

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

В.В. МЯСНИКОВ

ОСНОВЫ СТАТИСТИЧЕСКОЙ ТЕОРИИ РАСПОЗНАВАНИЯ ОБРАЗОВ И МАШИННОГО ОБУЧЕНИЯ

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С. П. Королева» в качестве практикума для обучающихся по основной образовательной программе высшего образования по направлению подготовки 10.05.03 Информационная безопасность автоматизированных систем

САМАРА
Издательство Самарского университета
2023

УДК 004.93(075)
ББК 16.633я7
М994

Рецензенты: д-р техн. наук, доц. С. Б. П о п о в,
д-р техн. наук, проф. В. А. Ф у р с о в

Мясников, Владислав Валерьевич

М994 **Основы статистической теории распознавания образов и машинного обучения:** практикум / *В.В. Мясников.* – Самара: Издательство Самарского университета, 2023. – 124 с.

ISBN 978-5-7883-1932-2

Издание представляет собой сборник указаний к лабораторным работам по статистической теории распознавания образов, для каждой из которых приводятся описание теоретических основ, задание и список контрольных вопросов. Разделы теоретических основ содержат описания наиболее распространенных методов и алгоритмов статистической теории распознавания образов: методов оптимальной классификации (классификатор Байеса, минимаксный классификатор и классификатор Неймана-Пирсона), методов построения и настройки линейных классификаторов и классификаторов, основанных на оценивании плотностей вероятностей, а также методов автоматической классификации – кластеризации и таксономии.

Настоящий практикум предназначен для обучающихся факультета информатики по направлениям 01.03.02 Прикладная математика и информатика и 10.05.03 Информационная безопасность автоматизированных систем.

УДК 004.93(075)
ББК 16.633я7

ISBN 978-5-7883-1932-2

© Самарский университет, 2023

ОГЛАВЛЕНИЕ

Введение	7
1 Моделирование экспериментальных данных	
для решения задач распознавания образов	8
1.1 Теоретические основы лабораторной работы.....	8
1.1.1 Моделирование случайного вектора с нормальным законом распределения.....	8
1.1.2 Оценивание параметров нормального закона распределения.....	11
1.1.3 Меры близости нормальных распределений	11
1.1.4 Моделирование бинарных случайных векторов с независимыми координатами	14
1.2 Порядок выполнения лабораторной работы.....	15
1.2.1 Исходные данные	15
1.2.2 Общий план выполнения работы.....	15
1.2.3 Содержание отчета.....	16
2 Оптимальные стратегии теории статистического	
расознавания образов	17
2.1 Теоретические основы лабораторной работы.....	17
2.1.1 Постановка задачи классификации	17
2.1.2 Качество классификатора	20
2.1.3 Оптимальные стратегии классификации: классификатор Байеса.....	21
2.1.4 Оптимальные стратегии классификации: минимаксный классификатор	24
2.1.5 Оптимальные стратегии классификации: классификатор Неймана-Пирсона	27
2.1.6 Типовые решения оптимальных стратегий классификации: нормально распределенные вектора признаков	28
2.1.7 Типовые решения оптимальных стратегий классификации: бинарные вектора признаков	33

2.1.8 Вычисление вероятностей ошибочной классификации: общий случай.....	35
2.1.9 Вычисление вероятностей ошибочной классификации: нормально распределенные вектора признаков	37
2.1.10 Вычисление вероятностей ошибочной классификации: бинарные вектора признаков.....	39
2.1.11 Экспериментальная оценка вероятностей ошибочной классификации.....	40
2.2 Порядок выполнения лабораторной работы	42
2.2.1 Исходные данные	42
2.2.2 Общий план выполнения работы.....	42
2.2.3 Содержание отчета.....	43
3 Линейные классификаторы	44
3.1 Теоретические основы лабораторной работы.....	44
3.1.1 Постановка задачи построения линейного классификатора	44
3.1.2 Линейный классификатор, минимизирующий суммарную вероятность ошибочной классификации	45
3.1.3 Обобщенная формула построения линейных классификаторов для различных критериев. Классификатор Фишера.....	49
3.1.4 Линейный классификатор, минимизирующий СКО решения	51
3.1.5 Последовательная корректировка линейного классификатора: алгоритм перцептрона	55
3.1.6 Последовательная корректировка линейного классификатора: стохастическая аппроксимация и процедура Роббинса-Монро.....	57
3.1.7 Общая схема построения линейных классификаторов, основанная на методе стохастической аппроксимации.....	60
3.2 Порядок выполнения лабораторной работы	63
3.2.1 Исходные данные	63
3.2.2 Общий план выполнения работы.....	64
3.2.3 Содержание отчета.....	65

4	Линейные классификаторы. Метод опорных векторов	66
4.1	Теоретические основы лабораторной работы.....	66
4.1.1.	Метод опорных векторов для случая линейно разделимых классов	66
4.1.2.	Метод опорных векторов для случая линейно неразделимых классов	70
4.1.3	Использование ядер в методе опорных векторов и переход в пространства высокой размерности.....	71
4.1.4	Достоинства и недостатки метода опорных векторов ..	73
4.2	Порядок выполнения лабораторной работы.....	74
4.2.1	Исходные данные	74
4.2.2	Общий план выполнения работы.....	74
4.2.3	Содержание отчета.....	75
5	Классификаторы, основанные на непараметрических оценках плотности вероятностей	76
5.1	Теоретические основы лабораторной работы.....	76
5.1.1.	Метод Парзена оценивания плотности вероятностей..	77
5.1.2.	Метод К ближайших соседей.....	81
5.1.3	Оценка априорных вероятностей.....	84
5.2	Порядок выполнения лабораторной работы.....	84
5.2.1	Исходные данные	84
5.2.2	Общий план выполнения работы.....	85
5.2.3	Содержание отчета.....	85
6	Автоматическая классификация	86
6.1	Теоретические основы лабораторной работы.....	86
6.1.1	Постановка задачи автоматической классификации	86
6.1.2	Меры сходства.....	87
6.1.3	Критерии кластеризации	88
6.1.4	Простой алгоритм выделения кластеров	89
6.1.5	Алгоритм максиминного расстояния	90
6.1.6	Алгоритм К внутригрупповых средних	92
6.3	Порядок выполнения лабораторной работы.....	96
6.3.1	Исходные данные	96

6.3.2	Общий план выполнения работы.....	96
6.3.3	Содержание отчета.....	97
7	Справочные сведения о языке программирования PYTHON и средствах машинного обучения.....	98
7.1	Язык Python.....	98
7.1.1	Общие сведения.....	98
7.1.2	Установка интерпретатора языка и пакетов расширений.....	100
7.2	PyCharm – интегрированная среда разработки для языка программирования Python	105
7.3	Библиотека машинного обучения scikit-learn (machine learning in Python)	108
7.4	Прикладная программа синтеза и классификации данных....	109
7.4.1	Синтеза данных для классификации	109
7.4.2	Классификации данных с использованием метода опорных векторов на scikit-learn.....	110
8	Контрольные вопросы.....	114
8.1	Контрольные вопросы к лабораторной работе № 1	114
8.2	Контрольные вопросы к лабораторной работе № 2	114
8.3	Контрольные вопросы к лабораторной работе № 3	115
8.4	Контрольные вопросы к лабораторной работе № 4	115
8.5	Контрольные вопросы к лабораторной работе № 5	116
8.6	Контрольные вопросы к лабораторной работе № 6	117
9	Варианты заданий	118
9.1	Варианты заданий к лабораторным работам № 1–5	118
9.2	Варианты заданий к лабораторной работе 6.....	119
	Заключение	121
	Библиографический список	122

ВВЕДЕНИЕ

Методы машинного обучения и распознавания образов в современное время являются ядром технологий искусственного интеллекта, используемых с целью построения современной цифровой экономики.

Настоящее учебное издание предназначен для изучения студентами основ классических (статистических) методов распознавания образов и машинного обучения, а также освоение ими практики их применения с использованием наиболее популярных средств в современных информационных технологиях: языка Python и библиотеки машинного обучения scikit-learn.

В данном издании последовательно рассматриваются методы и алгоритмы статистической теории распознавания образов: методы оптимальной классификации (классификатор Байеса, минимаксный классификатор и классификатор Неймана-Пирсона), методы построения и настройки линейных классификаторов и классификаторов, основанных на оценивании плотностей вероятностей, а также методы автоматической классификации – кластеризации и таксономии. Для каждой из тем представлен каркас программы на Python, демонстрирующий элементы реализации и/или использования соответствующего метода, представлен план выполнения для этой темы практического (лабораторного) задания и отчетности по нему. Завершает каждый раздел список вопросов по соответствующей теме.

В заключительной части настоящего издания представлено краткое описание способа установки и использования Python и иного программного обеспечения, необходимого для его установки, конфигурирования и использования.

1 МОДЕЛИРОВАНИЕ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ ДЛЯ РЕШЕНИЯ ЗАДАЧ РАСПОЗНАВАНИЯ ОБРАЗОВ

Цель работы – подготовка экспериментального материала для решения задач распознавания образов, получение навыков работы на языке Python.

