.3 Бустреп-анализ для единичной выборки**

Набор данных mtcars содержит информацию о 32 автомобилях, опубликованную в журнале Motor Trend за 1974 год. Предположим, вы используете множественную регрессию для предсказания расхода топлива по весу машины и по рабочему объему цилиндров двигателя. В дополнение к стандартным регрессионным статистикам вы хотели бы получить 95%-ный доверительный интервал для коэффициента детерминации (доли дисперсии зависимой переменной, которая объясняется независимыми переменными). Доверительный интервал можно получить при помощи непараметрического бутстреп-анализа.

Первая задача – это написание функции для вычисления коэффициента детерминации:

rsq <- function(formula, data, indices) {

d <- data[indices,]

fi t <- lm(formula, data=d)

return(summary(fi t)$r.square)

}

Выражение d <- data[indices,] нужно, чтобы функция boot() могла создавать выборки. Затем вы можете получить большое число повторных бутстреп-выборок (скажем, 1000) при помощи следующих команд:

library(boot)

set.seed(1234)

results <- boot(data=mtcars, statistic=rsq,

R=1000, formula=mpg~wt+disp)

Результаты можно вывести на экран так:

> print(results)

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = mtcars, statistic = rsq, R = 1000, formula = mpg ~ wt + disp)

Bootstrap Statistics :

original bias std. error

t1\* 0.7809306 0.01333670 0.05068926

Можно вычислить 95%-ный доверительный интервал для коэффициента детерминации:

> boot.ci(results, type=c(“perc”, “bca”))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 1000 bootstrap replicates

CALL :

boot.ci(boot.out = results, type = c(“perc”, “bca”))

Intervals :

Level Percentile BCa

95% ( 0.6838, 0.8833 ) ( 0.6344, 0.8549 )

Calculations and Intervals on Original Scale

Some BCa intervals may be unstable

Из этого примера видно, что разные способы вычисления доверительного интервала дают разные результаты. В данном случае два использованных метода дают умеренно различающиеся результаты. В этом разделе мы вычислили границы доверительного интервала для одной статистики. В следующем разделе мы оценим доверительные интервалы для нескольких статистик одновременно.

**5.2.4 Бустреп-анализ для нескольких выборок**

В предыдущем примере бутстреп-анализ был использован для оцен-ки доверительного интервала для одной статистики (коэффициента детерминации). Продолжая этот пример, получим 95%-ные доверительные интервалы для вектора из статистик. А именно рассчитаем доверительные интервалы для трех коэффициентов регрессионной модели (свободный член, вес автомобиля и объем цилиндров). Во-первых, создадим функцию, которая возвращает вектор регрессионных коэффициентов:

bs <- function(formula, data, indices) {

d <- data[indices,]

fi t <- lm(formula, data=d)

return(coef(fit))

}

Затем используем эту функцию для получения 1000 бутстреп-выборок:

library(boot)

set.seed(1234)

results <- boot(data=mtcars, statistic=bs,

R=1000, formula=mpg~wt+disp)

print(results)

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = mtcars, statistic = bs, R = 1000, formula = mpg ~ wt + disp)

Bootstrap Statistics :

original bias std. error

t1\* 34.9606 0.137873 2.48576

t2\* -3.3508 -0.053904 1.17043

t3\* -0.0177 -0.000121 0.00879

При бутстреп-анализе нескольких статистик нужно добавлять параметр индекса в функции plot() и boot.ci(), чтобы указать, какую колонку bootobject$t требуется проанализировать. В этом примере 1 соответствует свободному члену, 2 – весу машины, а 3 – объему цилиндров. Чтобы графически отобразить результаты для веса машины, используйте команду plot(results, index=2).

Чтобы получить 95%-ные доверительные интервалы регрессионных коэффициентов для веса машины и объема двигателя, используйте следующие команды:

> boot.ci(results, type=”bca”, index=2)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 1000 bootstrap replicates

CALL :

boot.ci(boot.out = results, type = “bca”, index = 2)

Intervals :

Level BCa

95% (-5.66, -1.19 )

Calculations and Intervals on Original Scale

> boot.ci(results, type=”bca”, index=3)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 1000 bootstrap replicates

CALL :

boot.ci(boot.out = results, type = “bca”, index = 3)

Intervals :

Level BCa

95% (-0.0331, 0.0010 )

Calculations and Intervals on Original Scale

* 1. **Практические применения методов повторных выборок**

**5.3.1 Подход создания набора для валидации**

Начнем с использования функции sample() для разделения набора наблюдений sample () на две половины, выбрав случайное подмножество из 196 наблюдений исходные 392 наблюдения. Мы называем эти наблюдения обучающим набором.

