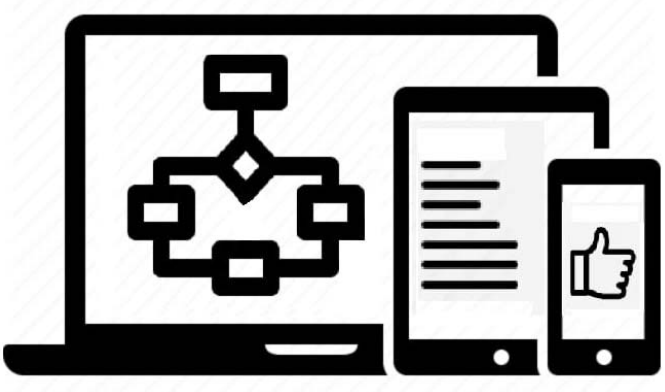
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

БАЗЫ ЗНАНИЙ И ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ В СИСТЕМАХ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Методические рекомендации к лабораторным работам для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» дневной и заочной форм обучения



Могилев 2024

УДК 004.8
БКК 32.973.26
Б17

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «26» апреля 2024 г., протокол № 10

Составитель канд. физ.-мат. наук В. А. Ливинская

Рецензент канд. техн. наук А. П. Прудников

Методические рекомендации предназначены к выполнению лабораторных работ для студентов специальности 1-40 05 01 «Информационные системы и технологии (по направлениям)» дневной и заочной форм обучения. Содержат цель, методические указания, задания для самостоятельного выполнения, контрольные вопросы.

Учебное издание

БАЗЫ ЗНАНИЙ И ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ В СИСТЕМАХ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Ответственный за выпуск	В. В. Кутузов
Корректор	А. А. Подошевко
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 16 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

Содержание

Введение

Выполнение лабораторной работы по дисциплине предполагает:

– ознакомление с теоретическим материалом по теме работы;

– выполнение работы согласно выданному индивидуальному заданию;

– оформление отчета в Word и размещение его вместе с файлами, содержащими код на языке программирования и набор данных, подвергающихся исследованию описанным методом в системе Moodle по расписанию;

– защита выполненной работы в аудитории.

При приёме защиты лабораторной работы преподаватель проверяет:

1) наличие исходного кода программы, реализующего требуемый метод анализа данных;

2) правильность работы программы для заданных наборов данных;

3) понимание студентом сути проведённых исследований и правильность сделанных выводов;

4) знание студентом ответов на контрольные вопросы;

5) наличие отчёта, оформленного в соответствии с требованиями к учебным текстовым документам.

Отчёт о выполнении лабораторной работы содержит следующее.

1 Титульный лист с указанием названия и автора работы, а также преподавателя, проверяющего работу.

2 Задание на лабораторную работу.

3 Краткое описание теоретической части работы, включающее некоторые формулы и утверждения, лежащие в основе исследуемого метода.

4 Описание исходных данных для работы, их природы и происхождения.

5 Полученные результаты, включая их графическое представление в форме таблиц, диаграмм и рисунков.

6 Заключение, включающее выводы, сделанные по результатам работы.

Лабораторная работа № 1. Формирование базы знаний с помощью различных техник

Цель: разработка программного обеспечения (ПО) для формирования базы знаний, основанной на опросе экспертов и с использованием таблицы решений.

Теоретические сведения

Выявление знаний от экспертов [1, с. 21–23].
Таблица решений [1, с. 24–27].

Задания

1 В соответствии с выданным заданием разработать ПО для экспертного оценивания.
2 Одним из типовых методов провести оценивание степени влияния объектов.
3 Провести оценку достоверности экспертизы.
4 В соответствии с выданным заданием разработать ПО для формирования таблицы решений.
5 С помощью множества предикатов описать исходную и конечную ситуации.
6 Разработать процедуру перехода из начального в конечное состояние.

Контрольные вопросы

1 Что такое экспертное оценивание, для чего оно необходимо?
2 Что включает в себя процедура сравнения?
3 Дайте определение эмпирической системы.
4 Какие существуют методы для измерения степени влияния объектов?
5 В чем заключается процедура ранжирования объектов?
6 Что такое непосредственная оценка объектов?
7 Назовите основные характеристики экспертов.
8 Какие виды опросов используются при коллективной экспертизе?
9 В чем заключается основная идея таблицы решений?
10 Назовите основное достоинство алгоритма поиска решений.
11 В чем заключается недостаток алгоритма поиска решений?
12 Для чего применяется система STRIPS?
13 Что является основной задачей системы STRIPS?

Лабораторная работа № 2. Знакомство со средой для интеллектуального анализа данных R

Цель: ознакомиться с интерфейсом R-Studio; научиться работать в режиме консоли и путем написания скриптов, а также подключать внешние пакеты; изучить основные методы обработки статистических данных.

Теоретические сведения

Начало работы и получение справочной информации [2, с. 19–34].

Для установки среды R и R-studio на свой домашний компьютер воспользуйтесь инструкцией по ссылке https://bdemeshev.github.io/installation/r/R_installation.html.

Директория – это место (папка), где находится ваш «проект». То есть там лежит скрипт, данные, картинки и прочее.

Зачастую, у каждого проекта своя директория, поэтому приходится часто их менять. Есть несколько способов это сделать.

1 Использовать функцию setwd("~/Desktop/R"), где в кавычках можно прописать путь к директории.

2 На панели R нажать кнопку Session и в выпадающем списке выбрать Set Working Directory. После этого можно выбрать нужную папку.

3 В нижнем правом окошке есть вся файловая система компьютера, где также можно установить нужную директорию. Нажав кнопку More, можно перейти в текущую директорию или установить новую.

С целью дальнейшего упрощения работы рекомендуется заранее установить некоторые основные пакеты с помощью функции install.packages, набрав команду: install.packages(«название пакета»), или, войдя во вкладку Packages/Install Packages, затем подключить нужный пакет.

Для работы чаще всего используются следующие пакеты:

psych – описательная статистика;

ggplot2 – графика;

lmtest – тестирование гипотез при построении линейных моделей;

MASS – поиск подходящих распределений;

dplyr – вычислительные операции с данными.

После загрузки всех необходимых пакетов нужно узнать в какой директории вы находитесь. Это можно сделать с помощью функцию getwd().

Импорт данных – это загрузка в среду R различных данных для последующей работы с ними. Важно понимать откуда мы берём данные. Обычно источником импорта выступают: обычные текстовые файлы (.csv, .txt), Excel-файлы (.xlsx, .xls), базы данных (SQL), интернет, статистические пакеты (SPSS, SAS, STATA).

Импорт файлов с расширениями csv, txt осуществляется с помощью функции read.csv(). Формат csv самый распространенный формат хранения данных в

мире для анализа данных. Его можно получить, например, сохранив Excel-файл в формате csv (разделитель – запятая).

Проверка того, что все данные импортировались нормально, осуществляется с помощью функции str().

Экспорт данных – это сохранение данных с требуемым расширением на свой компьютер. Для экспорта данных применяют функции: write.csv(df, "table_car.csv"), или write.xlsx(df, "table_car.xlsx"), или write.html(df, "table_car.html"), где df – имя фрейма данных в среде R.

Задания

1 Инсталлировать пакеты для ввода данных в различных форматах.

2 Изучить возможность получать справочную информацию по необходимым пакетам.

3 Создать свою директорию.

4 Загрузить данные для своего варианта, полученные у преподавателя, в переменную.

5 Получить справочную информацию по своим данным, просмотреть их содержимое.

6 Проверить, есть ли среди данных пропуски, удалить с помощью соответствующей команды.

7 Изучить типы данных.

8 Выполнить экспорт очищенных данных в MS Excel.

9 Составить отчет.

Контрольные вопросы

1 Какие типы данных используются в среде R?

2 Как начать работу в R-Studio?

3 Как получить справочную информацию по пакетам и встроенным в среду R датасетам?

4 Как импортировать в среду R файлы с различными расширениями?

5 Как экспортировать из среды R файлы с различными расширениями?

Лабораторная работа № 3. Разведывательный анализ данных средствами R

Цель: провести простой анализ данных, состоящий из описательной статистики и визуализации.

Теоретические сведения

Разведочный анализ данных основан на построении визуализаций и вычислении характеристик описательной статистики, представленных в таблице 1.

Таблица 1 – Параметры описательной статистики

Функция	Описание
mean()	Среднее значение
median()	Медиана
var()	Дисперсия
sd()	Стандартное отклонение
min()	Минимальное значение
max()	Максимальное значение
quantile()	Квантили (по умолчанию рассчитывается максимальное и минимальное значения, а также квартили)
summary()	Сводка по параметрам описательной статистики для всех переменных набора

Среди данных могут быть пропуски, которые не позволят рассчитать статистические параметры для столбцов количественных переменных с пропусками без специальных настроек.

Функция is.na() возвращает набор данных (вектор, таблицу и т. п.) заполненный значениями true (если значение отсутствует) и false (при его наличии). Так как true маркируется 1, а false маркируется нулем, то функция sum(), примененная к такому преобразованному набору данных, дает число пропусков.

Команда Glimpse(df) (данная функция из ранее уставленного пакета «dplyr») позволяет посмотреть краткие сведения о наборе данных df. Вывод на экран описательной статистики осуществляется также с помощью функции «describe», относящейся к пакету «psych».

Задания

1 Импортировать данные, полученные у преподавателя, согласно своему варианту.