1.1 Теоретические основы лабораторной работы

1.1.1 Моделирование случайного вектора с нормальным законом распределения

Пусть $\bar{X}=(X_0, \dots, X_{n-1})^T$ – n -мерный случайный вектор, имеющий нормальный закон распределения: $\bar{X} \sim N(\bar{M}, B)$. Это означает, что плотность вероятностей случайного вектора \bar{X} имеет вид

$$f(\bar{x}) = \frac{1}{(2\pi)^{n/2} \sqrt{|B|}} \exp\left(-\frac{1}{2}(\bar{x}-\bar{M})^T B^{-1}(\bar{x}-\bar{M})\right),$$

где $|\dots|$ – определитель матрицы; $(\dots)^T$ – оператор транспонирования; $\bar{M}=(M_0, \dots, M_{n-1})^T$ – вектор математических ожиданий координат вектора $\bar{X}: M_i = MX_i$ ($i=\overline{0, n-1}$); B – корреляционная матрица

$$B = \begin{pmatrix} B_{00} & B_{01} & \dots & B_{0(n-1)} \\ B_{10} & B_{11} & \dots & B_{1(n-1)} \\ \vdots & & & \\ B_{(n-1)0} & B_{(n-1)1} & \dots & B_{(n-1)(n-1)} \end{pmatrix},$$

элементами которой являются всевозможные корреляционные моменты: $B_{ij} = M(X_i - M_i)(X_j - M_j)$, $(i, j = \overline{0, n-1})$. Очевидно, что вектор \bar{M} и матрица B полностью определяют нормальный закон распределения.

Вектор $\bar{X} \sim N(\bar{M}, B)$ можно получить специальным линейным преобразованием вектора $\bar{\xi} = (\xi_0, \dots, \xi_{n-1})^T$, компоненты которого суть независимые случайные величины, имеющие стандартный нормальный закон распределения:

$$\xi_i \sim N(0, 1), \quad m_i = 0, \quad \sigma_i^2 = 1, \quad (i = \overline{0, n-1}).$$

Обычно предполагают, что матрица A преобразования $\bar{X} = A\bar{\xi} + \bar{M}$ является треугольной, то есть

$$A = \begin{pmatrix} a_{00} & 0 & \dots & 0 \\ a_{10} & a_{11} & \dots & 0 \\ \vdots & & & \\ a_{(n-1)0} & a_{(n-1)1} & \dots & a_{(n-1)(n-1)} \end{pmatrix}.$$

Коэффициенты a_{ij} легко определяются рекуррентным образом. Действительно, для диагональных элементов матрицы B справедливо соотношение

$$\begin{aligned} B_{ii} &= M[X_i - M_i]^2 = M \left[\left(\sum_{k=0}^i a_{ik} \xi_k + M_i \right) - M_i \right]^2 = \\ &= M \sum_{k=0}^i \sum_{l=0}^i a_{ik} a_{il} \xi_k \xi_l = \sum_{k=0}^i a_{ik}^2 = a_{ii}^2 + \sum_{k=0}^{i-1} a_{ik}^2, \end{aligned}$$

откуда

$$a_{ii} = \sqrt{B_{ii} - \sum_{k=0}^{i-1} a_{ik}^2}, \quad i = \overline{1, n-1}; \quad a_{00} = \sqrt{B_{00}}.$$

Для недиагональных элементов матрицы B выполняется равенство

$$B_{ij} = M(X_i - M_i)(X_j - M_j) = M\left(\sum_{k=0}^i a_{ik} \xi_k\right)\left(\sum_{l=0}^j a_{jl} \xi_l\right) = \sum_{k=0}^i \sum_{l=0}^j a_{ik} a_{jl} \underbrace{M \xi_k \xi_l}_{\delta_{kl}}.$$

Предполагая, что $i < j$, получаем

$$B_{ij} = \sum_{k=0}^i a_{ik} a_{jk} = \sum_{k=0}^{i-1} a_{ik} a_{jk} + a_{ii} a_{ji},$$

откуда

$$a_{ji} = \frac{1}{a_{ii}} \left(B_{ij} - \sum_{k=0}^{i-1} a_{ik} a_{jk} \right), \quad 1 \leq i < j \leq n-1; \quad a_{j0} = \frac{B_{0j}}{a_{00}}, \quad j = \overline{1, n-1}.$$

В частном случае, когда случайный вектор является двумерным ($n=2$), получаем следующие выражения для элементов матрицы преобразования:

$$a_{00} = \sqrt{B_{00}}, \quad a_{10} = \frac{B_{01}}{\sqrt{B_{00}}}, \quad a_{11} = \sqrt{B_{11} - \frac{B_{01}^2}{B_{00}}}.$$

Заметим, что поскольку для элементов матрицы B справедливо неравенство $|B_{ij}| \leq \sqrt{B_{ii} B_{jj}}$, ($i, j = \overline{0, n-1}$), то все коэффициенты a_{ij} корректно определены в том смысле, что подкоренные выражения в приведенных соотношениях всегда неотрицательны.

1.1.2 Оценивание параметров нормального закона распределения

Если n -мерный случайный вектор \bar{X} имеет нормальный закон распределения $N(\bar{M}, B)$, то оценки максимального правдоподобия его математического ожидания $\hat{\bar{M}}$ и корреляционной матрицы \hat{B} , рассчитанные по выборке $\bar{x}_1, \dots, \bar{x}_N$ объема N , выглядят следующим образом:

$$\hat{\bar{M}} = \frac{1}{N} \sum_{i=1}^N \bar{x}_i, \quad \hat{B} = \frac{1}{N} \sum_{i=1}^N (\bar{x}_i - \hat{\bar{M}})(\bar{x}_i - \hat{\bar{M}})^T = \frac{1}{N} \sum_{i=1}^N \bar{x}_i \bar{x}_i^T - \hat{\bar{M}} \hat{\bar{M}}^T.$$

1.1.3 Меры близости нормальных распределений

Пусть $f_0(\bar{x})$ и $f_1(\bar{x})$ – плотности вероятностей нормально распределенного случайного вектора с параметрами:

$$f_0 \sim N(\bar{M}_0, B_0) \text{ и } f_1 \sim N(\bar{M}_1, B_1).$$

Мерой близости распределений $f_0(\bar{x})$ и $f_1(\bar{x})$ является *расстояние Бхатачария*, вычисляемое по формуле

$$\rho_B = \frac{1}{4} (\bar{M}_1 - \bar{M}_0)^T \left(\frac{B_1 + B_0}{2} \right)^{-1} (\bar{M}_1 - \bar{M}_0) + \frac{1}{2} \ln \frac{\left| \frac{B_1 + B_0}{2} \right|}{\sqrt{|B_1| \cdot |B_0|}}. \quad (1.1)$$

Для случая равных корреляционных матриц ($B_1 = B_0 = B$) в качестве меры близости распределений используют расстояние Махаланобиса между векторами средних двух нормальных распределений:

$$\rho_M(\bar{M}_0, \bar{M}_1) = (\bar{M}_1 - \bar{M}_0)^T B^{-1} (\bar{M}_1 - \bar{M}_0), \quad (1.2)$$

которое в этой ситуации с точностью до постоянного множителя совпадает с расстоянием Бхатачария. Если компоненты случайного вектора \bar{X} независимы и одинаково распределены, то есть корреляционная матрица удовлетворяет условию $B=D_X I$, где I – единичная $N \times N$ матрица, а D_X – дисперсия компонент случайного вектора, то близость нормальных распределений в смысле расстояния Махаланобиса и, соответственно, Бхатачария эквивалентна близости в смысле евклидова расстояния между векторами средних:

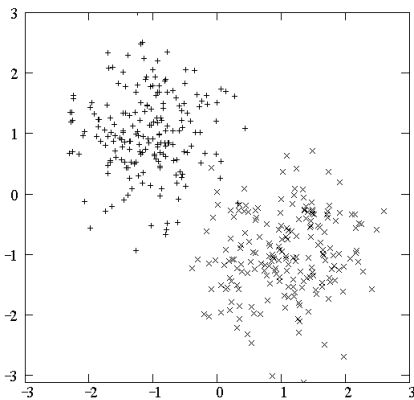
$$\rho_E(\bar{M}_0, \bar{M}_1) = \|\bar{M}_1 - \bar{M}_0\|^2 = (\bar{M}_1 - \bar{M}_0)^T (\bar{M}_1 - \bar{M}_0).$$

Использование метрик Бхатачария или Махаланобиса в общем случае предпочтительнее евклидовой, поскольку они учитывают как дисперсии отдельных компонент случайного вектора, так и их взаимные корреляции. При этом они обладают следующим важным свойством.