> library(ISLR)

> set.seed(1)

> train=sample (392,196)

>lm.fit=lm(mpg~horsepower,data=Auto,subset=train)

> attach(Auto)

> > mean((mpg -predict (lm.fit ,Auto))[-train ]^2)

Error: unexpected '>' in ">"

> attach(Auto)

acceleration, cylinders, displacement, horsepower, mpg, name,

origin, weight, year

> mean((mpg-predict (lm.fit ,Auto))[-train ]^2)

[1] 26.14142

Таким образом, оценочный тест MSE для подгонки линейной регрессии равен 26.14. Мы можем использовать функцию poly() для оценки ошибки теста для квадратичной и кубической регрессии

> lm.fit2=lm(mpg~poly(horsepower,2),data=Auto,subset=train)

> mean((mpg-predict (lm.fit2 ,Auto ))[- train]^2)

[1] 19.82259

> lm.fit3=lm(mpg~poly(horsepower ,3),data=Auto , subset=train)

> mean((mpg -predict (lm.fit3 ,Auto ))[- train]^2)

[1] 19.78252

> set.seed(2)

> train=sample (392,196)

> lm.fit=lm(mpg~horsepower ,subset=train)

> mean((mpg-predict (lm.fit ,Auto))[-train ]^2)

[1] 23.29559

> lm.fit2=lm(mpg~poly(horsepower ,2),data=Auto , subset=train)

> mean((mpg-predict (lm.fit2 ,Auto ))[- train]^2)

[1] 18.90124

> lm.fit3=lm(mpg~poly(horsepower ,3),data=Auto , subset=train)

> mean((mpg-predict (lm.fit3 ,Auto ))[- train]^2)

[1] 19.2574

Эти результаты согласуются с нашими предыдущими выводами: модель предсказывает mpg, используя квадратичную функцию лошадиной силы, лучше, чем модель, которая включает только линейную функцию лошадиной силы, и в то же время там мало доказательств в пользу модели, которая использует кубическую функцию лошадиных сил.

**5.3.2. Перекрестная проверка по принципу «leave-one-out»**

Оценка LOOCV может быть автоматически вычислена для любого обобщения линейной модели с использованием glm () и cv-функции glm (). В лабораторных работах для главы 3 мы использовали функцию glm () для выполнения логистической регрессии, передавая в семействе "биномиальный" аргумент. Но если мы используем glm (), чтобы соответствовать модели, не передавая аргумент семейства, он выполняет линейную регрессию, как и функция lm (). Так, например

> glm.fit=glm(mpg~horsepower ,data=Auto)

> coef(glm.fit)

(Intercept) horsepower

39.9358610 -0.1578447

> lm.fit=lm(mpg~horsepower ,data=Auto)

> coef(lm.fit)

(Intercept) horsepower

39.9358610 -0.1578447

> library(boot)

> glm.fit=glm(mpg~horsepower ,data=Auto)

> cv.err=cv.glm(Auto ,glm.fit)

> cv.err$delta

[1] 24.23151 24.23114

Функция cv.glm () создает список из нескольких компонентов. Два числа в дельта-векторе содержат результаты перекрестной проверки. В этом случае, если числа идентичны (до двух десятичных знаков) и соответствуют статистике LOOCV, приведенной в (4.1). Ниже мы обсуждаем ситуацию в которые два числа отличаются. Наша оценка перекрестной проверки для теста ошибка примерно 24,23.

> for (i in 1:5){

+ glm.fit=glm(mpg~poly(horsepower,i),data=Auto)

+ cv.error[i]=cv.glm(Auto,glm.fit)$delta[1]

+ }

> cv.error

[1] 24.23151 19.24821 19.33498 19.42443 19.03321

Мы видим резкое падение оценочного теста MSE между линейной и квадратичной подгонками, но тогда нет явного улучшения от использования полинома высшего порядка

**5.3.3. Перекрестная проверка k-Fold**

Функция cv.glm () также может использоваться для реализации k-кратного CV. Ниже мы используем k = 10, общий выбор для k, в наборе данных Auto. Мы снова поставили random seed и инициализировали вектор, в котором мы будем хранить ошибки CV соответствующей полиномиальной очередности порядков от одного до десяти.