2 Создать новую переменную-вектор, в которой будут 1, если значение в исходном векторе больше среднего, и -1, если значение переменной меньше среднего, и 0, если значение равно среднему с команды ifelse.

3 Вывести описательную статистику.

4 Оформить отчет.

Контрольные вопросы

1 Как создать количественную переменную в среде R?
2 Как создать вектор качественных переменных в среде R?
3 Как рассчитать показатели описательной статистики?
4 Как построить гистограмму и график плотности распределения в среде R?
5 Какие типы данных вам известны?
6 Что такое генеральная совокупность? Что такое выборочная совокупность?
7 Что такое распределение случайной величины?
8 Методы предобработки данных (удаление дубликатов, обработка пропусков, выявление аномальных наблюдений).
9 Как визуализировать распределение количественных данных в R?
10 Как визуализировать распределение категориальных данных в R?
11 Перечислите числовые характеристики выборочной совокупности.
12 Какие параметры описывают нормальное распределение?
13 Что такое медиана и процентили, квантили, мода?
14 Что такое выбросы? Как определить их наличие в данных визуально в R?

Лабораторная работа № 4. Визуализация данных средствами R

Цель: освоить основные распространенные типы графиков в R, приобрести основные навыки работы с пакетом ggplot2 в R.

Теоретические сведения

Особенность R – возможность создания качественной графики с широким спектром графических возможностей R. На сайте R Graph Gallery (https://www.r-graph-gallery.com/all-graphs) представлены не только примеры всевозможных графиков, но и исходный R-код, написанный для их построения.

Для построения графиков используют высокоуровневые функции, доступные без подключения специализированных пакетов для построения визуализаций (таблица 2).

Таблица 2 – Функции для построения графиков

Функция	Описание
plot()	Общая функция для построения графиков. В зависимости от передаваемых данных и настройки параметров, определяемая ей визуализация может быть и точечным графиком, и столбчатой диаграммой
boxplot()	График «ящик с усами»
hist()	Гистограмма
barplot()	Столбчатая диаграмма
scatterplot()	Диаграмма рассеяния (точечная)
pie()	Круговая диаграмма

Функция plot() – функция общего назначения, которая используется для построения графиков. С помощью plot() можно создавать диаграммы рассеяния, точечные графики с гладкими линиями и маркерами, точечные графики с отрезками линий и маркерами и др. Новая диаграмма, которая создается при помощи команды высокого уровня, обычно заменяет предыдущую диаграмму. Для создания ещѐ одной диаграммы, сохранив предыдущую, в стандартной графической оболочке RGui можно открыть новое графическое устройство функцией dev. new(). Каждая новая диаграмма будет появляться в последнем открытом окне. Можно использовать функции dev.next(), dev.prev(), dev.set() и dev.off() для одновременного открытия нескольких окон графики, выбора необходимой диаграммы и закрытия окон. В R очень легко объединить несколько диаграмм в одну общую, используя функции par() или layout(). Имеется возможность добавить графический параметр mfrow=c(nrows, ncols) в функцию par() для создания матрицы из диаграмм размером nrows×ncols, которая будет заполнена по рядам. Для заполнения этой матрицы по столбцам нужно использовать параметр mfcol=c(nrows, ncols). Например, следующий программный код позволяет создать две диаграммы и расположить их в две строки и один столбец (рисунок 1):

```
library(car)
attach(mtcars)
 opar <- par(no.readonly=TRUE)
 par(mfcol=c(2,1))
 plot(wt,mpg, main = "Диаграмма рассеяния для расхода топлива и
веса машины")
 plot(wt,disp, main="Диаграмма рассеяния для объѐма двигателя и
веса машины")
 par(opar)
 detach(mtcars)
```

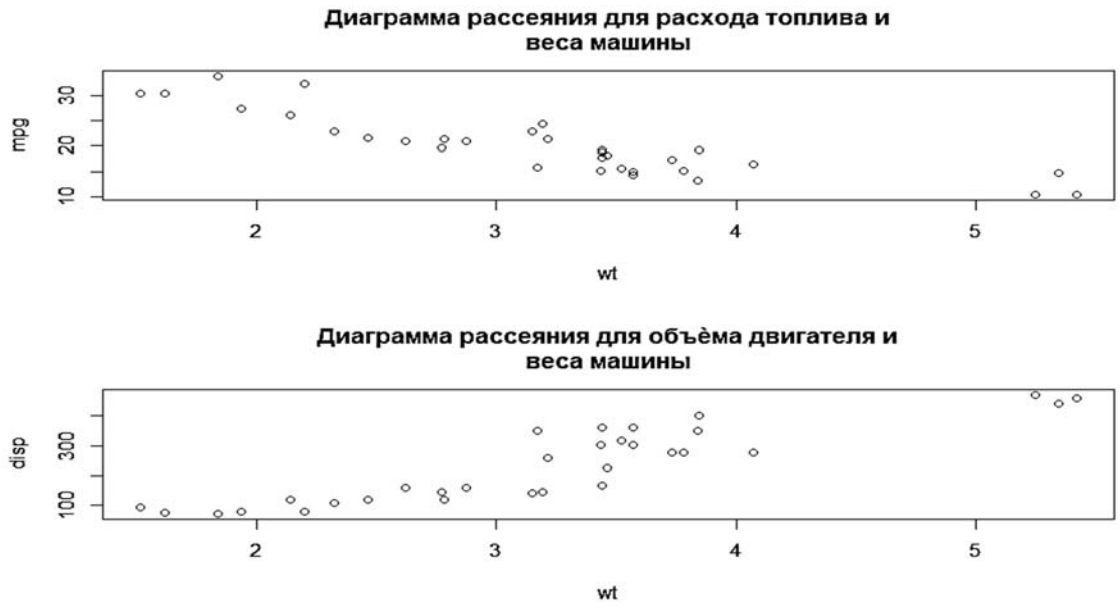


Рисунок 1 – Точечные графики по встроенному набору данных mtcars

Параметры функции plot:

type – тип графика ("p" – точки (по умолчанию);

"l" – линия, "b"– точки и отрезки;

"o"– точки и линия, "n" – ничего не рисуется);

col – цвет ("red", "blue", "green", "cyan", "magenta", "black" и др.).

Также можно определить цвет с помощью функции rgb(r,g,b), аргументы которой могут иметь значения в диапазоне от 0 до 1.

xlab, ylab – названия осей абсцисс и ординат. xlim, ylim – векторы из двух элементов, определяющие размеры диапазонов по x и y.

Если второй элемент вектора меньше первого, то ось меняет направление.

main, sub – основное и дополнительное названия (вверху и внизу рисунка).

lty – тип линии ("blank" – нет линии, "solid" – сплошная, "dashed" – пунктирная, "dotted" – точки, "dotdash" – штрихпунктирная, "longdash" – длинные штрихи, "twodash" – короткие и длинные штрихи).

lwd – толщина линии.

pch – вид точек (0 – квадратики, 1 – кружки, 2 – треугольники, 3 – крестики и т. д. до 24).

log – задание логарифмического масштаба для указанной оси ("x", "y" или "xy"). asp – число, задающее пропорции окна (y/x).

Для построения **столбчатой диаграммы** Bar Graph воспользуемся barplot() и передаем ему вектор значений и вектор надписей.

Гистограммы рисуются с помощью функции hist (). Параметры идентичны функции plot(). В качестве примера создадим нормально распределенную совокупность X из 100 наблюдений со средним значением 0 и стандартным отклонением 1:

```
X <- rnorm (100) # N(0, 1)
```

Строим гистограмму совокупности функцией hist (рисунок 2).

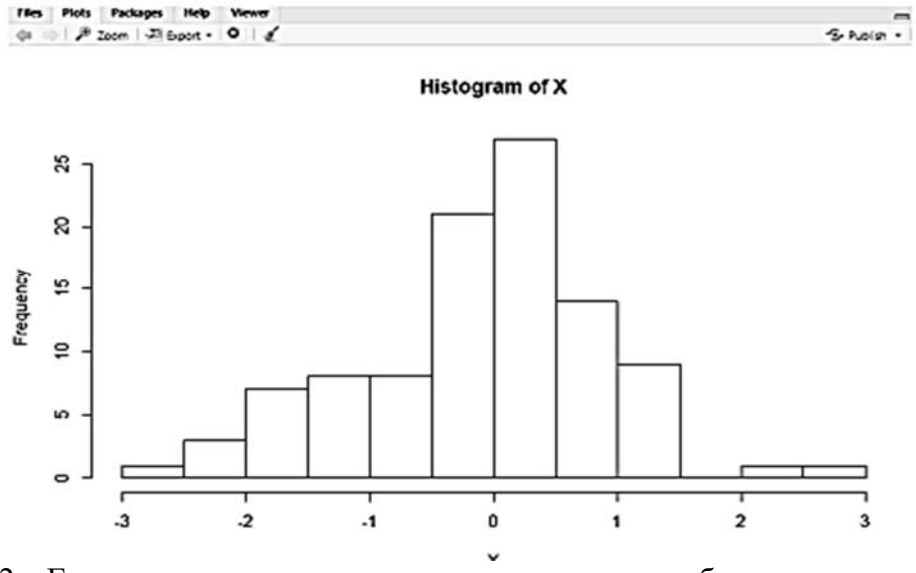


Рисунок 2 – Гистограмма совокупности с автоматически выбранными настройками

Для детального изучения совокупности можно увеличить количество столбцов аргументом breaks. При необходимости залить столбцы цветом аргументом col:

```
hist(X, breaks = 20,col = "blue")
```

Для построения гистограммы с помощью библиотеки ggplot2 воспользоваться [3].