Утверждение. *Расстояния Бхатачария и Махаланобиса инвариантны относительно любого невырожденного линейного преобразования случайного вектора.*

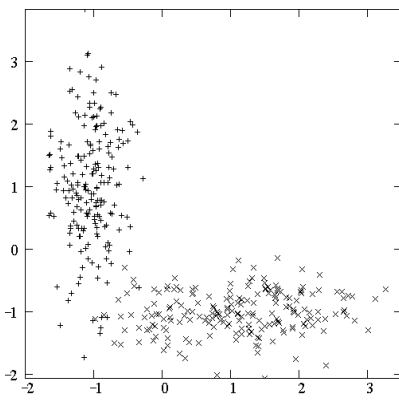
Действительно, пусть вектор \bar{Y} получен в результате линейного преобразования нормально распределенного случайного вектора \bar{X} : $\bar{Y} = C\bar{X} + \bar{E}$, где C – матрица преобразования с отличным от нуля определителем ($|C| \neq 0$), отвечающая за поворот и масштабирование координатных осей, а \bar{E} – вектор, определяющий смещения начала координат. Случайный вектор \bar{Y} оказывается также распределенным нормально с параметрами:

$$\bar{M}_l^Y = C\bar{M}_l^X + \bar{E}, \quad B_l^Y = CB_l^X C^T, \quad (l=0,1). \quad (1.3)$$



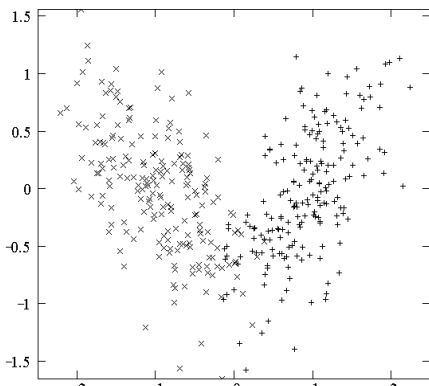
$$B_0 = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}, B_1 = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix};$$

$$\rho_B = 5, \rho_M = 20$$



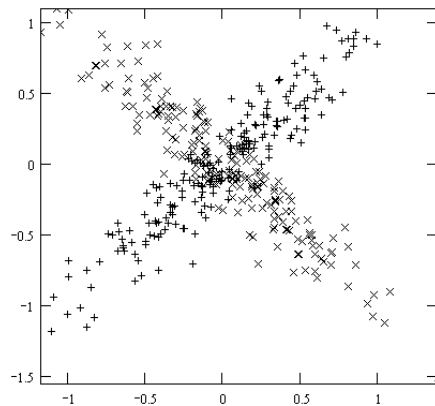
$$B_0 = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}, B_1 = \begin{bmatrix} b & 0 \\ 0 & a \end{bmatrix};$$

$$\rho_B = 5, \rho_M = 18$$



$$B_0 = \begin{bmatrix} a & -b \\ -b & a \end{bmatrix}, B_1 = \begin{bmatrix} a & b \\ b & a \end{bmatrix};$$

$$\rho_B = 3.5, \rho_M = 14$$



$$B_0 = \begin{bmatrix} a & -b \\ -b & a \end{bmatrix}, B_1 = \begin{bmatrix} a & b \\ b & a \end{bmatrix}.$$

$$\rho_B = 1, \rho_M = 0$$

Рисунок 1.1 – Примеры реализаций нормально распределенных случайных векторов (f_0 – "x", f_1 – "+", $a > b > 0$)

Подставляя (1.3) в выражения для расстояний (1.1) и (1.2) и учитывая справедливость тождеств для произвольных невырожденных матриц C и B

$$|BC| = |B||C|, \quad (BC)^T = C^T B^T, \quad (BC)^{-1} = C^{-1} B^{-1},$$

убеждаемся в справедливости приведенного утверждения.

1.1.4 Моделирование бинарных случайных векторов с независимыми координатами

Пусть $\bar{X} = (X_0, \dots, X_{n-1})^T$ – n -мерный бинарный случайный вектор, компоненты которого принимают одно из двух значений $\{0, 1\}$. Закон распределения бинарного случайного вектора задается совокупностью вероятностей $P(\bar{X} = \bar{x})$ для всех возможных значений $\bar{x} = (x_0, \dots, x_{n-1})^T$ вектора. Если координаты вектора \bar{X} независимы, то распределение вероятностей записывается в виде

$$P(\bar{X} = \bar{x}) = \prod_{i=0}^{n-1} P(X_i = x_i) = \prod_{i=0}^{n-1} (p_i x_i + (1-p_i)(1-x_i)),$$

где $p_i = P(X_i = 1)$. Таким образом, для формирования одной реализации бинарного случайного вектора с независимыми координатами необходимо получить по одной реализации каждой из n бинарных случайных величин X_i ($i=0, n-1$).

Способ моделирования бинарной случайной величины X с распределением вероятностей $P(X=1)=p$, $P(X=0)=1-p$ основан на следующих очевидных соотношениях:

$$P\{0 \leq U < 1-p\} = 1-p, \quad P\{1-p < U \leq 1\} = p,$$

где U – равномерно распределенная на отрезке $[0,1]$ случайная величина: $U \sim R[0,1]$. Таким образом случайные величины U и X связаны соотношением

$$X = [U/(1-p)],$$

где $[...]$ – целая часть числа. Следовательно, компоненты одной реализации искомого вектора могут быть получены по формуле

$$x_i = [u_i/(1-p_i)], \quad i = \overline{0, n-1},$$

здесь u_i – независимые реализации случайной величины U .

1.2 Порядок выполнения лабораторной работы

1.2.1 Исходные данные

- Вариант задания (предоставляется преподавателем).
- Математические ожидания для трех наборов двумерных нормально распределенных случайных векторов (из соответствующего варианта задания, см. раздел 7).
- Два бинарных вектора (из соответствующего варианта задания, см. раздел 7).

1.2.2 Общий план выполнения работы

1. Разработать алгоритм моделирования нормально распределенного случайного вектора с заданными математическим ожиданием и корреляционной матрицей.
2. Смоделировать и изобразить графически обучающие выборки объема $N=200$ для двух нормально распределенных двумерных случайных векторов с заданными математическими ожиданиями и самостоятельно подобранными равными корреляционными матрицами.

3. Смоделировать и изобразить графически обучающие выборки объема $N=200$ для трех нормально распределенных двумерных случайных векторов с заданными математическими ожиданиями и с неравными корреляционными матрицами, которые выбрать самостоятельно.
4. На основании полученных выборок найти точечные оценки параметров нормального закона для каждого из распределений.
5. Смоделировать обучающие выборки объема $N=200$ двух бинарных случайных векторов с распределениями, которые обеспечивают вероятность изменения указанной в представителе компоненты случайного вектора, равной $p=0.3$.

1.2.3 Содержание отчета

Отчет по работе должен содержать:

- исходные параметры моделируемых нормальных распределений; их оценки, полученные по обучающим выборкам, расстояния Бхатачария и Махаланобиса;
- графическое изображение значений векторов и имена файлов (с расширением .пру), в которые они записаны;
- распределения бинарных случайных векторов и имена записанных файлов, содержащих их реализации (с расширением .пру).

2 ОПТИМАЛЬНЫЕ СТРАТЕГИИ ТЕОРИИ СТАТИСТИЧЕСКОГО РАСПОЗНАВАНИЯ ОБРАЗОВ

Цель работы – изучение теоретических основ и экспериментальное исследование методов построения классификаторов, соответствующих оптимальным стратегиям принятия решений.

В лабораторной работе изучаются методы построения классификаторов, основанные на байесовской стратегии, минимаксной стратегии и стратегии Неймана-Пирсона, то есть при наличии различного количества априорной информации. В качестве примеров приводятся типовые задачи распознавания образов.

2.1 Теоретические основы лабораторной работы

2.1.1 Постановка задачи классификации

Пусть задано некоторое множество из I подлежащих распознаванию *объектов*:

$$\Omega = \{\omega_0, \omega_1, \dots, \omega_{I-1}\},$$

и задано его разбиение на L непересекающихся подмножеств, называемых в дальнейшем *образами* или *классами*:

$$P_\Omega = \{\Omega_0, \Omega_1, \dots, \Omega_{L-1}\}, \quad \bigcup_{l=0}^{L-1} \Omega_l = \Omega.$$

Пусть каждый из объектов $\omega \in \Omega$ представляется набором числовых характеристик, называемым *вектором признаков*:

$$\bar{x} = (x_0, x_1, \dots, x_{N-1})^T.$$

Задача классификации заключается в отыскании решающего правила, которое по заданному вектору признаков $\bar{x}(\omega)$ указыва-

ет, какому классу Ω_l принадлежит соответствующий объект ω . Построение такого решающего правила эквивалентно разбиению метрического пространства признаков $D = \{\bar{x} : \bar{x} \in D\}$ на множество непересекающихся областей:

$$P_D = \{D_0, D_1, \dots, D_{L-1}\}, \bigcup_{l=0}^{L-1} D_l = D. \quad (2.1)$$

При этом решение о принадлежности некоторого объекта $\omega \in \Omega$ к классу Ω_l принимается в том случае, если соответствующий объекту вектор признаков $\bar{x}(\omega) \in D$ принадлежит области D_l .

Решающее правило, предназначенное для указания, какой области D_l признакового пространства D принадлежит предъявленный вектор признаков \bar{x} , называется классификатором.

В идеале классификатор должен быть таким, чтобы области, выделяемые в пространстве признаков, соответствовали классам, то есть в идеале для элементов множеств P_Ω и P_D должно выполняться следующее условие: объект ω принадлежит классу Ω_l тогда и только тогда, когда соответствующий объекту вектор признаков $\bar{x}(\omega)$ принадлежит области D_l :

$$\forall \omega \in \Omega: \omega \in \Omega_l \Leftrightarrow \bar{x}(\omega) \in D_l. \quad (2.2)$$

Как правило, на практике данное условие не выполняется, и существует вероятность неверно проклассифицировать объект или допустить ошибку при распознавании.

Обозначим далее

$$p_{lj} \quad (l, j = \overline{0, L-1})$$

вероятность того, что классификатор принимает решение об отношении вектора признаков некоторого объекта к области D_j , в то время как сам объект принадлежит классу Ω_l :

$$p_{lj} = P(\bar{X} \in D_j / \Omega_l). \quad (2.3)$$

При $l \neq j$ вероятности p_{lj} характеризуют ошибки распознавания и называются *вероятностями неверной* или *ошибочной классификации*, а вероятности p_{ll} определяют вероятности *верной* (*правильной*) *классификации* представителей соответствующего класса. Уменьшение вероятностей ошибочной классификации – это основная задача, которая возникает при построении классификатора.