> set.seed(17)

> cv.error.10=rep(0 ,10)

> cv.error.10[i]=cv.glm(Auto ,glm.fit ,K=10) $delta [1]

> for (i in 1:10){

+ glm.fit=glm(mpg~poly(horsepower ,i),data=Auto)

+ cv.error.10[i]=cv.glm(Auto ,glm.fit ,K=10) $delta [1]

+ }

>

> cv.error.10

[1] 24.24309 19.24813 19.30470 19.32010 19.08039 18.99459 19.46065 18.89762

[9] 19.29570 19.63523

Время вычисления намного меньше, чем в предыдущем методе. Из полученных данных видно, что доказательств недостаточно для того, чтобы сказать, что увеличение порядка полинома приводит к более низкой ошибке.

**5.3.4. Bootstrap**

Одним из больших преимуществ подхода начальной загрузки является то, что он может применяется практически во всех ситуациях. Сложные математические расчеты не являются обязательными. Выполнение начального анализа в R влечет за собой только два шага. Во-первых, мы должны создать функцию, которая вычисляет статистику выигрыша. Во-вторых, мы используем функцию boot (), которая является частью библиотеки загрузки, чтобы boot () выполняла загрузку путем многократной выборки наблюдений из данных. комплект с заменой.

> alpha.fn=function (data, index){

+ X=data$X[index]

+ Y=data$Y[index]

+ return ((var(Y)-cov(X,Y))/(var(X)+var(Y) -2\*cov(X,Y)))

+ }

> alpha.fn(Portfolio ,1:100)

[1] 0.5758321

> set.seed(1)

> alpha.fn(Portfolio ,sample (100,100, replace=T))

[1] 0.5963833

> boot(Portfolio ,alpha.fn,R=1000)

Реализовать анализ можно, выполнив команду много раз.

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = Portfolio, statistic = alpha.fn, R = 1000)

Bootstrap Statistics :

original bias std. error

t1\* 0.5758321 -7.315422e-05 0.08861826

В итоге = 0,5758321, SE = 0,08861826

**5.3.5 Оценка точности модели линейной регрессии**

Метод начальной загрузки может быть использован для оценки изменчивости оценок коэффициентов и прогнозов из статистического метода обучения. Мы используем подход начальной загрузки, чтобы оценить изменчивость оценки для β0 и β1, условия пересечения и наклона для модели линейной регрессии, которая использует мощность для прогнозирования миль на галлон в наборе данных Auto. Мы будем сравнивать оценки, полученные с помощью бутстрепа, с полученными при использовании формулы для SE (βˆ0) и SE (βˆ1)

> boot.fn=function (data ,index)

+ return(coef(lm(mpg~horsepower ,data=data , subset=index)))

> boot.fn(Auto ,1:392)

(Intercept) horsepower

39.9358610 -0.1578447

> set.seed(1)

> boot.fn(Auto ,sample (392,392, replace=T))

(Intercept) horsepower

38.7387134 -0.1481952

> boot.fn(Auto ,sample (392,392, replace=T))

(Intercept) horsepower

40.0383086 -0.1596104

> boot(Auto ,boot.fn ,1000)

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = Auto, statistic = boot.fn, R = 1000)

Bootstrap Statistics :

original bias std. error

t1\* 39.9358610 0.02972191 0.860007896

t2\* -0.1578447 -0.00030823 0.007404467

Отсюда мы видим, что SE(β0) = 0, 8607896 , SE(β1) = 0,007404467

> summary (lm(mpg∼horsepower ,data=Auto))$coef

Error: unexpected input in "summary (lm(mpg\"

> summary (lm(mpg~horsepower ,data=Auto))$coef

Estimate Std. Error t value Pr(>|t|)

(Intercept) 39.9358610 0.717498656 55.65984 1.220362e-187

horsepower -0.1578447 0.006445501 -24.48914 7.031989e-81

Ниже мы вычисляем оценки стандартных ошибок bootstrap и стандартные оценки линейной регрессии, полученные в результате подгонки квадратичной модели к данным. Поскольку данная модель обеспечивает хорошую подгонку к данным теперь существует лучшее соответствие между оценками bootstrap и стандарт оценки SE(βˆ0), SE(βˆ1) и SE(βˆ2).