Задания

1 Импортировать данные, полученные у преподавателя, согласно своему варианту.

2 Изучить возможности библиотеки ggplot2.

3 С помощью базового пакета и библиотеки ggplot2 выполнить графические представления, позволяющие проиллюстрировать неоднородность исследуемого набора данных. Построение желательно произвести по нескольким проекциям.

4 Оформить отчет.

Контрольные вопросы

1 Как изменить подпись к легенде?
2 Как изменить подписи к осям координат?
3 Каким образом производится разбивка на фасеты?
4 Каким образом меняется тема оформления графика?
5 Каким образом добавить название к графику?

Лабораторная работа № 5. Проверка гипотез

Цель: научиться формулировать статистические гипотезы и оценивать вероятности ошибок принятия гипотезы, проверять статистические гипотезы согласно алгоритму, делать выводы по результатам статистической проверки.

Теоретические сведения

Для освоения навыков проверки статистических гипотез в среде R воспользоваться материалом [3, с. 231–246; 4, с. 151–176].

Алгоритм проверки статистических гипотез

1 По выборочным данным формулируют основную H_0 и альтернативную H_1 гипотезы.

2 Задают уровень значимости α (0,05 или 0,01).

3 В зависимости от H_0 определяют статистический критерий K, имеющий известное распределение.

4 По выборке и формуле критерия K рассчитывают наблюдаемое значение

критерия $K_{набл}$.

5 В зависимости от вида H_1 определяют вид критической области W и критические точки по соответствующим таблицам для распределения критерия K.

6 По результатам проверки принадлежности $K_{набл}$ к критической области делают вывод о принятии или отклонении гипотезы H_0. Формулируют общий вывод, исходя из поставленной задачи.

В зависимости от имеющейся у исследователя информации, гипотезы классифицируются следующим образом (таблицы 3 и 4).

Таблица 3 – Функции для тестирования параметрических гипотез

Функция	Описание
Одновыборочные критерии о равенстве числовому параметру	
t. test (x, μ_0)	математического ожидания
Критерии о равенстве числовых характеристик	
var.test (x, y)	дисперсий двух групп
t. test (x, y)	математических ожиданий двух групп
prop.test(x, y)	долей двух групп
aov()	математических ожиданий в нескольких группах
bartlett.test ()	дисперсий в нескольких группах

Таблица 4 – Функции для тестирования непараметрических гипотез

Функция	Описание
Проверка гипотезы о законе распределения	
shapiro.test(x)	о согласии с нормальным законом распределения
Критерии об однородности	
wilcox.test(x,y,paired=F)	критерий Манна – Уитни для двух независимых выборок
wilcox.test(x,y,paired=T)	критерий Уилкоксона для двух зависимых выборок
kruskal.test()	критерий Краслелла Уоллеса о равенстве распределений в нескольких группах.

Задания

1 Получить данные у преподавателя согласно своему варианту.

2 Описать исходную совокупность с помощью показателей дескриптивной статистики.

3 Построить гистограммы для количественных признаков и ящик с усами для исходного набора данных и выдвинуть предположение о функции распределения.

4 Проверить гипотезу о нормальном распределении с помощью различных критериев.

5 Протестировать гипотезу об однородности и о различии в типичном значении выбранного признака выборки согласно варианту.

6 Оформить отчет.

Контрольные вопросы

1 Что такое нулевая гипотеза?

2 Что называется уровнем значимости (*P-value*)?

3 Какие параметры определяют нормальное распределение?

4 Какая нулевая гипотеза может быть протестирована на основании гистограммы?

5 Как проверяется гипотеза о соответствии выборке нормальному распределению в R?

6 Приведите пример независимых выборок.

7 Приведите пример зависимых выборок.

8 Какая нулевая гипотеза может быть протестирована на основании boxplot()?

9 Зачем проверять гипотезу о равенстве дисперсий в двух выборках, извлеченных из нормально распределенной генеральной совокупности (*F*-критерий)?

10 Как протестировать гипотезу об однородности двух выборок в зависимости от законов распределения генеральных совокупностей, из которых они были извлечены (*T*-критерий, Манна – Уитни, Уилкоксона)?

Лабораторная работа № 6. Методы кластеризации

Цель: выработка практических навыков проведения многомерной классификации методами кластерного анализа средствами R и последующего анализа результатов.

Теоретические сведения

Постановка задачи кластеризации. Пусть имеется выборка U из n объектов наблюдения, каждый из которых характеризуется m числовыми признаками X_j. Известно, что каждый из этих объектов на самом деле относится к одному из k классов, причём признаки объектов из одного класса не слишком сильно различаются, а признаки объектов из разных классов различаются более существенно. Мера различия между признаками объектов может быть определена как расстояние $\rho(x, y)$ между векторами признаков в соответствующем признаковом пространстве.

Алгоритм *k* внутригрупповых средних. Идея алгоритма k внутригрупповых средних (англ. *k-means*) заключается в последовательном пересчёте внутригрупповых средних. Пусть вначале для каждого из k кластеров имеется некоторое начальное внутригрупповое среднее $\overline{x}_l$. Если внутригрупповые средние заданы, то каждый кластер U_l очевидным образом определяется, как множество объектов наблюдения, вектора признаков для которых ближе к центру, чем к центрам других кластеров. После этого внутригрупповое среднее можно пересчитать, после чего снова пересчитать кластеры, пока средние не перестанут

меняться.

В языке *R* для кластеризации методом *k* внутригрупповых средних используется функция kmeans() из пакета stats. Она позволяет производить кластеризацию данных различными методами.

Ниже приведена сигнатура этой функции:

```
kmeans(x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong",
                "Lloyd", "Forgy", "MacQueen"), trace=FALSE) ,
```

где *x* – матрица данных, каждая строка которой является вектором признаков очередного объекта наблюдения:

centers – количество кластеров *k* или набор начальных центров кластеров. Если набор начальных центров не задан, то в качестве начальных центров выбираются *k* случайных объектов наблюдения;

iter.max – максимальное количество итераций алгоритма;

nstart – количество наборов случайных начальных центров кластеров в случае, если центры выбираются случайно;

algorithm – реализация алгоритма *k* внутригрупповых средних, с помощью которой будет производиться кластеризация.

Функция возвращает объект класса *k*-means, который имеет методы print и fitted. Этот объект представляет собой список, содержащий следующие элементы:

cluster – вектор из *n* целых положительных чисел, обозначающих номер кластера соответствующего объекта наблюдения;

centers – матрица, строки которой представляют собой центры соответствующих кластеров;

totss – сумма квадратов расстояний от объектов наблюдения до соответствующих центров кластеров;

withinss – вектор внутрикластерных сумм квадратов расстояний между объектами одного кластера для каждого кластера;

tot.withinss – общая сумма внутрикластерных сумм квадратов расстояний между объектами, т. е. сумма элементов вектора withinss;

betweenss – междукластерная сумма квадратов расстояний, т. е. разница между totss и tot.withinss;

size – количество объектов наблюдения, попавших в каждый из кластеров;

iter – количество внешних итераций, совершённых в ходе работы алгоритма;

ifault – код ошибки, возникшей в ходе работы алгоритма.

Доступ к этим значениям можно получить путём записи их названия в двойных квадратных скобках справа от переменной, содержащей модель, либо через знак доллара. Например, a[["cluster"]] или a$cluster.

Иерархическая кластеризация. Часто задача кластеризации, описанная выше, возникает в несколько более сложной постановке: точное число кластеров *k* может быть не известно. В этом случае использование алгоритма *k* внутригрупповых средних не представляется возможным. Конечно, можно просто пе-

ребирать значения k и для каждого из них использовать этот алгоритм, но сравнение среднеквадратических расстояний до центров кластеров для различных количеств кластеров лишено смысла.

Вместо этого используются алгоритмы иерархической кластеризации (англ. *hierarchical clustering*), которые разбивают множество объектов наблюдения на кластеры, которые, в свою очередь, также разбиваются на кластеры и т. д. Получившийся граф, вершинами которого являются кластеры, а дуги направлены от кластеров более высокого уровня к соответствующим кластерам более низкого уровня, называется дендрограммой.

Построенная дендрограмма содержит информацию о близости между отдельными подмножествами объектов наблюдения, что позволяет подбирать число кластеров и производить более тщательный кластерный анализ в отдельных случаях. Иерархическая кластеризация, как и обыкновенная, производится на основе расстояний $\rho(x, y)$ между объектами наблюдения из заданной выборки. Эти расстояния также могут быть определены по-разному. На их основе строятся более сложные показатели качества кластеризации, которые допускают сравнение для различного числа кластеров. Далее с помощью различных методов, основанных на последовательном объединении и разбиении имеющихся кластеров, строится дендрограмма.

Существует ряд различных методов иерархической кластеризации.

1 Метод Уорда (англ. *Ward's method*).

2 Метод одиночной связи (англ. *single linkage*).

3 Метод полной связи (англ. *complete linkage*).

4 Метод средней связи (англ. *pair-group method using arithmetic averages*).

5 Метод Мак-Куитти (англ. *McQuitty's method*).

Задания

Входные данные: n объектов, каждый из которых характеризуется двумя числовыми признаками $\{x_i\}$ и $\{y_i\}$. Необходимо исследовать работу алгоритмов кластеризации объектов наблюдения по двум признакам. Для каждого набора данных требуется выполнить следующие задания.