Обычно классификатор задается не в виде областей признакового пространства (2.1), а в виде набора так называемых *дискриминантных* или *решающих функций* $d_l(\bar{x}(\omega))$, ($l = \overline{0, L-1}$). При этом процесс принятия решения осуществляется по следующему правилу: объект считается принадлежащим тому классу, дискриминантная функция которого для соответствующего вектора признаков является максимальной:

$$\forall j \neq l: d_l(\bar{x}(\omega)) \geq d_j(\bar{x}(\omega)) \Rightarrow \bar{x}(\omega) \in D_l. \quad (2.4)$$

Замечание 1. Выбор решающих функций не единственен. Наряду с функциями $d_l(\bar{x})$, ($l = \overline{0, L-1}$) решающими функциями также являются:

- $g_1(\bar{x})d_l(\bar{x}) + g_2(\bar{x})$, где $g_1(\bar{x})$ – любая неотрицательная функция, а $g_2(\bar{x})$ – любая функция, не зависящая от номера класса l ;
- $\phi(d_l(\bar{x}))$, где $\phi(\dots)$ – любая монотонно возрастающая функция, не зависящая от номера класса.

Часто за счет приведенных преобразований удается существенно упростить вид классификатора.

2.1.2 Качество классификатора

Качество классификатора характеризуется величиной, называемой в теории статистических решений *условным средним риском*. Она задает среднюю величину потерь, связанных с принятием классификатором решения об отнесении данного вектора признаков \bar{x} к классу с номером j :

$$R_j(\bar{x}) = \frac{1}{f(\bar{x})} \sum_{l=0}^{L-1} c_{lj} P(\Omega_l) f(\bar{x}/\Omega_l). \quad (2.5)$$

В данном выражении:

- $P(\Omega_l)$ – априорная вероятность появления объектов из класса Ω_l ;
- $f(\bar{x}/\Omega_l)$ – условная плотность вероятностей случайного вектора признаков \bar{X} для объектов класса Ω_l ;
- $f(\bar{x})$ – безусловная плотность вероятностей случайного вектора \bar{X} ;
- элементы квадратной матрицы

$$C = \| \| c_{lj} \|_{l,j=0}^{L-1} \quad (2.6)$$

характеризуют величины *штрафов* или *потерь* за ошибки классификатора. Матрица C может быть достаточно произвольной. Единственным ограничением на ее элементы является то, что штраф за ошибочное решение должен быть больше, чем штраф за решение правильное, то есть $c_{lj} > c_{ll}$.

Интегральной величиной, характеризующей качество классификатора, является *математическое ожидание потерь* или *общий риск*, который с учетом (2.5) и (2.3) имеет вид

$$R = \sum_{j=0}^{L-1} \int_{D_j} R_j(\bar{x}) f(\bar{x}) d\bar{x} = \sum_{j=0}^{L-1} \sum_{l=0}^{L-1} c_{lj} P(\Omega_l) p_{lj}. \quad (2.7)$$

Процесс классификации аналогичен игре двух лиц с нулевой суммой, в которой одним из игроков является классификатор. В такой игре выигрыш (проигрыш) одного из участников равен проигрышу (выигрышу) другого. Выбор оптимальной стратегии в игре зависит от количества исходной информации. Могут использоваться *байесовская*, *минимаксная* стратегии или *стратегия Неймана-Пирсона*. В зависимости от того, какая из стратегий используется для построения классификатора, последний называют, соответственно, *байесовским классификатором*, *минимаксным классификатором* или *классификатором Неймана-Пирсона*.

2.1.3 Оптимальные стратегии классификации: классификатор Байеса

Байесовская стратегия используется при наличии полной априорной информации о классах, то есть когда известны:

- функции правдоподобия для каждого из классов;
- матрица штрафов;
- априорные вероятности для каждого из классов.

Стратегия решения выбирается таким образом, чтобы обеспечить минимум общего риска (2.7). Минимальный общий риск при этом называется *байесовским риском*. В соответствии с выражениями (2.5) и (2.7) минимум общего риска R будет обеспечен, если разбиение пространства признаков D будет осуществляться по следующему правилу: вектор $\bar{x} \in D$ относится к области D_l толь-

ко тогда, когда соответствующий условный средний риск $R_l(\bar{x})$ минимален:

$$\forall j \neq l \quad R_l(\bar{x}) < R_j(\bar{x}) \Rightarrow \bar{x} \in D_l. \quad (2.8)$$

Графическая иллюстрация байесовской стратегии приведена на рисунке 2.1а.

Если матрица потерь (2.6) является простейшей¹, то после подстановки в (2.8) выражения для условного среднего риска (2.5) имеем следующий явный вид байесовского классификатора (рисунк 2.1б):

$$\forall j \neq l \quad P(\Omega_l) f(\bar{x}/\Omega_l) \geq P(\Omega_j) f(\bar{x}/\Omega_j) \Rightarrow \bar{x} \in D_l. \quad (2.9)$$

Из (2.9), в частности, видно, что решающими функциями байесовского классификатора являются функции

$$d_l(\bar{x}) = P(\Omega_l) f(\bar{x}/\Omega_l), \quad l = \overline{0, L-1}. \quad (2.10)$$

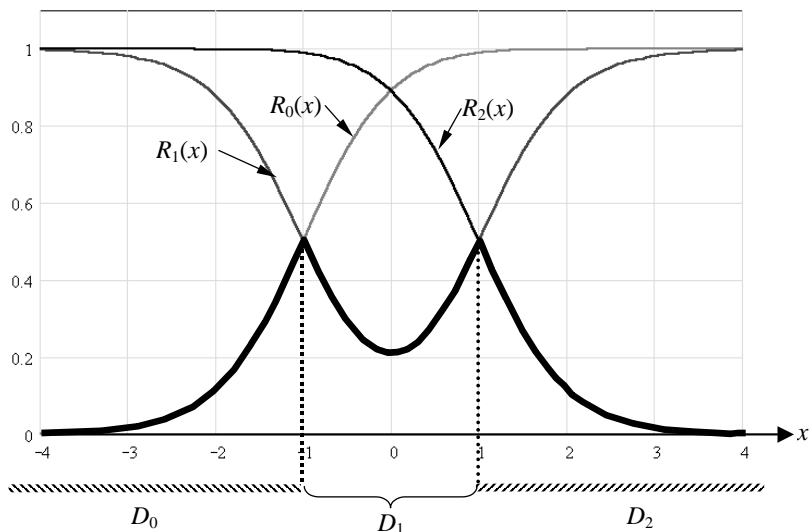
Часто используют также следующую форму записи байесовского классификатора:

$$\forall j \neq l \quad \frac{f(\bar{x}/\Omega_l)}{f(\bar{x}/\Omega_j)} \geq \frac{P(\Omega_j)}{P(\Omega_l)} \Rightarrow \bar{x} \in D_l. \quad (2.11)$$

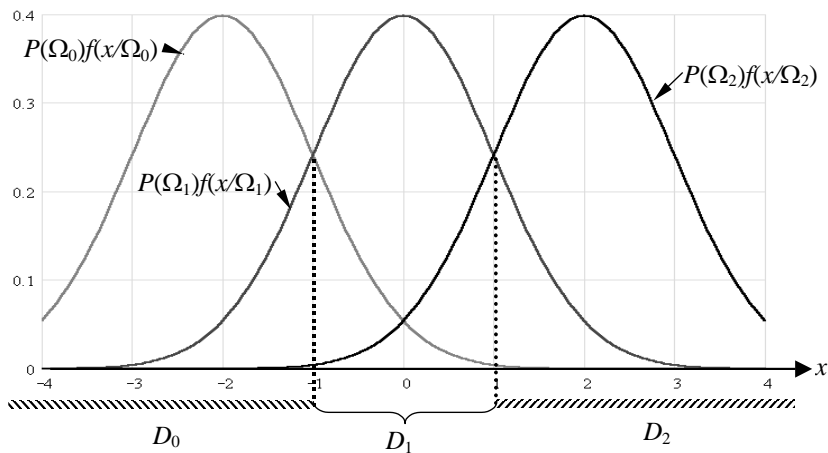
При этом функция $\Lambda_{ij}(\bar{x}) = \frac{f(\bar{x}/\Omega_l)}{f(\bar{x}/\Omega_j)}$ называется *отношением*

правдоподобия, а величина $\lambda_{jl} = \frac{P(\Omega_j)}{P(\Omega_l)}$ – *пороговым значением*.

¹ Матрица потерь C называется *простейшей*, если ее элементы $c_{ij} = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases}$.



а



б

Рисунок 2.1 – Построение байесовского классификатора для простейшей матрицы штрафов: а – байесовская стратегия минимизации общего риска; б – байесовская классификатор

Таким образом, байесовский классификатор основан на сравнении отношения правдоподобия с пороговым значением:

$$\forall j \neq l \quad \Lambda_{jl}(\bar{x}) \geq \lambda_{jl} \Rightarrow \bar{x} \in D_j,$$

и называется поэтому классификатором отношения правдоподобия.