> boot.fn=function (data ,index)

+ coefficients(lm(mpg~horsepower +I(horsepower ^2),data=data ,

+ subset=index))

> set.seed(1)

> boot(Auto ,boot.fn ,1000)

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = Auto, statistic = boot.fn, R = 1000)

Bootstrap Statistics :

original bias std. error

t1\* 56.900099702 6.098115e-03 2.0944855842

t2\* -0.466189630 -1.777108e-04 0.0334123802

t3\* 0.001230536 1.324315e-06 0.0001208339

> summary (lm(mpg~horsepower +I(horsepower ^2),data=Auto))$coef

Estimate Std. Error t value Pr(>|t|)

(Intercept) 56.900099702 1.8004268063 31.60367 1.740911e-109

horsepower -0.466189630 0.0311246171 -14.97816 2.289429e-40

I(horsepower^2) 0.001230536 0.0001220759 10.08009 2.196340e-21

# Рекомендуемая литература

1. R: A Language and Environment for Statistical Computing. Reference Index. The R Development Core Team. Version 2.4.1 (2006-12-18), 2580 pages.
2. Дьяконов А.Г. Анализ данных, обучение по прецедентам, логические игры, системы WEKA, RapidMiner и MatLab. Учебное пособие. ВМК МГУ. 2010.
3. Hastie T., Tibshirani R. and Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer. Second Edition. 2009.
4. James G., Witten D., Hastie T. and Tibshirani R. An Introduction to Statistical Learning with Applications in R. Springer. 2013.
5. Кабаков Р. И. R в действии. Анализ и визуализация данных в программе R /пер. с англ. Полины А. Волковой. – М.: ДМК Пресс, 2014. – 588 с.: ил.
6. Venables W. and Ripley B. Modern Applied Statistics with S. Fourth Edition. Springer, New York, 2002.
7. Вьюгин В.В. Математические основы теории машинного обучения и прогнозирования. М.: 2013. - 387 с.
8. Мерков А.Б. Введение в методы статистического обучения, версия: 25 декабря 2014. – 260 c.
9. Ефимов Д.А. и Никулин В.Н. Предсказание биологического состояния молекул исходя из их химических свойств. Advanced Science, Вятский Государственный Университет. Том 2(2), сс. 107-123, 2013.
10. Никулин В.Н. и Прозорова Т.Г. Два алгоритма на основе техники стохастического градиентного спуска для рекомендательных систем. Вестник Пермского Университета: Математика-Механика-Информатика. Том 3(26), 2014, pp. 48-56, 2014.
11. Никулин В.Н., Палешева С.А. и Зубарева Д.С. Об однородных ансамблях при использовании метода бустинга в приложении к классификации несбалансированных данных. Вестник Пермского Университета. Экономика. Том 4(15), сс. 8-14, 2012.
12. Hinton G. and Salakhutdinov R. Reducing the dimensionality of data with neural networks. Science, Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.
13. Ciresan D., Meier U., Gambardella L., Schmidhuber J. Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition Neural Computation, Volume 22, Number 12, December 2010, pp.3207–3220.
14. Кабаков Р. R в действии. Анализ и визуализация данных на языке R. ДМК Пресс. 2011. – 472 с.
15. Мастицкий С.Э., Шитиков В.К. Статистический анализ и визуализация данных с помощью R. – Электронная книга. 2014
16. Andrieu C., De Freitas N. and Doucet A. and Jordan M. An Introduction to MCMC for Machine Learning. Machine Learning, 50, 5–43, 2003.
17. Freudenthaler C., Schmidt-Thieme L. and Rendle S. Bayesian Factorization Machines. In Workshop on Sparse Representation and Low-rank Approximation, Neural Information Processing Systems (NIPS-2011), Granada, Spain.
18. Andrieu C. and Thoms J. A tutorial on adaptive MCMC. Stat Comput (2008) 18: 343–373.
19. The R Journal [Электронный ресурс]: Электрон. журн. – Режим доступа к журн.: <http://journal.r-project.org/index.html>