1 Провести кластеризацию объектов наблюдения с помощью алгоритма k внутригрупповых средних. Выбрать оптимальное количество кластеров, исходя из критерия минимизации внутригрупповых дисперсий и максимизации расстояния между центрами кластеров.

2 Графически изобразить на плоскости разбиения объектов наблюдения в соответствии с кластерами и в соответствии с классами c_i. Также отметить центры каждого кластера. Количество кластеров должно соответствовать количеству классов.

3 Для разбиения на кластеры вычислить сумму квадратов расстояний от каждого объекта наблюдения до центра соответствующего кластера.

4 Провести кластеризацию исходных данных иерархическим способом. Сравнить результаты кластеризации на тестовом множестве.

Все описанные задания требуется выполнить для двух наборов данных:

1) смоделированные независимые случайные векторы (X, Y), n_1 из которых

относятся к первому классу, а n_2 – ко второму классу. Векторы, относящиеся к первому классу, распределены по гауссовскому закону с математическим ожиданием a_1 и корреляционной матрицей R_1, а векторы, относящиеся ко второму классу, – по гауссовскому закону с математическим ожиданием a_2 и корреляционной матрицей R_2;

2) реальные статистические данные из заданного набора (выдаются преподавателем).

Отчёт, кроме прочих обязательных элементов, должен включать:

– изображения данных в виде точек на плоскости, причём данные из разных классов должны изображаться отличающимися друг от друга;

– изображения результатов кластеризации данных в виде точек на плоскости, причём данные из разных кластеров должны изображаться отличающимися друг от друга; кроме того, требуется отметить центры кластеров;

– значения суммы квадратов расстояний от каждого объекта наблюдения до центра соответствующего кластера.

5 Оформить отчет.

Варианты заданий на лабораторную работу

Все описанные задания требуется выполнить для двух наборов данных.

1 На рисунке А.1 для каждого варианта задания приведены значения количеств объектов наблюдения для каждого класса (n_1 и n_2), значения векторов математических ожиданий для каждого класса (a_1 и a_2) и корреляционные матрицы для каждого класса (R_1 и R_2) для моделируемой выборки из гауссовских случайных векторов.

2 Файлы с наборами реальных данных и вариантами индивидуальных заданий расположены в папке DATA на странице курса в Moodle.

Контрольные вопросы

1 В чём состоит задача кластеризации данных?

2 Какие существуют различные способы определения расстояния между объектами наблюдения по их признакам?

3 К какому классу сложности относится задача кластеризации в классической постановке?

4 Как работает классическая реализация алгоритма k внутригрупповых средних?

5 Что можно сказать о сходимости алгоритма k внутригрупповых средних?

6 Какую функцию минимизирует алгоритм k внутригрупповых средних?

7 Какие существуют альтернативные варианты реализации алгоритма k внутригрупповых средних?

8 Какие существуют методы автоматического выбора начальных центров кластеров для алгоритма k внутригрупповых средних?

9 Что такое иерархическая кластеризация?

Лабораторная работа № 7. Методы прогнозирования. Модель линейной регрессии

Цель: изучение возможности построения уравнения регрессии для прогнозирования непрерывного результативного признака.

Теоретические сведения

Уравнение регрессии – функция, позволяющая по величине изменения одного корреллируемого признака определить среднюю величину другого признака. Выделим основные этапы регрессионного анализа.

Первый этап. Предположение. На этом этапе происходит выбор формы связи между переменными (модель).

Второй этап. Параметризация – происходит оценка значений параметра в выбранной формуле статистической связи. Форма связи (функция) линейная, нелинейная.

Третий этап. Проверка надёжности полученных оценок. На этом этапе осуществляются следующие тесты: *F*-тест (проверка статистической значимости выбранной формы связи), *t*-тест (проверка статистической значимости найденных числовых значений параметра). В результате анализа статистических данных, выбора и построения модели последовательно выполняются все три этапа.

В регрессионном анализе в машинном обучении для оценки качества предсказаний применяется несколько метрик.

1 Средняя абсолютная ошибка (Mean Absolute Error, MAE) – представляет собой среднее абсолютное значение разности между фактическими и предсказанными значениями.

2 Среднеквадратичная ошибка (Mean Squared Error, MSE) – измеряет среднее значение квадрата разности между фактическими и предсказанными значениями.

3 Квадратный корень из MSE (Root Mean Squared Error, RMSE) – представляет интерпретируемую метрику, измеряющую среднее абсолютное отклонение модели.

4 Коэффициент детерминации (*R*-squared, R^2) – измеряет долю дисперсии зависимой переменной, которая объясняется моделью, показывает, насколько хорошо модель соответствует данным. Значения R^2 лежат в диапазоне от 0 до 1, где 1 указывает на идеальное соответствие.

Задания

1 *Сбор данных*. Собрать набор данных, который содержит как обучающие, так и тестовые данные. Обучающие данные будут использоваться для обучения модели, а тестовые данные – для оценки ее производительности.

2 *Предварительная обработка данных*. Перед обучением модели данные должны быть предварительно обработаны. Это может включать в себя удаление выбросов, заполнение пропущенных значений, масштабирование данных и т. д.

3 *Выбор модели.* Выберите модель регрессии, которая лучше всего подходит для вашего набора данных. Некоторые из популярных моделей регрессии включают линейную регрессию, полиномиальную регрессию, регрессию на основе деревьев и т. д.

4 *Обучение модели*. Используйте выбранную модель для обучения на обучающих данных. Это может включать в себя настройку гиперпараметров модели и оптимизацию функции потерь.

5 *Оценка модели*. Оцените производительность модели на тестовых данных. Это может включать в себя расчет метрик, таких как среднеквадратичная ошибка (*RMSE*), средняя абсолютная ошибка (*MAE*) и коэффициент детерминации (R^2).

6 *Анализ результатов*. Проанализируйте результаты и сделайте выводы о производительности модели. Если производительность модели неудовлетворительна, вы можете рассмотреть возможность использования другой модели или дополнительной предварительной обработки данных.

7 *Применение модели*. Если производительность модели удовлетворительна, вы можете использовать ее для предсказания значений для новых данных.

8 Оформить отчет.

Файлы с наборами реальных данных и вариантами индивидуальных заданий расположены в папке DATA на странице курса в Moodle.

Контрольные вопросы

1 Что такое задача регрессии в контексте машинного обучения?

2 Какие основные отличия между задачами регрессии и классификации?

3 Назовите основные метрики, используемые для оценки качества моделей регрессии.

4 Какая метрика будет значима для задачи, где большие ошибки более критичны, и почему?

5 Что такое переобучение и недообучение в контексте моделей регрессии?

6 Какие методы можно применить для борьбы с переобучением модели регрессии?

7 Какую информацию о датасете может передать среднеквадратичная ошибка (MSE) модели регрессии?

8 Что означает значение коэффициента детерминации, равное 1,0?

9 Почему важен процесс отбора признаков в моделях регрессии?

10 Какие методы отбора признаков могут быть применены для моделей регрессии и какие у них преимущества?

Лабораторная работа № 8. Классификация с помощью логистической регрессии

Цель: научиться строить модель логистической регрессии и интерпретировать полученные результаты с точки зрения решения задачи классификации объектов.

Теоретические сведения

Задача классификации в машинном обучении – это задача отнесения объекта к одному из заранее определенных классов на основании его формализованных признаков. Каждый из объектов в этой задаче представляется в виде вектора в n-мерном пространстве, каждое измерение в котором представляет собой описание одного из признаков объекта. Для обучения классификатора необходимо иметь набор объектов, для которых заранее определены классы. Это множество называется обучающей выборкой, её разметка производится вручную, с привлечением специалистов в исследуемой области.

Логистическая регрессия – это алгоритм классификации машинного обучения, используемый для прогнозирования вероятности категориальной зависимой переменной. В логистической регрессии зависимая переменная является бинарной переменной, содержащей данные, закодированные как 1 (да, успех и т. п.) или 0 (нет, провал и т. п.). Другими словами, модель логистической регрессии предсказывает $P(Y=1)$ как функцию X.

Условия логистической регрессии.

Бинарная логистическая регрессия требует, чтобы зависимая переменная также была бинарной.

Для бинарной регрессии фактор уровня 1 зависимой переменной должен представлять желаемый вывод.

Использоваться должны только значимые переменные.

Независимые переменные должны быть независимы друг от друга. Это значит, что модель должна иметь малую мультиколлинеарность или не иметь её вовсе.

Независимые переменные связаны линейно с логарифмическими коэффициентами.

Логистическая регрессия требует больших размеров выборки.

Логистическая регрессия применяется для прогнозирования вероятности возникновения некоторого события по значениям множества признаков. Для этого вводится так называемая зависимая переменная y, принимающая лишь одно из двух значений – как правило, это числа 0 (событие не произошло) и 1 (событие произошло), и множество независимых переменных (также называемых признаками, предикторами или регрессорами) – вещественных x_1, x_2, ..., x_n, на основе значений которых требуется вычислить вероятность принятия того или иного значения зависимой переменной.

Логистическая функция имеет вид

$$y = f(Z) = \frac{1}{1 + e^{-Z}},$$

где Z – линейная комбинация независимых переменных.

Подбор коэффициентов регрессионного уравнения осуществляется на обучающей выборке с помощью метода максимального правдоподобия.

Ключевые термины для логистической регрессии.

Логит (logit) – функция, которая отображает вероятность принадлежности классу в интервале от-∞ до + ∞ (вместо интервала от 0 до 1).

Отношение шансов (odds) – Отношение «успеха» (1) к «неуспеху» (0).