Легко показать, что при произвольном виде матрицы штрафов в случае двух классов байесовский классификатор имеет вид

$$\begin{aligned} \frac{f(\bar{x}/\Omega_1)}{f(\bar{x}/\Omega_0)} > \frac{P(\Omega_0)(c_{01}-c_{00})}{P(\Omega_1)(c_{10}-c_{11})} &\Rightarrow \bar{x} \in \begin{cases} D_1 \\ D_0 \end{cases} \end{aligned}$$

с дискриминантными функциями:

$$d_j(\bar{X}) = P(\Omega_j)(c_{j(1-j)} - c_{jj})f(\bar{x}/\Omega_j), \quad j = \overline{0,1}.$$

2.1.4 Оптимальные стратегии классификации: минимаксный классификатор

Классификатор, основанный на минимаксной стратегии, используется для случая двух классов и если известны:

- функции правдоподобия для каждого из классов;
- матрица штрафов.

Минимизировать величину общего риска при отсутствии информации об априорных вероятностях классов, очевидно, невозможно. В то же время, предполагая возможность произвольного изменения значений априорных вероятностей классов, можно минимизировать максимально возможное значение риска. Действительно, общий риск (2.7) в случае двух классов может быть представлен в следующем виде:

$$R = (c_{11} + p_{10}(c_{10} - c_{11})) + P(\Omega_0) \cdot [(c_{00} + p_{01}(c_{01} - c_{00})) - (c_{11} + p_{10}(c_{10} - c_{11}))]. \quad (2.12)$$

При фиксированном классификаторе изменение априорной вероятности приводит к изменению величины общего риска, причем характер зависимости в (2.12) линейный (рисунок 2.2).

Поэтому поиск классификатора, минимизирующего максимально возможную величину общего риска, эквивалентен поиску такого байесовского классификатора, для которого величина (2.12) является постоянной, не зависящей от значения априорной вероятности $P(\Omega_0)$ величиной. Таким классификатором, очевидно, является байесовский классификатор, удовлетворяющий следующему дополнительному условию:

$$(c_{00} + p_{01}(c_{01} - c_{00})) - (c_{11} + p_{10}(c_{10} - c_{11})) = 0. \quad (2.13)$$

Из рисунка 2.2 видно, что значение величины общего риска для минимаксного классификатора равно максимальному значению байесовского (минимального) риска. Пара априорных вероятностей $(P^*(\Omega_0), 1 - P^*(\Omega_0))$, при которых байесовский риск принимает максимальное значение, называется *наименее благоприятным распределением априорных вероятностей*.

Таким образом, минимаксный классификатор – это байесовский классификатор, полученный для пары наименее благоприятных априорных вероятностей.

В более простой ситуации, когда элементы матрицы штрафов таковы, что

$$c_{00} = c_{11} = 0, \quad c_{10} = c_1, \quad c_{01} = c_0,$$

условие (2.13) преобразуется в следующее:

$$p_{01}c_0 = p_{10}c_1. \quad (2.14)$$

Последнее выражение представляет собой условие выбора областей D_0, D_1 в байесовском классификаторе.

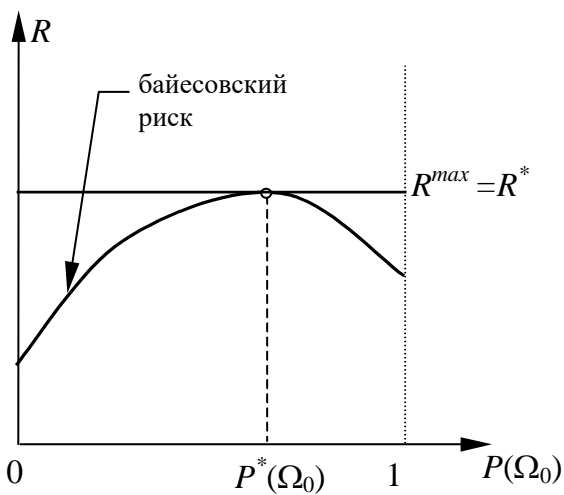
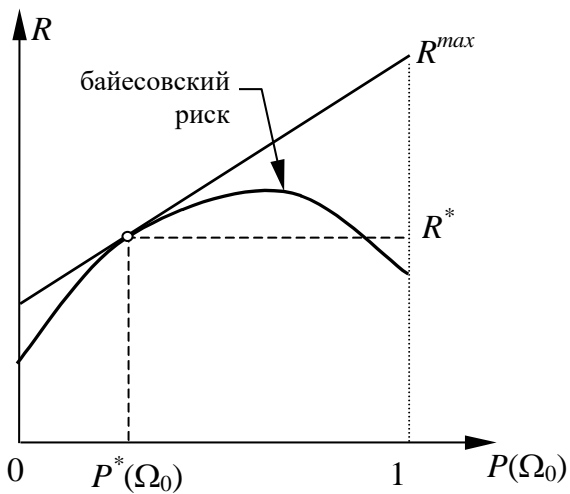


Рисунок 2.2 – Иллюстрация минимаксной стратегии построения классификатора

2.1.5 Оптимальные стратегии классификации: классификатор Неймана-Пирсона

Классификатор, основанный на стратегии Неймана-Пирсона, используется для случая двух классов, и если известны только функции правдоподобия для каждого из классов. Суть стратегии Неймана-Пирсона состоит в следующем: задается допустимое значение вероятности ошибки первого рода¹ p_0 , а затем классификатор строится таким образом, чтобы обеспечить минимум вероятности ошибки второго рода p_1 :

$$\begin{cases} p_1 \rightarrow \min_{D_0, D_1} \\ p_0 = p_0^* \end{cases} \quad (2.15)$$

Решением задачи Неймана-Пирсона является классификатор вида

$$\Lambda(\bar{x}) = \frac{f(\bar{x}/\Omega_1)}{f(\bar{x}/\Omega_0)} > \lambda \Rightarrow \begin{cases} \bar{x} \in D_1 \\ \bar{x} \in D_0 \end{cases}, \quad (2.16)$$

где значение пороговой величины λ определяется, исходя из условия: $p_0 = p_0^*$ (Рисунок 2.3). Из выражения (2.16) следует, что

классификатор Неймана-Пирсона – это классификатор отношения правдоподобия.

¹ Критерий Неймана-Пирсона в теории статистических решений традиционно используется для проверки гипотез. Поскольку в классической постановке задачи используется только две возможные гипотезы, то различают два типа ошибок:

- ошибку первого рода p_0 – в контексте настоящего изложения $p_0 = p_{01}$,
- ошибку второго рода p_1 – в контексте настоящего изложения $p_1 = p_{10}$.

Заметим, что в общем случае $p_1 + p_0 \neq 1$. В дальнейшем изложении данная терминология и приведенные обозначения также используются.

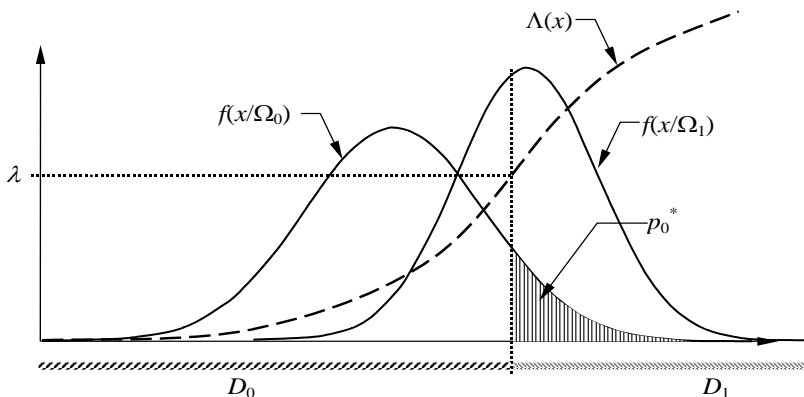


Рисунок 2.3 – Иллюстрация стратегии Неймана-Пирсона построения классификатора

2.1.6 Типовые решения оптимальных стратегий классификации: нормально распределенные вектора признаков

На практике часто возникает задача распознавания детерминированных объектов или сигналов в условиях помех. Она стала традиционной в таких дисциплинах, как теория сигналов, обработка изображений, распознавание образов. В настоящем и следующем разделах приведены два достаточно типичных примера постановки подобной задачи и ее решения с использованием байесовской стратегии.

Пусть входной сигнал, задаваемый вектором $\bar{x} = (x_0, \dots, x_{n-1})^T$ и подлежащий распознаванию, представляет собой аддитивную смесь детерминированной и шумовой составляющих. Будем считать, что наблюдаемые вектора имеют нормальный закон распределения в каждом из L классов, то есть имеют плотность вероятностей вида

$$f(\bar{x}/\Omega_l) = \frac{1}{(2\pi)^{n/2} \sqrt{|B_l|}} \exp\left(-\frac{1}{2}(\bar{x} - \bar{M}_l)^T B_l^{-1} (\bar{x} - \bar{M}_l)\right), \quad l=0, L-1. \quad (2.17)$$

Здесь

$$B_l = M \left((\bar{X} - \bar{M}_l)(\bar{X} - \bar{M}_l)^T / \Omega_l \right), \quad \bar{M}_l = M(\bar{X} / \Omega_l)$$

корреляционная матрица и математическое ожидание вектора признаков из класса Ω_l соответственно. Математические ожидания или *средние* характеризуют детерминированные составляющие распознаваемых сигналов, а корреляционные матрицы – характер шумовой составляющей. Считаются также известными априорные вероятности $P(\Omega_l)$ появления векторов из каждого класса. Требуется по реализации \bar{x} случайного вектора \bar{X} определить класс, к которому данный вектор принадлежит.