# Приложения

Приложение 1 Алг. 9. Фрактал Мандельброта

pr1(-2.2,1.0,-1.2,1.2,300,50,0.25) #\*\*\* вызов функции

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

pr1 <- function(x1,x2,y1,y2,n,m,alfa){

A = matrix(0,n,n)

u = c(1:n)

v = c(1:n)

dx = (x2-x1)/n

dy = (y2-y1)/n

a = x1

for(i in 1:n){

b = y1

for(j in 1:n){

at = 0

bt = 0

k = 0

Q = 0.0

while(Q < 100000 & k < m){

at = at\*at - bt\*bt + a

bt = 2.0\*at\*bt + b

Q = at\*at + bt\*bt

k = k + 1

}

A[i,j] =Q^alfa

v[j] = b

b = b + dy

}

u[i] = a

a = a + dx

}

win.graph()

image(u, v, A, col=terrain.colors(100))

}

Приложение 2. Алг. 10. Скорость вычислений

tst <- function(n){

#Зададим количество строк в матрице и создадим её

nc = 30

Z = matrix(0,nc,3)

#Выведем текущее время и дату

ptm <- date()

print(ptm)

#Создадим массив случайных 5000000 чисел

nr = 5000000

u = sample(1:20,nr,replace=TRUE)

v = matrix(0,nr,1)

#Выведем текущее время и дату

ptm1 <- date()

print(ptm1)

#Выполним алгоритм по заполнению матрицы Z с некоторыми условиями с помощью синтаксиса языка R:

delta = 7

for(i in 1:nr){

if(u[i] > delta){v[i] = 1}

else{v[i] = 0}

}

#Выведем текущее время и дату

ptm2 <- date()

print(ptm2)

for(i in 1:nc){

Z[i,1] = u[i]

}

Z[1:nc,2] = v[1:nc]

v = matrix(0,nr,1)

#Теперь выполним этот же алгоритм с помощью встроенной функции, которая реализована на C

v[which(u > delta)] = 1

v[which(u <= delta)] = 0

Z[1:nc,3] = v[1:nc]

#Вновь выведем текущую дата после выполнения алгоритма и выведем полученную матрицу

ptm3 <- date()

print(ptm3)

print(Z)

}

Приложение 3. Алг. 11. Метод случайных сбалансированных подмножеств

gbm1 <- function(n){

B = rbind(A,V)

nrs = 10

alfa = 0.85

nr = nrow(A)

nt = nrow(T)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

i1 = 0

i2 = 0

for(i in 1:nr){

if(B[i,1]==0){i1 = i1 + 1}

else{i2=i2+1}

}

ind = matrix(0,nr,1)

fr = i2/i1

sol\_pass = matrix(0,nr,2)

sol\_tst = matrix(0,nt,1)

trace = matrix(0,nrs,4)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

for(jj in 1:nrs){

ii = 0

for(i in 1:nr){

vs = runif(1)

if(B[i,1]==1 && vs<=alfa){

ii = ii + 1

ind[i,1] = 1

}

else{

if(B[i,1]==0 && vs <=fr){

ii = ii + 1

ind[i,1]=1

}

else{ind[i,1]=0}

}

}

object = gbm(tg~.,data=B[ind==1,],distribution="adaboost",n.trees=100,

shrinkage=0.01,interaction.depth=10,n.minobsinnode=7)

tr = predict(object,newdata=B,n.trees=100,type="response")

sol\_tst = sol\_tst + predict(object,newdata=T,n.trees=100,type="response")

k = 0

k1 = 0

for(i in 1:nr){

if(ind[i,1] == 1){k = k + 1}

else{

sol\_pass[i,1] = sol\_pass[i,1] + 1

sol\_pass[i,2] = sol\_pass[i,2] + tr[i]

k1 = k1 + 1

}

}

m = 0

for(i in 1:nr){

if(sol\_pass[i,1] >= 1){m = m + 1}

}

sol\_val = matrix(0,m,1)

y = matrix(0,m,1)

m = 0

for(i in 1:nr){

if(sol\_pass[i,1] == 1){

m = m + 1

y[m] = B[i,1]

sol\_val[m,1] = sol\_pass[i,2]