Логарифм отношения шансов (log odds) – результат в преобразованной модели (в виде линейной комбинации), который отображается назад в вероятность.

Реализация на языке R представлена в следующем фрагменте:

```
# чтение данных в формате CSV
rdata <- read.csv("input.csv", sep = ',', header = FALSE)
# оценивание модели
model = glm(formula = target ~ x + y + z, data = rdata, family = binomial)
# вывод результатов оценки модели
print(summary(model))
```

Визуализация трёхмерных данных в языке R может быть проведена с помощью функции plot3d из пакета rgl. Она принимает координаты точек в каждом из трёх пространственных измерений и изображает эти точки на графике. В полученном графическом окне график можно вращать и масштабировать. Сигнатура этой функции приведена ниже.

plot3d(x, y, z, xlab, ylab, zlab, type = "p", col, size, lwd, radius, add = FALSE, aspect = !add, ...),

где x, y, z – векторы координат точек по каждому измерению;

xlab, ylab, zlab – названия координатных осей;

type – способ отображения точек на графике.

Поддерживаемые значения:

"p" – для точек;

"s" – для сфер;

"l" – для линий;

"h" – для отрезков, перпендикулярных плоскости z = 0;

"n" – для отсутствия графического представления;

col – вектор цветов для каждой точки;

size – размер изображаемых точек;

lwd – толщина линий для соответствующего способа представления точек;

radius – радиус сфер для соответствующего способа представления точек;

add – логическое значение, означающее добавлять ли точки на уже имеющийся график или же создать новый;

aspect – логическое значение, означающее подбирать ли масштаб автоматически.

Задания

Провести классификацию объектов методом логистической регрессии для двух наборов данных: смоделированных и реальных. Визуализировать данные.

1 Смоделированные независимые случайные векторы (X, Y, Z), n_1 из которых относятся к первому классу, а n_2 – ко второму классу. Значения количеств объектов наблюдения для каждого класса (n_1 и n_2), значения векторов математических ожиданий для каждого класса (a_1 и a_2) и корреляционные матрицы для каждого класса (R_1 и R_2) для моделируемой выборки из гауссовских случайных векторов представлены на рисунке Б.1.

2 Файлы с наборами реальных данных и вариантами индивидуальных заданий расположены в папке DATA на странице курса в Moodle.

3 Оформить отчет.

Контрольные вопросы

1 Что такое логистическая регрессия?
2 Какие задачи решает логистическая регрессия?
3 В чем отличие логистической регрессии от линейной?
4 Как интерпретировать коэффициенты логистической регрессии?
5 Какие методы оценки качества модели вы знаете?
6 Как выбрать оптимальное количество признаков для модели?
7 Какие ограничения есть у логистической регрессии?
8 Какие существуют методы улучшения качества модели?

Лабораторная работа № 9. Вероятностное обучение с помощью наивного байесовского классификатора

Цель: научиться создавать модель наивного байесовского классификатора.

Теоретические сведения

Наивный байесовский классификатор (Naive Bayes classifier) – вероятностный классификатор на основе формулы Байеса со строгим (наивным) предположением о независимости признаков между собой при заданном классе, что сильно упрощает задачу классификации из-за оценки одномерных вероятностных плотностей вместо одной многомерной.

В данном случае одномерная вероятностная плотность – это оценка вероятности каждого признака отдельно при условии их независимости, а многомерная – оценка вероятности комбинации всех признаков, что вытекает из случая их

зависимости. Именно по этой причине данный классификатор называется наивным, поскольку позволяет сильно упростить вычисления и повысить эффективность алгоритма. Однако такое предположение не всегда является верным на практике и в ряде случаев может привести к значительному ухудшению качества прогнозов.

В основе байесовской классификации (**алгоритм Наивный Байес)** лежит гипотеза максимальной вероятности, т. е. объект d_i считается принадлежащим классу c_j, если при этом достигается наибольшая апостериорная вероятность $\max_c P(c_j / d)$.

По формуле Байеса

$$P(c_j/d) = P(c_j)\cdot P(d/c_j)/P(d) \approx P(\mathrm{c}_j)\cdot P(d/c_j),$$

где $P(d|c_j))$ – вероятность встретить объект d среди объектов класса c_j,

$P(c_j)$ и $P(d)$ – априорные вероятности класса c_j и объекта d (последняя, не влияет на выбор класса и может быть опущена).

В среде *R* расчеты выполняются обычно с использованием фунции NaiveBayes() из пакета klaR или naiveBayes() из пакета e1071.

Задания

1 Провести классификацию объектов методом наивного байесовского классификатора (использовать наборы данных из лабораторной работы № 8).

2 Оформить отчет.

Контрольные вопросы

1 В чём состоит задача классификации?

2 Как вероятность ошибочной классификации оценивается по контрольной выборке? Каким свойствам отвечает её оценка?

3 Какую функцию минимизирует байесовский классификатор?

4 Как записывается формула Байеса? Что такое априорная и апостериорная вероятности и где они фигурируют в этой формуле?

5 В чём достоинства и недостатки байесовского классификатора? Почему он редко используется на практике?

6 Как работает байесовский классификатор для случая двух классов и одинаковых априорных вероятностей появления объектов?

7 Описать процесс генерации модельных данных из многомерных нормальных векторов с заданными векторами математического ожидания и ковариационной матрицей.

8 Подтвердить соответствие полученных данных условиям (проверить равенство выборочных средних и дисперсий математическим ожиданиям и элементам ковариационным матрицам) с помощью проверки гипотез и визуализации.

Лабораторная работа № 10. Классификация с помощью дискриминантного анализа

Цель: изучение основных процедур дискриминантного анализа: дискриминации и классификации, построение и определение количества дискриминантных функций и их разделительной способности.

Теоретические сведения

Линейный дискриминантный анализ (LDA) используется для анализа данных в том случае, когда зависимая переменная категориальная, а предикторы (независимые переменные) интервальные. Основное требование для применения линейного дискриминантного анализа – удовлетворения исходных данных обучающей выборки многомерному гауссовскому закону распределения. Дискриминантный анализ используется для принятия решения о том, какие переменные различают (дискриминируют) две или более возникающие совокупности (группы).

Дискриминантный анализ преследует следующие цели.

1 Определение дискриминантных функций (*discriminant functions*) или линейных комбинаций независимых переменных, которые наилучшим образом различают (дискриминируют) категории (группы) зависимой переменной.

2 Проверка существования между группами значимых различий с точки зрения независимых переменных.

3 Определение предикторов, вносящих наибольший вклад в межгрупповые различия.

4 Отнесение случаев к одной из групп (классификация), исходя из значений предикторов.

5 Оценка точности классификации данных на группы.

Функции классификации. Функции классификации предназначены для определения того, к какой группе наиболее вероятно может быть отнесен каждый объект. Имеется столько же функций классификации, сколько групп.

Модель дискриминантного анализа имеет следующий вид:

$$Группа = \beta_1 X_1 + \beta_2 X_2 ... \beta_k X_k,\ i = 1 \ldots k,$$

где β_j – дискриминантный коэффициент или вес;

X_i – предиктор или независимая переменная.

Дискриминантные переменные используются для того, чтобы отличать один класс (подмножество) от другого. Коэффициенты или веса β_j определяют таким образом, чтобы группы максимально возможно отличались значениями дискриминантной функции. Это происходит тогда, когда отношение межгрупповой суммы квадратов к внутригрупповой сумме квадратов для дискриминантных показателей максимально. Любая другая линейная комбинация предикторов приводит к меньшему значению этого отношения.

Рассмотрим несколько упрощенную геометрическую интерпретацию алгоритма *LDA* для случая двух классов. Пусть дискриминантные переменные x – оси m-мерного евклидова пространства. Каждый объект (наблюдение) является точкой этого пространства с координатами, представляющими собой фиксируемые значения каждой переменной. Если оба класса отличаются друг от друга по наблюдаемым переменным, их можно представить как скопления точек в разных областях рассматриваемого пространства, которые могут частично перекрываться. Для определения положения каждого класса можно вычислить его «центроид», который является воображаемой точкой, координатами которой являются средние значения переменных в данном классе.

Задача дискриминантного анализа заключается в проведении дополнительной оси z, которая проходит через облако точек таким образом, что проекции на нее обеспечивают наилучшую разделяемость на два класса. Ее положение задается линейной дискриминантной функцией (linear discriminant, LD) с весовыми коэффициентами β_j, определяющими вклад каждой исходной переменной x_j.

Если сделать предположение, что ковариационные матрицы объектов классов 1 и 2 равны, т. е. $C = C1 = C2$, то вектор коэффициентов $\beta_1, \ldots, \beta_k$ линейного дискриминанта $z(x)$ может быть вычислен по формуле

$$\beta = C^{-1}(\mu_1 - \mu_2),$$

где C^{-1} – матрица, обратная к ковариационной;

μ_k – вектор средних k-го класса.

Полученная ось совпадает с уравнением прямой, проходящей через центроиды двух групп объектов классов.

Таким образом, в LDA, кроме предположения о нормальности распределения данных в каждом классе, которое на практике выполняется довольно редко, выдвигается еще и более серьезное предположение о статистическом равенстве внутригрупповых матриц дисперсий и корреляций. Если между ними нет серьезных отличий, их объединяют в расчетную ковариационную матрицу

Для проверки гипотезы о многомерном нормальном распределении данных используется многомерная версия критерия согласия Шапиро – Уилка, которая реализована в функции mshapiro.test() из пакета mvnormtest. На вход этой функции подается матрица, строки которой соответствуют переменным, а столбцы – наблюдениям.