Решением данной задачи является байесовский классификатор с дискриминантными функциями следующего вида:

$$d_l(\bar{x}) = \ln P(\Omega_l) - \ln \sqrt{|B_l|} - \frac{1}{2} (\bar{x} - \bar{M}_l)^T B_l^{-1} (\bar{x} - \bar{M}_l), \quad l = \overline{0, L-1}. \quad (2.18)$$

Выражение (2.18) может быть существенно упрощено в некоторых частных случаях.

Случай 1

Предположим, что компоненты наблюдаемого вектора \bar{X} являются независимыми и имеют одинаковую дисперсию D_X , то есть $B_l = D_X I$, где I – единичная $N \times N$ матрица. Тогда законы распределения (2.17) отличаются только средними значениями, а решающие функции байесовского классификатора преобразуются к следующему виду:

$$d_l(\bar{x}) = 2D_X \ln P(\Omega_l) - \|\bar{x} - \bar{M}_l\|^2, \quad l = \overline{0, L-1}, \quad (2.19)$$

здесь $\|\dots\|$ – евклидова норма. При равных априорных вероятностях данное решающее правило приобретает очевидную трактовку:

вектор признаков \bar{x} относится к тому классу, расстояние до центра которого минимально.

Классификатор в этом случае называют *классификатором по минимуму евклидова расстояния*. Пример разбиения пространства признаков при использовании подобного классификатора для случая трех классов приведен на рисунке 2.4а.

Нетрудно видеть, что решающие функции (2.19) можно преобразовать к линейной форме:

$$d_l(\bar{x}) = \overline{M}_l^T \bar{x} - \frac{1}{2} \overline{M}_l^T \overline{M}_l + D_X \ln P(\Omega_l), \quad l = \overline{0, L-1}.$$

В этом случае *разделяющие границы* между различными областями D_l , задаваемые соотношениями вида

$$d_{lj}(\bar{x}) \equiv d_l(\bar{x}) - d_j(\bar{x}) = 0, \quad 0 \leq l < j \leq L-1,$$

также являются линейными:

$$d_{lj}(\bar{x}) = (\overline{M}_l - \overline{M}_j)^T \bar{x} - \frac{1}{2} (\overline{M}_l + \overline{M}_j)^T (\overline{M}_l - \overline{M}_j) + D_X \ln \frac{P(\Omega_l)}{P(\Omega_j)}, \quad 0 \leq l < j \leq L-1.$$

Случай 2

Предположим, что все корреляционные матрицы одинаковы: $B_l = B$. Тогда решающие функции байесовского классификатора представимы в виде

$$d_l(\bar{x}) = 2 \ln P(\Omega_l) - (\bar{x} - \overline{M}_l)^T B^{-1} (\bar{x} - \overline{M}_l), \quad l = \overline{0, L-1}.$$

Величина

$$\rho(\bar{x}, \overline{M}_l) = (\bar{x} - \overline{M}_l)^T B^{-1} (\bar{x} - \overline{M}_l) \quad (2.20)$$

называется *расстоянием Махаланобиса* между векторами \bar{x} и \bar{M}_l и является мерой близости вектора \bar{x} к центру класса Ω_l , учитывающей как дисперсии компонент вектора \bar{X} , так и их взаимную корреляцию. Очевидно, что в данной ситуации классификатор снова оказывается классификатором по минимуму расстояния Махаланобиса (рисунок 2.4б). Кроме того, и решающие функции, и разделяющие границы снова являются линейными:

$$d_l(\bar{x}) = \bar{M}_l^T B^{-1} \bar{x} - \frac{1}{2} \bar{M}_l^T B^{-1} \bar{M}_l + \ln P(\Omega_l), \quad l = \overline{0, L-1},$$

$$d_{lj}(\bar{x}) = (\bar{M}_l - \bar{M}_j)^T B^{-1} \bar{x} - \frac{1}{2} (\bar{M}_l + \bar{M}_j)^T B^{-1} (\bar{M}_l - \bar{M}_j) + \ln \frac{P(\Omega_l)}{P(\Omega_j)},$$

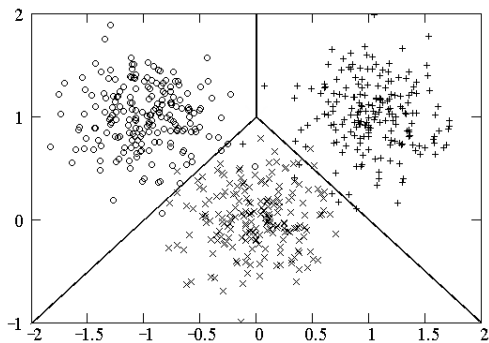
$$0 \leq l < j \leq L-1,$$

Случай 3

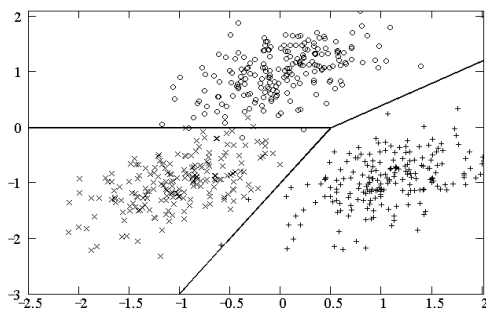
В ситуации, когда все корреляционные матрицы различны, необходимо пользоваться выражением (2.18) для дискриминантных функций. Разделяющие границы в этом случае представляются в следующем виде:

$$d_{ij}(\bar{x}) = \frac{1}{2} \bar{x}^T (B_j^{-1} - B_l^{-1}) \bar{x} + \left(\bar{M}_l^T B_l^{-1} - \bar{M}_j^T B_j^{-1} \right) \bar{x} + \left[\frac{1}{2} \bar{M}_j^T B_j^{-1} \bar{M}_j - \frac{1}{2} \bar{M}_l^T B_l^{-1} \bar{M}_l + \frac{1}{2} \ln \frac{|B_j|}{|B_l|} + \ln \frac{P(\Omega_l)}{P(\Omega_j)} \right] = 0 \quad (\forall j \neq l \Rightarrow \bar{x} \in D_l)$$

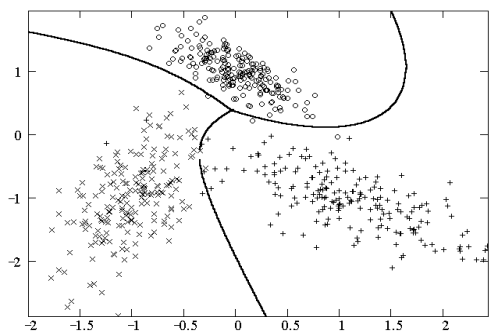
и являются, очевидно, квадратичными функциями. Такие границы называются *гиперквадриками* (гиперсферы, гиперпараболы и т.д., пример их приведен на рисунке 2.4в), а сам классификатор называется *квадратичным*.



а



б



в

Рисунок 2.4 – Байесовский классификатор в случае нормально распределенных векторов признаков: а – признаки статистически независимые и одинаково распределенные; б – корреляционные матрицы одинаковы; в – корреляционные матрицы различны

2.1.7 Типовые решения оптимальных стратегий классификации: бинарные вектора признаков

На практике достаточно часто возникает задача распознавания векторов признаков, компоненты которых являются бинарными. Эта задача, в частности, решается при автоматическом распознавании печатного текста в известных системах CuneiForm и FineReader. Ниже приведено ее решение с использованием байесовской стратегии.

Пусть закон распределения бинарного случайного вектора \bar{X} для каждого из классов Ω_l ($l = \overline{0, L-1}$) задан распределением вероятностей $P(\bar{X} = \bar{x}/\Omega_l)$; пусть также известны априорные вероятности появления представителей каждого класса и матрица штрафов.

При наличии этой информации выражение для условного среднего риска (2.5) переписывается с учетом дискретного характера вектора признаков в следующем виде:

$$R_j(\bar{x}) = \frac{1}{P(\bar{X} = \bar{x})} \sum_{l=0}^{L-1} c_{lj} P(\Omega_l) P(\bar{X} = \bar{x}/\Omega_l).$$

Предположим, что матрица штрафов является простейшей. Тогда байесовский классификатор может быть записан в одной из двух форм: либо в терминах дискриминантных функций (2.10), либо в терминах отношения правдоподобия (2.11). С учетом дискретного характера вектора признаков эти выражения имеют следующий вид:

$$\forall j \neq l \quad P(\Omega_l) P(\bar{X} = \bar{x}/\Omega_l) = d_l(\bar{x}) \geq d_j(\bar{x}) = P(\Omega_j) P(\bar{X} = \bar{x}/\Omega_j) \Rightarrow \bar{x} \in D_l,$$

$$\forall j \neq l \quad \frac{P(\bar{X} = \bar{x}/\Omega_l)}{P(\bar{X} = \bar{x}/\Omega_j)} = \Lambda_{lj} \geq \lambda_{jl} = \frac{P(\Omega_j)}{P(\Omega_l)} \Rightarrow \bar{x} \in D_l. \quad (2.21)$$

В общем случае аналитически получить окончательные выражения для байесовского классификатора не представляется возможным. Однако это может быть сделано в предположении независимости компонент вектора признаков. В этом случае

$$P(\bar{X} = \bar{x}/\Omega_l) = \prod_{i=0}^{N-1} P(X_i = x_i/\Omega_l).$$

Учитывая, что возможные значения компонент вектора “0” или “1”, получаем следующее выражение для дискриминантной функции:

$$d_l(\bar{x}) = P(\Omega_l) \prod_{i=0}^{N-1} \left((1 - P(X_i = 1/\Omega_l))(1 - x_i) + P(X_i = 1/\Omega_l)x_i \right), \quad l = \overline{0, L-1}.$$