}

else{

if(sol\_pass[i,1] >= 2){

m = m + 1

y[m] = B[i,1]

sol\_val[m] = sol\_pass[i,2]/sol\_pass[i,1]

}

}

}

trace[jj,1] = jj

trace[jj,2] = k

trace[jj,3] = m

trace[jj,4] = auc(y,sol\_val)

print(trace[jj,])

}

sol\_tst = sol\_tst/nrs

write.table(sol\_tst, file = "c:/contest/teaching/output/solut\_vrs10\_tst\_d140314.txt", quote=FALSE, row.names=FALSE, col.names=FALSE, sep = " ")

write.table(sol\_pass, file = "c:/contest/teaching/output/solut\_vrs10\_pass\_d140314.txt", quote=FALSE, row.names=FALSE, col.names=FALSE, sep = " ")

write.table(trace, file = "c:/contest/teaching/output/table\_vrs10\_d140314.txt", quote=FALSE, row.names=FALSE, col.names=FALSE, sep = " ")

}

Приложение 4. Алг. 12. Сумма и произведение переменных

nc = ncol(T)

na = nc + 7

nb = na + 1

B1 = matrix(0,nr,nb)

T1 = matrix(0,nt,na)

B2 <- data.frame(B1)

T2 <- data.frame(T1)

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

names(B2)[1] = "tg"

for(j in 1:na){

j2 = j + 1

names(B2)[j2] = paste("v",j)

names(T2)[j] = paste("v",j)

}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

alfa = 0.85

delta = 20

B2[,1] = B[,4]

B2[,2] = B[,8]

B2[,3] = B[,10]

T2[,1] = T[,3]

T2[,2] = T[,7]

T2[,3] = T[,9]

print("step N1")

for(j in 1:3){

B2[which(B2[,j]>delta),j] = delta

T2[which(T2[,j]>delta),j] = delta

}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

B2[,4] = B2[,1]+B2[,2]+B2[,3] + 0.001

T2[,4] = T2[,1]+T2[,2]+T2[,3] + 0.001

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

for(j in 5:7){

j4 = j - 4

B2[,j] = B2[,j4]/B2[,4]

T2[,j] = T2[,j4]/T2[,4]

}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

print("step N2")

nc2 = nc + 2

B2[,nc2] = B2[,5]+B2[,6]

nc3 = nc + 3

B2[,nc3] = B2[,5]+B2[,7]

nc4 = nc + 4

B2[,nc4] = B2[,6]+B2[,7]

nc5 = nc + 5

B2[,nc5] = B2[,5]\*B2[,6]

nc6 = nc + 6

B2[,nc6] = B2[,5]\*B2[,7]

nc7 = nc + 7

B2[,nc7] = B2[,6]\*B2[,7]

nc8 = nc + 8

B2[,nc8] = B2[,5]\*B2[,6]\*B2[,7]

nc1 = nc + 1

B2[,1:nc1] = B

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

T2[,nc1] = T2[,5]+T2[,6]

T2[,nc2] = T2[,5]+T2[,7]

T2[,nc3] = T2[,6]+T2[,7]

T2[,nc4] = T2[,5]\*T2[,6]

T2[,nc5] = T2[,5]\*T2[,7]

T2[,nc6] = T2[,6]\*T2[,7]

T2[,nc7] = T2[,5]\*T2[,6]\*T2[,7]

T2[,1:nc] = T

Приложение 5. Алг. 13. Аппроксимирующие и разделяющие функции потерь

pr1 <- function(n){

y<-c(-1,-1,-1,-1,1,1,1)

x<-c(-0.23,-0.98,-0.54,-0.1,0.35,0.56,0.72)

trace=matrix(0,1,2)

n=7

S=0.0

for(i in 1:n){

S=S+(y[i]-x[i])^2.0

}

S=sqrt(S/n)

trace[1,1]=S

S=0.0

for(i in 1:n){

S=S+exp(-y[i]\*x[i])

}

S=S/n

trace[1,2]=S

print(trace[1,])

x=10.0\*x

S=0.0

for(i in 1:n){

S=S+(y[i]-x[i])^2.0

}

S=sqrt(S/n)

trace[1,1]=S

S=0.0

for(i in 1:n){

S=S+exp(-y[i]\*x[i])