Для проверки гипотезы о гомогенности матриц ковариаций используется так называемый М-критерий Бокса, который реализован в функции boxM() из пакета biotools.

Дискриминантный анализ реализован в нескольких пакетах для *R*, но мы рассмотрим применение функции lda() из базового пакета MASS. Поскольку важной характеристикой прогнозирующей эффективности модели является ее ошибка при перекрестной проверке, то в функции lda() пакета MASS заложена реализация скользящего контроля (leave-one-out CV). Напомним, что при этом

из исходной выборки поочередно отбрасывается по одному объекту, строится n моделей дискриминации по $(n - 1)$ выборочным значениям, а исключенное наблюдение каждый раз используется для учета ошибки классификации.

Ниже приведен пример кода, который демонстрирует, как выполнить LDA с помощью функции lda пакета MASS:

```
install.packages("MASS")
library(MASS)
# Предположим, у нас есть данные о цветах, где каждая строка представляет собой образец, а столбцы – различные характеристики
data(iris) # Используем встроенные данные iris для примера
# Выполняем LDA
lda_model <- lda(Species ~ ., data = iris[, -5])
# Получаем прогнозы
pred <- predict(lda_model)$class
# Проверяем точность прогнозов
table(pred, iris$Species)
```

В этом примере мы используем встроенные данные iris, которые содержат информацию о трех видах ирисов: setosa, versicolor и virginica. Мы выполняем LDA, используя все переменные, кроме пятой (пятый столбец – это целевая переменная Species), чтобы предсказать вид ириса на основе остальных характеристик. Затем мы получаем прогнозы и проверяем их точность, сравнивая с реальными значениями вида ириса.

Как только модель установлена и получены дискриминирующие функции, возникает вопрос о том, как хорошо они могут предсказывать, к какой совокупности принадлежит конкретный образец.

Априорная и апостериорная классификация. Классификация действует лучшим образом для выборки, по которой была проведена оценка дискриминирующей функции (апостериорная классификация), чем для свежей выборки (априорная классификация). Поэтому оценивание качества процедуры классификации никогда не производят по той же самой выборке, по которой была оценена дискриминирующая функция. Если желают использовать процедуру для классификации будущих образцов, то ее следует «испытать» на новых объектах. Можно использовать следующий прием: получить дискриминирующую функцию на части выборки (выполнить обучение), а на второй части выборки оценить качество классификации.

Дискриминация наблюдений. Для каждого наблюдения рассчитывается значение дискриминирующей функции. В общем случае наблюдение считается принадлежащим той совокупности, для которой получено наибольшее значение дискриминирующей функции.

Задания

1 Провести классификацию объектов методом *LDA* (использовать наборы данных из лабораторной работы № 8).

2 Оформить отчет.

Контрольные вопросы

1 Какова цель дискриминантного анализа?

2 Что такое дискриминирующая функция?

3 Перечислите методы расчёта коэффициентов в дискриминирующих функциях.

4 Суть метода пошагового анализа с включением.

5 Суть метода пошагового анализа с исключением.

6 В чём суть дискриминантного анализа с обучением?

7 Что такое априорная и апостериорная классификация?

8 Каким образом можно классифицировать новый объект, если известны дискриминирующие функции для всех групп?

Лабораторная работа № 11. Классификация с использованием деревьев решений и правил

Цель: научиться применять алгоритмы построения деревьев решений, решающих правил, а также проводить сравнительный анализ классификационных моделей на их основе.

Теоретические сведения

Алгоритм CART (Classification and Regression Trees) представляет собой метод построения двоичных деревьев решений для задач классификации и регрессии.

Цель алгоритма: разбить данные на подмножества таким образом, чтобы в каждом подмножестве объекты были как можно более однородными с точки зрения целевой переменной.

На каждом шаге алгоритм выбирает переменную и значение, чтобы разделить данные на два подмножества. Выбор происходит на основе критерия неоднородности, такого как критерий Джини для задач классификации или критерий наименьших квадратов для задач регрессии. Этот процесс повторяется рекурсивно для каждого подмножества, пока не выполнится условие останова.

Дерево строится до тех пор, пока не будет достигнуто определенное количество объектов в листовых узлах, или до тех пор, пока нет возможности провести дальнейшее разделение.

Простота интерпретации полученной модели в виде дерева решений.

Устойчивость к выбросам и некорректным данным.

Склонность к переобучению при недостаточном прунинге дерева.

Не всегда способен улавливать сложные зависимости в данных.

Алгоритм CART представляет собой мощный инструмент для построения простых и понятных моделей, особенно в случаях, когда интерпретация решений играет важную роль.

Пример использования алгоритм CART для классификации в R:

```
# Установка пакета rpart, если он не установлен
# install.packages("rpart")
# Загрузка необходимых библиотек
library(rpart)
library(rpart.plot)  # Дополнительная библиотека для визуализации деревьев
# Создание и обучение модели с использованием алгоритма CART
# Предположим, что у вас есть данные в переменной "data", где последний столбец-это целевая переменная, а все предыдущие – признаки
model <- rpart(target ~ feature1 + feature2 + ..., data = data, method = "class")
# Визуализация построенного дерева решений
rpart.plot(model)  # Это отобразит дерево решений, используя библиотеку rpart.plot
```

Задания

1 Провести классификацию объектов CART (использовать наборы данных из лабораторной работы № 8), выполнив настройку параметров модели и оценить ее производительность на обучающем и тестовом наборах данных.

2 Визуализировать полученные деревья решений для наглядного представления классификационных правил.

3 Сформулировать выводы о производительности и применимости классификационных моделей на основе деревьев решений и решающих правил.

4 Оформить отчет.

Контрольные вопросы

1 Что такое дерево решений?

2 Какие методы используются для построения деревьев решений?

3 Как определить количество уровней в дереве решений?

4 Что такое минимальное значение ошибки предсказания (minimum prediction error)?

5 Каковы преимущества использования деревьев решений?

6 В чём заключаются недостатки использования деревьев решений?

7 Что такое pruning (обрезка) дерева решений?

8 Какие существуют виды pruning (обрезки) деревьев решений?

9 Каковы критерии остановки процесса построения дерева решений?

10 Что такое переобучение (overfitting)?

Лабораторная работа № 12. Метрики качества решения задач классификации. ROC-анализ

Цель: научиться выбирать лучший алгоритм для осуществления классификации объектов с помощью ROC-анализа.

Теоретические сведения

Когда речь заходит о сравнении методов классификации в R, важно учитывать несколько ключевых аспектов. Сравнение методов классификации в R осуществляется на основе нескольких факторов, включая производительность модели, удобство использования, способность к обобщению, требования к предварительной обработке данных и т. д. К широко используемым инструментам сравнения качества предсказания класса в R различными методами относятся: кросс-валидация для оценки производительности модели, а также сравнение их метрик качества, точность, полнота, *F*-мера или ROC-кривая. Необходимо также учитывать интерпретируемость моделей, а также их способность обобщения для новых данных.

Большой выбор методов классификации в R дает возможность подобрать подходящий метод в зависимости от конкретной задачи и особенностей данных.

Задача алгоритма классификации состоит в том, чтобы относить ранее неизвестные объекты к тому или иному классу.

Результатом классификации могут быть четыре варианта:

1) истинно положительный результат (true-positive, *TP*) – предсказано True, в действительности True;

2) ложноположительный результат (false-positive, *FP*) – предсказано True, в действительности False;

3) истинно отрицательный результат (true-negative, *TN*) – предсказано False, в действительности False;

4) ложноотрицательный результат (false-negative, *FN*) – предсказано False, в действительности True.

Для численного представления качества оценки в R используют матрицу ошибок (confusion matrix). Матрица ошибок в машинном обучении (таблица 5) – это таблица, которая показывает количество правильных и неправильных предсказаний модели, в которой по строкам указываются реальные значения (истинные классы), а по столбцам – предсказанные значения (прогнозы модели).

Таблица 5 – Матрица ошибок

	Predicted = 0	Predicted = 1
Actual = 0	True Negatives (*TN*)	False Positives (*FP*)
Actual = 1	False Negatives (*FN*)	True Positives (*TP*)

ROC-анализ (Receiver Operating Characteristic) – это метод оценки качества модели классификации, основанный на анализе кривой зависимости истинно положительных результатов от ложноположительных. ROC-анализ используется для сравнения методов классификации и определения оптимального порога для принятия решений в задаче логистической регрессии.

Критерий AUC-ROC устойчив к несбалансированным классам и может быть интерпретирован как вероятность того, что случайно выбранный positive-объект будет проранжирован классификатором выше (будет иметь более высокую вероятность быть positive), чем случайно выбранный negative-объект.

Для оценки качества логистической модели строится матрица ошибок, рассчитываются коэффициенты чувствительности и специфичности, которые варьируются в зависимости от порога отсечения.

Результатом логистической регрессии является вероятность того, что событие произойдет, но, чтобы сделать предсказание, нам необходимо определить пороговое (threshold) значение t такое, что:

$$P(y = 1) >= t, \text{ событие произошло;}$$

$$P(y = 1) < t, \text{ событие не произошло.}$$

При высоком пороговом значении модель редко будет предсказывать положительный результат (только при высокой вероятности $P(y = 1)$). И обратно, при низком пороговом значении модель будет чаще предсказывать положительный исход и реже отрицательный. Матрица ошибок также может использоваться для настройки модели и выбора оптимального порога принятия решений.