Окончательным решением задачи является классификатор с дискриминантной функцией вида

$$d_l(\bar{x}) = \left[\ln(P(\Omega_l)) + \sum_{i=0}^{N-1} \ln(1 - P(X_i = 1/\Omega_l)) \right] + \sum_{i=0}^{N-1} x_i \cdot \ln \left(\frac{P(X_i = 1/\Omega_l)}{1 - P(X_i = 1/\Omega_l)} \right),$$

который, очевидно, является линейным. Аналогичным образом можно получить выражение для байесовского классификатора в терминах отношения правдоподобия (2.21):

$$\forall j \neq l \quad \tilde{\Lambda}_{lj}(\bar{x}) \geq \tilde{\lambda}_{jl} \Rightarrow \bar{x} \in D_l,$$

где

$$\begin{aligned} \tilde{\Lambda}_{lj}(\bar{x}) &= \sum_{i=0}^{N-1} w_{lj}^i \cdot x_i, \\ w_{lj}^i &= \ln \left(\frac{P(X_i = 1/\Omega_l)}{1 - P(X_i = 1/\Omega_l)} \cdot \frac{1 - P(X_i = 1/\Omega_j)}{P(X_i = 1/\Omega_j)} \right) \\ \tilde{\lambda}_{jl} &= \ln \left(\frac{P(\Omega_j)}{P(\Omega_l)} \right) + \sum_{i=0}^{N-1} \ln \left(\frac{1 - P(X_i = 1/\Omega_j)}{1 - P(X_i = 1/\Omega_l)} \right). \end{aligned} \quad (2.22)$$

Очевидно, отношение правдоподобия также является линейной функцией компонент вектора признаков. Пример байесовского классификатора в терминах отношения правдоподобия приведен на рисунке 2.5.

2.1.8 Вычисление вероятностей ошибочной классификации: общий случай

Эффективность любого классификатора характеризуется вероятностями ошибок. Однако их нахождение в общем случае оказывается достаточно сложной задачей, поскольку требует вычисления многомерных интегралов:

$$p_{lj} = \int_{D_j} f(\bar{x}/\Omega_l) d\bar{x}, \quad l \neq j, \quad l, j = \overline{0, L-1}. \quad (2.23)$$

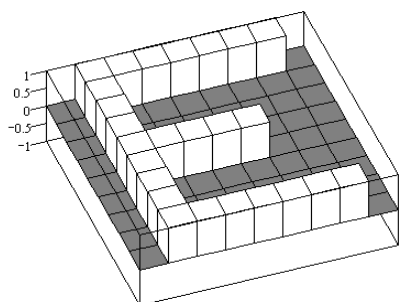
При использовании байесовского классификатора, который является классификатором отношения правдоподобия, многомерный интеграл (2.23) может быть заменен одномерным от плотности вероятностей отношения правдоподобия Λ_{ij} в каждом из классов. В частности, в случае двух классов для вероятностей ошибок имеем следующие выражения:

$$p_0 = \int_{\lambda}^{+\infty} f_{\Lambda}(u/\Omega_0) du, \quad p_1 = \int_{-\infty}^{\lambda} f_{\Lambda}(u/\Omega_1) du, \quad (2.24)$$

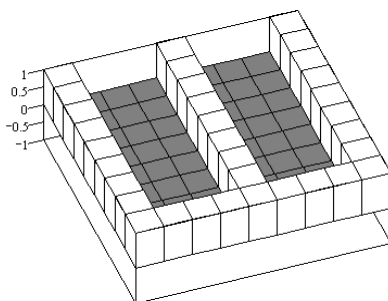
где

$$\Lambda = \Lambda(\bar{X}) = \frac{f(\bar{X}/\Omega_1)}{f(\bar{X}/\Omega_0)}, \quad \lambda = \frac{P(\Omega_0)}{P(\Omega_1)},$$

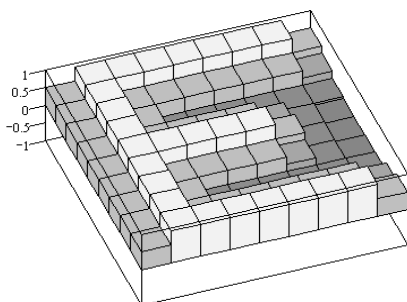
здесь λ – пороговое значение. Плотность вероятностей отношения правдоподобия удастся найти далеко не всегда. Однако, когда случайный вектор \bar{X} имеет нормальный закон распределения, это может быть сделано.



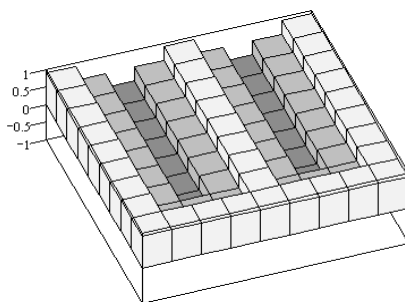
а



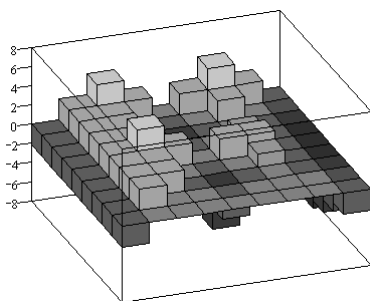
б



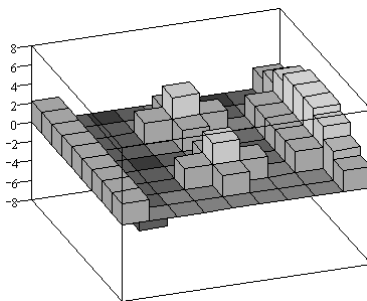
в



г



д



е

Рисунок 2.5 – Пример байесовского классификатора для бинарных векторов признаков: а – представитель класса Ω_0 ; б – представитель класса Ω_1 ; в – распределение $P(X_{ij}=1/\Omega_0)$; г – распределение $P(X_{ij}=1/\Omega_1)$; д – компоненты вектора \bar{w}_{01} ; е – компоненты вектора $(-\bar{w}_{01})$

2.1.9 Вычисление вероятностей ошибочной классификации: нормально распределенные вектора признаков

Пусть вектор признаков в каждом из двух классов характеризуется нормальным законом распределения, причем все корреляционные матрицы являются равными $B_l = B$ ($l=0,1$). Тогда случайная величина $\tilde{\Lambda} = \ln(\Lambda(\bar{X}))$ имеет нормальный закон распределения с параметрами:

$$\begin{aligned} M(\tilde{\Lambda}/\Omega_0) &= M(\ln(\Lambda(\bar{X}))/\Omega_0) = -\frac{1}{2}\rho(\bar{M}_0, \bar{M}_1), \\ M(\tilde{\Lambda}/\Omega_1) &= M(\ln(\Lambda(\bar{X}))/\Omega_1) = \frac{1}{2}\rho(\bar{M}_0, \bar{M}_1), \\ D(\tilde{\Lambda}/\Omega_l) &= D(\ln(\Lambda(\bar{X}))/\Omega_l) = \rho(\bar{M}_0, \bar{M}_1), \quad l=0,1, \end{aligned}$$

где $\rho(\bar{M}_0, \bar{M}_1)$ – расстояние Махаланобиса между векторами средних \bar{M}_0 и \bar{M}_1 . Выражения для вероятностей ошибок (2.24) преобразуются к следующему виду:

$$p_0 = 1 - \Phi\left(\frac{\tilde{\lambda} + \frac{1}{2}\rho(\bar{M}_0, \bar{M}_1)}{\sqrt{\rho(\bar{M}_0, \bar{M}_1)}}\right), \quad p_1 = \Phi\left(\frac{\tilde{\lambda} - \frac{1}{2}\rho(\bar{M}_0, \bar{M}_1)}{\sqrt{\rho(\bar{M}_0, \bar{M}_1)}}\right),$$

где $\Phi(\dots)$ – функция Лапласа, а

$$\tilde{\lambda} = \ln \lambda = \ln\left(\frac{P(\Omega_0)(c_{01} - c_{00})}{P(\Omega_1)(c_{10} - c_{11})}\right)$$

новая пороговая величина (рисунок 2.6). В частном случае, когда матрица штрафов является простейшей и априорные вероятности классов совпадают, тогда имеем:

$$\lambda=1, \tilde{\lambda}=0, p_0=1-\Phi\left(\frac{1}{2}\sqrt{\rho(\bar{M}_0, \bar{M}_1)}\right) \quad p_1=\Phi\left(-\frac{1}{2}\sqrt{\rho(\bar{M}_0, \bar{M}_1)}\right).$$

Общий риск при этом определяется нижеследующей формулой и монотонно убывает с ростом расстояния Махаланобиса между средними:

$$R=1-\Phi\left(\frac{1}{2}\sqrt{\rho(\bar{M}_0, \bar{M}_1)}\right).$$

Минимаксный классификатор

Предположим, что матрица штрафов имеет следующие элементы: $c_{01}=c_{10}=c$, $c_{00}=c_{11}=0$. Тогда соотношение (2.14) для выбора разделяющей границы байесовского классификатора, соответствующего минимаксной стратегии, превращается в равенство вероятностей ошибочной классификации: $p_0=p_1$. С учетом выражений (2.24) получаем, что пороговое значение для минимаксного классификатора: $\lambda=1$, $\tilde{\lambda}=0$.