}

S=S/n

trace[1,2]=S

print(trace[1,])

}

Приложение 6. Алг. 14. Метод опорных векторов

svm1 <- function(n){

nc = ncol(X)

for(i in 1:nc){

q=max(X[,i])+1.0

X[,i]=X[,i]/q

T[,i]=T[,i]/q

}

X <- cbind(y,X)

X[is.na(X)] <- 0

T[is.na(T)] <- 0

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

names(X)[1] <- "tg"

for(i in 1:nc){

i1=i+1

names(X)[i1] <- paste("v",i,sep="")

names(T)[i] <- paste("v",i,sep="")

}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

m = 10

nr = nrow(X)

nt = nrow(T)

R <- matrix(0,nt,10)

trace = matrix(0,m,2)

Z = 0.0

for(i in 1:m){

j = i-1

X[,1] <- matrix(0,nr,1)

X[which(y[,1]==j),1] = 1

zt <- matrix(0,nt,1)

zt[which(z[,1]==j),1] = 1

zt <- data.frame(zt)

object <- svm(tg~., data=X)

R[,i] = predict(object, newdata=T)

trace[i,1] = i

S = auc(zt[,1], R[,i])

trace[i,2] = S

Z = Z + S

print(trace[i,])

}

Z = Z/m

print(Z)

h = matrix(0,nt,1)

S=0

for(i in 1:nt){

q=-1

for(j in 1:m){

if(R[i,j]>q){

q=R[i,j]

k=j-1

}

}

h[i]=k

if(k==z[i,1]){

S=S+1

}

}

r=S/nt

print(r)

write.table(h, file = "c:/contest/Kaggle/MNIST/svm/tst\_svm\_d271114.txt", quote=FALSE, row.names=FALSE, col.names=FALSE, sep = " ")

write.table(trace, file = "c:/contest/Kaggle/MNIST/svm/trace\_svm\_d271114.txt", quote=FALSE, row.names=FALSE, col.names=FALSE, sep = " ")

}

Приложение 7. Алг. 15. Случайный лес – RandomForest

rf1 <- function(n){

nc = ncol(X)

for(i in 1:nc){

q=max(X[,i])+1.0

X[,i]=X[,i]/q

T[,i]=T[,i]/q

}

X <- cbind(y,X)

X[is.na(X)] <- 0

T[is.na(T)] <- 0

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

names(X)[1] <- "tg"

for(i in 1:nc){

i1=i+1

names(X)[i1] <- paste("v",i,sep="")

names(T)[i] <- paste("v",i,sep="")

}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

m = 10

nr = nrow(X)

nt = nrow(T)

R <- matrix(0,nt,10)

trace = matrix(0,m,2)

Z = 0.0

for(i in 1:m){

j = i-1

X[,1] <- matrix(0,nr,1)

X[which(y[,1]==j),1] = 1

zt <- matrix(0,nt,1)

zt[which(z[,1]==j),1] = 1

zt <- data.frame(zt)

object <- randomForest(tg~.,data=X,do.trace=TRUE,ntree=100,mtry=28,nodesize=5)

R[,i] = predict(object, newdata=T)

trace[i,1] = i

S = auc(zt[,1], R[,i])

trace[i,2] = S

Z = Z + S

print(trace[i,])

}

Z = Z/m

print(Z)

h = matrix(0,nt,1)

S=0

for(i in 1:nt){

q=-1

for(j in 1:m){

if(R[i,j]>q){

q=R[i,j]

k=j-1

}

}

h[i]=k

if(k==z[i,1]){

S=S+1

}

}

r=S/nt

print(r)

write.table(h, file = "c:/contest/Kaggle/MNIST/rf/tst\_rf\_d271114.txt", quote=FALSE, row.names=FALSE, col.names=FALSE, sep = " ")

write.table(trace, file = "c:/contest/Kaggle/MNIST/rf/trace\_rf\_d271114.txt", quote=FALSE, row.names=FALSE, col.names=FALSE, sep = " ")

}

1. https://www.kaggle.com/ [↑](#footnote-ref-1)
2. http://cran.r-project.org [↑](#footnote-ref-2)
3. www.kaggle.com [↑](#footnote-ref-3)