ROC-кривая строится на графике, где по оси абсцисс откладываются значения ложноположительных результатов, а по оси ординат – истинно положительные результаты. Каждой точке на кривой соответствует определенный порог принятия решения. Чем ближе кривая к верхнему левому углу, тем лучше модель классификации. Отсюда можно получить два значения, которые помогут нам определить какие ошибки делает модель.

Чувствительность – доля верно предсказанных позитивных исходов (True positive rate).

$$Sensitivity = TP/(TP + FN).$$

Специфичность – доля верно предсказанных негативных исходов (True negative rate).

$$Specificity = TN/(TN + FP).$$

Модель с более высоким пороговым значением будет иметь более высокую чувствительность и низкую специфичность. Модель с низким пороговым значением наоборот. Если предпочтений нет, можно оставить пороговое значение $t = 0{,}5$. Подобрать необходимое пороговое значение можно с помощью ROC-кривой (рисунок 3).

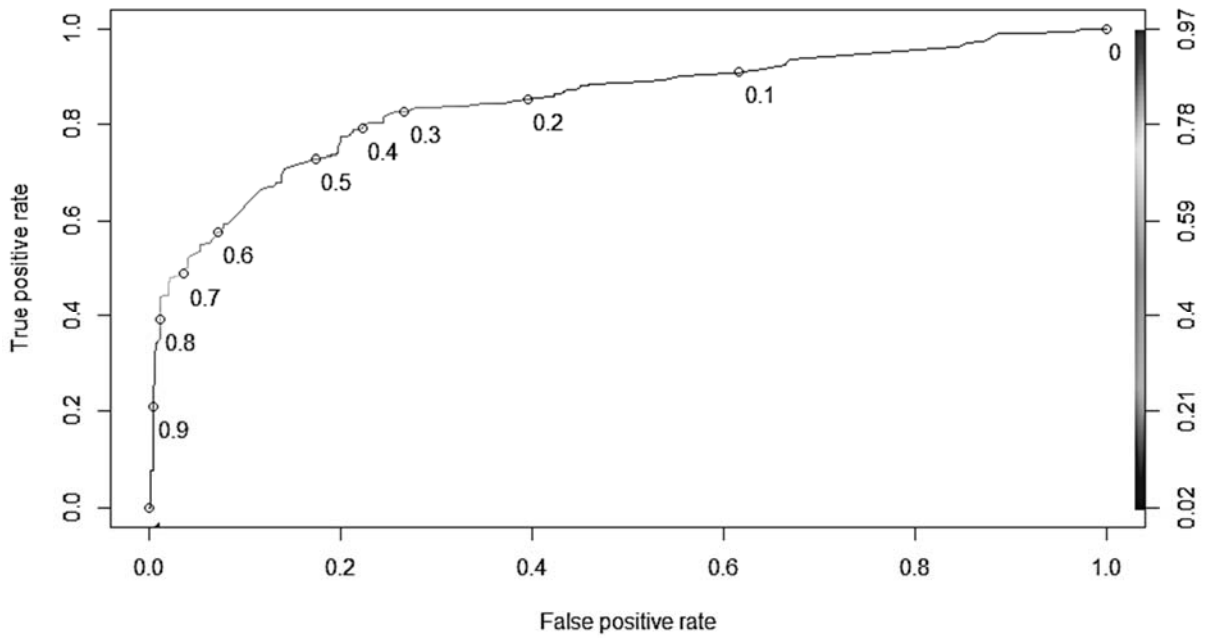


Рисунок 3 – Визуализация ROC-кривой

True negative rate указывается на оси абсцисс, True positive rate – на оси ординат. Кривая показывает соотношения этих величин при разных значениях пороговой величины.

Построить ROC-кривую в R можно следующим набором команд (используется встроенный набор данных iris):

```
# Загрузка необходимых библиотек
library(pROC)
library(ggplot2)
# Создание примера модели логистической регрессии и ROC-кривой
# Имитация обучающего и тестового набора данных (используем встроенный набор данных iris)
data(iris)
set.seed(123)
train_indices <- sample(1:nrow(iris), nrow(iris)*0.7)  # 70 % данных для обучения
train_data <- iris[train_indices, ]
test_data <- iris[-train_indices, ]
# Создание модели логистической регрессии
logit_model <- glm(Species ~ ., data = train_data, family = "binomial")
# Построение ROC-кривой
roc_curve <- roc(test_data$Species, predict(logit_model, newdata = test_data, type = "response")[, "versicolor"])
# Визуализация ROC-кривой
ggplot(as.data.frame(roc_curve), aes(x = 1-specificity, y = sensitivity)) +
  geom_line() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
```

```
  labs(title = "ROC Curve",
       x = "False Positive Rate",
       y = "True Positive Rate") +
  theme_minimal()
```

Задание

С помощью ROC- анализа провести сравнение результатов классификации, проведенных в лабораторных работах № 8–10.

Контрольные вопросы

1 Какие метрики используются для оценки качества предсказания?

2 Что такое матрица ошибок?

3 Что представляет собой ROC-кривая и для чего она используется в контексте оценки качества классификационных моделей?

4 Как интерпретировать ROC-кривую и какие параметры из неё могут быть полезны для оценки качества модели?

5 Какие значения на ROC-кривой указывают на хорошую предсказательную способность модели, а какие наоборот?

6 Как соотносятся ROC-кривая и метрики, такие как AUC-ROC (площадь под ROC-кривой), с точностью, полнотой и *F*-мерой?

7 Каковы преимущества использования ROC-анализа для сравнения методов классификации по сравнению с использованием только точности или полноты?

Лабораторная работа № 13. Метрики качества решения задач классификации

Цель: научиться выбирать лучший алгоритм для осуществления классификации объектов с помощью различных метрик оценки качества классификации.

Теоретические сведения

Три из основных метрик, которые широко используются для оценки качества работы классификационных моделей, включают точность (Precision), полноту (Recall) и *F*-меру (F1-score).

1 Точность измеряет, сколько из объектов, которые модель отнесла к положительному классу, действительно принадлежат к положительному классу. Она определяется как отношение числа верно классифицированных положительных объектов к общему количеству объектов, которые модель отнесла к положительному классу.

$$Precision = TP/(TP + FP).$$

2 Полнота измеряет, сколько из общего числа реальных положительных объектов модель успешно обнаружила. Она определяется как отношение числа верно классифицированных положительных объектов к общему количеству реальных положительных объектов.

$$Recall = TP/(TP + FN).$$

3 *F*-мера представляет собой сбалансированную метрику, которая учитывает и точность, и полноту. Она является гармоническим средним между точностью и полнотой.

$$F1\text{-}score = 2\cdot(Precision\cdot Recall)/(Precision + Recall).$$

Точность и полнота обычно работают в противоположных направлениях: увеличение точности может привести к снижению полноты и наоборот. *F*-мера учитывает обе эти метрики и представляет собой компромисс между точностью и полнотой. *F*-мера особенно полезна в случаях, когда классы несбалансированы, т. е. когда количество объектов в одном классе существенно превышает количество объектов в другом.

Эти метрики помогают в оценке качества работы классификационных моделей и помогают понять, насколько хорошо модель способна правильно классифицировать положительные и отрицательные объекты.

Задание

Выполнить сравнение методов классификации, проведенных в лабораторных работах № 8–10, с использованием перечисленных метрик Precision, Recall и *F*-меры (*F1*-score).

Контрольные вопросы

1 Что такое точность (Accuracy) и как она рассчитывается?

2 Что такое полнота (Recall или True Positive Rate) и как она связана с точностью?

3 Что такое точность (Precision) и как она влияет на качество модели?

4 Что такое *F*-мера (*F1*-score) и как она комбинирует точность и полноту?

5 Что такое матрица ошибок (Confusion Matrix) и как она помогает оценить качество модели?

6 Как соотносятся ROC-кривая и метрики, такие как AUC-ROC (площадь под ROC-кривой), с точностью, полнотой и *F*-мерой?

Список литературы

1 **Борисов, В. В.** Экспертные системы: учебное пособие / В. В. Борисов, А. В. Бобряков, А. Е. Мисник. – Смоленск: Универсум, 2021. – 110 с.

2 **Кабаков, Р.** R в действии. Анализ и визуализация данных на языке R: практическое руководство / Р. Кабаков. – 2-е изд. – Москва: ДМК Пресс, 2023. – 590 с.

3 **Лонг, Дж. Д.** Книга рецептов: Проверенные рецепты для статистики, анализа и визуализации данных / Дж. Д. Лонг, П. Титор. – Москва: ДМК Пресс, 2020. – 510 с.

4 **Мастицкий, С. Э.** Статистический анализ и визуализация данных с помощью R: практическое руководство / С. Э. Мастицкий, В. К. Шитиков. – 2-е изд. – Москва : ДМК Пресс, 2023. – 497 с.

5 **Гайдель, А. В.** Лабораторный практикум по курсу «Интеллектуальный анализ данных»: практикум / А. В. Гайдель, А. Г. Храмов. – Самара: Самар. ун-т, 2019. – 104 с.

Приложение А
(обязательное)

№	n_1	a_1	R_1	n_2	a_2	R_2
0	100	$\begin{pmatrix}0\\0\end{pmatrix}$	$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$	100	$\begin{pmatrix}2\\2\end{pmatrix}$	$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$
1	100	$\begin{pmatrix}-1\\-1\end{pmatrix}$	$\begin{pmatrix}1 & 0,5\\0,5 & 2\end{pmatrix}$	200	$\begin{pmatrix}2\\2\end{pmatrix}$	$\begin{pmatrix}2 & 0,5\\0,5 & 1\end{pmatrix}$
2	1000	$\begin{pmatrix}-1\\0\end{pmatrix}$	$\begin{pmatrix}1 & 0,9\\0,9 & 2\end{pmatrix}$	2000	$\begin{pmatrix}2\\3\end{pmatrix}$	$\begin{pmatrix}2 & 0,9\\0,9 & 1\end{pmatrix}$
3	100	$\begin{pmatrix}-1\\1\end{pmatrix}$	$\begin{pmatrix}1 & 0,1\\0,1 & 2\end{pmatrix}$	50	$\begin{pmatrix}2\\4\end{pmatrix}$	$\begin{pmatrix}2 & 0,1\\0,1 & 1\end{pmatrix}$
4	1000	$\begin{pmatrix}0\\-1\end{pmatrix}$	$\begin{pmatrix}1 & -0,5\\-0,5 & 2\end{pmatrix}$	500	$\begin{pmatrix}3\\2\end{pmatrix}$	$\begin{pmatrix}2 & -0,5\\-0,5 & 1\end{pmatrix}$
5	100	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1 & -0,9\\-0,9 & 2\end{pmatrix}$	1000	$\begin{pmatrix}3\\4\end{pmatrix}$	$\begin{pmatrix}2 & -0,9\\-0,9 & 1\end{pmatrix}$
6	1000	$\begin{pmatrix}1\\-1\end{pmatrix}$	$\begin{pmatrix}1 & -0,1\\-0,1 & 2\end{pmatrix}$	100	$\begin{pmatrix}4\\2\end{pmatrix}$	$\begin{pmatrix}2 & -0,1\\-0,1 & 1\end{pmatrix}$
7	100	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}2 & 1\\1 & 1\end{pmatrix}$	200	$\begin{pmatrix}4\\3\end{pmatrix}$	$\begin{pmatrix}2 & 0,5\\0,5 & 1\end{pmatrix}$
8	1000	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}2 & 1\\1 & 1\end{pmatrix}$	2000	$\begin{pmatrix}4\\4\end{pmatrix}$	$\begin{pmatrix}2 & 0,1\\0,1 & 1\end{pmatrix}$
9	100	$\begin{pmatrix}-1\\-1\end{pmatrix}$	$\begin{pmatrix}2 & 1\\1 & 1\end{pmatrix}$	50	$\begin{pmatrix}-4\\-4\end{pmatrix}$	$\begin{pmatrix}2 & -1\\-1 & 1\end{pmatrix}$
10	1000	$\begin{pmatrix}-1\\0\end{pmatrix}$	$\begin{pmatrix}2 & -1\\-1 & 1\end{pmatrix}$	500	$\begin{pmatrix}-4\\-3\end{pmatrix}$	$\begin{pmatrix}2 & -0,5\\-0,5 & 1\end{pmatrix}$
11	100	$\begin{pmatrix}-1\\1\end{pmatrix}$	$\begin{pmatrix}2 & -1\\-1 & 1\end{pmatrix}$	1000	$\begin{pmatrix}-4\\-2\end{pmatrix}$	$\begin{pmatrix}2 & -0,1\\-0,1 & 1\end{pmatrix}$
12	1000	$\begin{pmatrix}0\\-1\end{pmatrix}$	$\begin{pmatrix}2 & -1\\-1 & 1\end{pmatrix}$	100	$\begin{pmatrix}-3\\-4\end{pmatrix}$	$\begin{pmatrix}2 & 0,1\\0,1 & 1\end{pmatrix}$
13	100	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1 & 1\\1 & 2\end{pmatrix}$	200	$\begin{pmatrix}-3\\-2\end{pmatrix}$	$\begin{pmatrix}1 & 0,5\\0,5 & 2\end{pmatrix}$
14	1000	$\begin{pmatrix}1\\-1\end{pmatrix}$	$\begin{pmatrix}1 & 1\\1 & 2\end{pmatrix}$	2000	$\begin{pmatrix}-2\\-4\end{pmatrix}$	$\begin{pmatrix}1 & 0,1\\0,1 & 2\end{pmatrix}$
15	100	$\begin{pmatrix}1\end{pmatrix}$	$\begin{pmatrix}1 & 1\end{pmatrix}$	50	$\begin{pmatrix}-2\end{pmatrix}$	$\begin{pmatrix}1 & -1\end{pmatrix}$

Рисунок А.1

Приложение Б
(обязательное)

№	n_1	a_1	R_1	n_2	a_2	R_2
0	100	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$	100	$\begin{pmatrix}2\\2\\2\end{pmatrix}$	$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$
1	100	$\begin{pmatrix}-2\\-2\\-2\end{pmatrix}$	$\begin{pmatrix}2&1&0,2\\1&4&1\\0,2&1&2\end{pmatrix}$	200	$\begin{pmatrix}4\\4\\4\end{pmatrix}$	$\begin{pmatrix}4&1&0,2\\1&2&0,2\\0,2&0,2&2\end{pmatrix}$
2	1000	$\begin{pmatrix}-2\\0\\2\end{pmatrix}$	$\begin{pmatrix}2&1&1\\1&4&1,4\\1&1,4&2\end{pmatrix}$	2000	$\begin{pmatrix}4\\6\\8\end{pmatrix}$	$\begin{pmatrix}2&1&1\\1&2&1,4\\1&1,4&4\end{pmatrix}$
3	100	$\begin{pmatrix}-2\\2\\-2\end{pmatrix}$	$\begin{pmatrix}2&0,2&0,4\\0,2&4&0,4\\0,4&0,4&2\end{pmatrix}$	50	$\begin{pmatrix}4\\8\\-8\end{pmatrix}$	$\begin{pmatrix}4&0,2&0,3\\0,2&2&0,4\\0,3&0,4&2\end{pmatrix}$
4	1000	$\begin{pmatrix}0\\-2\\2\end{pmatrix}$	$\begin{pmatrix}2&-1&1\\-1&4&1\\1&1&2\end{pmatrix}$	500	$\begin{pmatrix}4\\2\\-2\end{pmatrix}$	$\begin{pmatrix}2&-1&1\\-1&3&-1\\1&-1&3\end{pmatrix}$
5	100	$\begin{pmatrix}0\\2\\-2\end{pmatrix}$	$\begin{pmatrix}2&-1&0,5\\-1&4&0,5\\0,5&0,5&2\end{pmatrix}$	1000	$\begin{pmatrix}4\\6\\2\end{pmatrix}$	$\begin{pmatrix}4&1&-1\\1&2&1\\-1&1&2\end{pmatrix}$
6	1000	$\begin{pmatrix}2\\-2\\0\end{pmatrix}$	$\begin{pmatrix}2&-1&0,1\\-1&4&-1\\0,1&-1&2\end{pmatrix}$	100	$\begin{pmatrix}4\\2\\-4\end{pmatrix}$	$\begin{pmatrix}2&0,1&-1\\0,1&2&-1\\-1&-1&4\end{pmatrix}$
7	100	$\begin{pmatrix}2\\0\\-2\end{pmatrix}$	$\begin{pmatrix}4&1&1\\1&2&1\\1&1&2\end{pmatrix}$	200	$\begin{pmatrix}6\\4\\-6\end{pmatrix}$	$\begin{pmatrix}2&1&1\\1&2&-1\\1&-1&4\end{pmatrix}$
8	1000	$\begin{pmatrix}3\\1\\3\end{pmatrix}$	$\begin{pmatrix}4&1&-1\\1&2&1\\-1&1&2\end{pmatrix}$	2000	$\begin{pmatrix}-1\\4\\-1\end{pmatrix}$	$\begin{pmatrix}4&0,1&-1\\0,1&2&0,1\\-1&0,1&2\end{pmatrix}$
9	100	$\begin{pmatrix}-1\\-1\\-1\end{pmatrix}$	$\begin{pmatrix}4&1&1\\1&2&0,1\\1&0,1&2\end{pmatrix}$	50	$\begin{pmatrix}3\\3\\3\end{pmatrix}$	$\begin{pmatrix}4&-1&-1\\-1&2&0,1\\-1&0,1&2\end{pmatrix}$

Рисунок Б.1

10	1000	$\begin{pmatrix}-2\\0\\2\end{pmatrix}$	$\begin{pmatrix}3&-1&1\\-1&3&1\\1&1&2\end{pmatrix}$	500	$\begin{pmatrix}2\\-4\\-2\end{pmatrix}$	$\begin{pmatrix}2&-1&1\\-1&2&1\\1&1&4\end{pmatrix}$
11	100	$\begin{pmatrix}3\\3\\3\end{pmatrix}$	$\begin{pmatrix}2&1&0,2\\1&4&1\\0,2&1&2\end{pmatrix}$	1000	$\begin{pmatrix}-1\\-1\\-1\end{pmatrix}$	$\begin{pmatrix}2&1&1\\1&4&1,4\\1&1,4&2\end{pmatrix}$
12	1000	$\begin{pmatrix}3\\3\\5\end{pmatrix}$	$\begin{pmatrix}2&0,2&0,4\\0,2&4&0,4\\0,4&0,4&2\end{pmatrix}$	100	$\begin{pmatrix}-1\\-1\\9\end{pmatrix}$	$\begin{pmatrix}2&-1&1\\-1&4&1\\1&1&2\end{pmatrix}$

Окончание рисунка Б.1