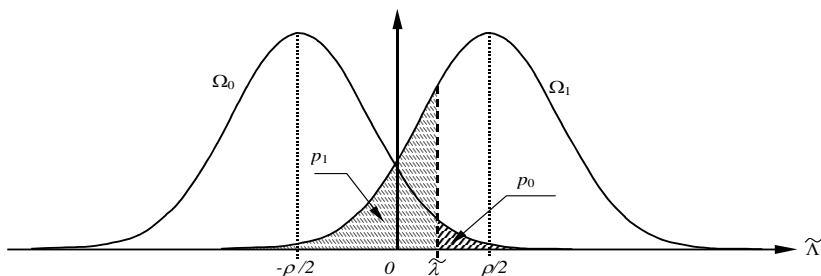


Рисунок 2.6 – Плотности вероятностей логарифма отношения правдоподобия для нормально распределенных признаков с равными корреляционными матрицами

Классификатор Неймана-Пирсона

Используя условие (2.15) и равенства (2.24), получаем, что пороговое значение классификатора Неймана-Пирсона определяется по формуле

$$\lambda = e^{\tilde{\lambda}}, \quad \tilde{\lambda} = -\frac{1}{2}\rho(\bar{M}_0, \bar{M}_1) + \sqrt{\rho(\bar{M}_0, \bar{M}_1)}\Phi^{-1}(1-p_0^*),$$

где p_0^* – заданная величина вероятности ошибки первого рода.

2.1.10 Вычисление вероятностей ошибочной классификации: бинарные вектора признаков

Получить аналитическое выражение для вероятностей ошибочной классификации бинарных векторов признаков в общем случае невозможно. Однако при небольшой размерности вектора признаков ($N \leq 26$) значения этих вероятностей можно вычислить на ПЭВМ, используя дискретный аналог формулы (2.23):

$$p_{lj} = \sum_{\bar{x} \in D_j} P(\bar{X} = \bar{x} / \Omega_l), \quad l \neq j, \quad l, j = 0, L-1,$$

поскольку в признаковом пространстве находится всего 2^N элементов. Когда число компонент вектора признаков велико, можно получить приближенные выражения для вероятностей ошибочной классификации, если предположить независимость компонент случайного вектора \bar{X} . В подобной ситуации в соответствии с центральной предельной теоремой можно считать закон распределения случайной величины $\tilde{\Lambda}_{lj} \equiv \tilde{\Lambda}_{lj}(\bar{X})$ нормальным. Для простоты рассмотрим ситуацию с разделением двух классов. В этом случае числовые характеристики закона распределения случайной величины $\tilde{\Lambda}_{10}$ имеют вид

$$\begin{aligned} m_l &= M(\tilde{\Lambda}_{10}(\bar{X}) / \Omega_l) = \\ &= \sum_{i=0}^{N-1} \ln \left(\frac{P(X_i = 1 / \Omega_1)}{1 - P(X_i = 1 / \Omega_1)} \frac{1 - P(X_i = 1 / \Omega_0)}{P(X_i = 1 / \Omega_0)} \right) P(X_i = 1 / \Omega_l), \quad l = 0, 1. \end{aligned}$$

$$\sigma_l^2 = D(\tilde{\Lambda}_{10}(\bar{X})/\Omega_l) = \sum_{i=0}^{N-1} \left[\ln \left(\frac{P(X_i=1/\Omega_1)}{1-P(X_i=1/\Omega_1)} \frac{1-P(X_i=1/\Omega_0)}{P(X_i=1/\Omega_0)} \right) \right]^2 P(X_i=1/\Omega_l)(1-P(X_i=1/\Omega_l)) ,$$

а для вероятностей ошибочной классификации получаем следующие приближенные выражения:

$$p_0 = 1 - \Phi \left(\frac{\tilde{\lambda} - m_0}{\sigma_0} \right), \quad p_1 = \Phi \left(\frac{\tilde{\lambda} - m_1}{\sigma_1} \right), \quad \text{где } \tilde{\lambda} = \ln \frac{P(\Omega_0)}{P(\Omega_1)} .$$

В ситуации, когда условия центральной предельной теоремы не выполняются, а компоненты вектора независимы, можно воспользоваться следующими выражениями для верхних границ вероятностей ошибок, вытекающих из неравенства Чебышева:

$$p_0 \leq \frac{\sigma_0^2}{(m_0 - \tilde{\lambda})^2}, \quad p_1 \leq \frac{\sigma_1^2}{(m_1 - \tilde{\lambda})^2} .$$

2.1.11 Экспериментальная оценка вероятностей ошибочной классификации

На практике воспользоваться аналитическими выражениями для вычисления вероятностей ошибок классификации чаще всего не представляется возможным. Поэтому единственным способом определения искомым вероятностей является их статистическое оценивание.

Пусть выборочные данные представлены в виде набора из N объектов $\{\omega_i\}_{i=0}^{N-1}$ класса Ω_0 и N рассчитанных по ним векторов признаков $\{\bar{x}(\omega_i)\}_{i=0}^{N-1}$ (при этом говорят, что задана обучающая выборка объема N из класса Ω_0), а также задан некоторый клас-

сификатор, производящий классификацию объектов в соответствии со следующим правилом:

$$d(\bar{x}(\omega)) \begin{matrix} > \\ < \end{matrix} 0 \Rightarrow \begin{cases} \omega \in \Omega_0 \\ \omega \notin \Omega_0 \end{cases}.$$

Обозначим p_0 – истинное значение вероятности ошибочной классификации объектов класса Ω_0 :

$$p_0 = P(d(\bar{X}/\Omega_0) < 0).$$

Наилучшей точечной оценкой вероятности p_0 , как известно, является относительная частота события $(d(\bar{X}/\Omega_0) < 0)$:

$$\hat{p}_0 = \frac{1}{N} \sum_{i=0}^{N-1} I(d(\bar{x}(\omega_i)) < 0). \quad (2.25)$$

Качество оценки (2.25) можно охарактеризовать величиной ее относительной погрешности, которая имеет вид

$$\varepsilon \equiv \frac{\sqrt{D[\hat{p}]}}{\hat{p}} = \sqrt{\frac{1-\hat{p}}{N\hat{p}}}.$$

Последнее выражение можно использовать также с целью определения необходимого объема N обучающей выборки для получения оценки вероятности с заранее заданной относительной погрешностью ε .

Замечание 2. Аналогично (2.26) выглядит оценка вероятности p_l ошибочной классификации объектов класса Ω_l и вероятностей p_{lj} ошибочной классификации объектов из класса Ω_l в класс Ω_j по обучающим выборкам из соответствующих классов.

2.2 Порядок выполнения лабораторной работы

2.2.1 Исходные данные

- Два файла данных, полученных в процессе выполнения первой лабораторной работы (см. раздел 1 настоящего издания) и содержащих наборы двумерных нормально распределенных векторов признаков для ситуации равных корреляционных матриц; параметры этих законов распределения.
- Три файла данных, полученных в процессе выполнения первой лабораторной работы (см. раздел 1 настоящего издания) и содержащих наборы двумерных нормально распределенных векторов признаков для ситуации неравных корреляционных матриц; параметры этих законов распределения.
- Два файла данных, полученных в процессе выполнения первой лабораторной работы (см. раздел 1 настоящего издания) и содержащие наборы бинарных векторов признаков, распределения вероятностей бинарных векторов.

2.2.2 Общий план выполнения работы

1. Построить байесовскую решающую границу между классами Ω_0 и Ω_1 двумерных нормально распределенных векторов признаков для случая равных корреляционных матриц и равных априорных вероятностей и изобразить ее графически. Вычислить вероятности ошибочной классификации и суммарную вероятность ошибочной классификации в этом случае.
2. Построить минимаксный классификатор и классификатор Неймана-Пирсона для вероятности ошибки первого рода $p_0^* = 0.05$ для двух классов Ω_0 и Ω_1 двумерных нормально распределенных векторов признаков в случае равных корреляционных матриц. Изобразить решающие границы полученных классификаторов графически.

3. Построить байесовскую решающую границу между классами Ω_0 , Ω_1 и Ω_2 двумерных нормально распределенных векторов признаков для неравных корреляционных матриц и равных априорных вероятностей. Изобразить полученные решающие границы графически. Для любых двух классов оценить экспериментально вероятности ошибочной классификации в этом случае и определить относительную погрешность полученных оценок для заданного объема обучающей выборки N . Определить объем обучающей выборки, обеспечивающий получение оценок вероятностей ошибочной классификации с погрешностью не более 5%.
4. Построить байесовскую разделяющую границу между классами Ω_0 и Ω_1 двумерных бинарных векторов признаков. Вычислить вероятности ошибочной классификации аналитически и оценить их экспериментально.

2.2.3 Содержание отчета

Отчет по работе должен содержать:

- Аналитические выражения для классификаторов, полученных в результате выполнения пп.1–3 плана, и графическое изображение соответствующих им решающих границ вместе с элементами обучающих выборок.
- Параметры классификатора, полученного в результате выполнения п.4 плана, и его графическое изображение.
- Вероятности ошибочной классификации для построенных в пп.1–4 плана классификаторов, найденные аналитически и экспериментально. Для первого случая привести расчетные формулы, для второго – относительную погрешность оценки и объем выборки, гарантирующий величину погрешности не более 5%.

3 ЛИНЕЙНЫЕ КЛАССИФИКАТОРЫ

Цель работы – изучение теоретических основ и экспериментальное исследование методов построения линейных классификаторов для распознавания образов.

В лабораторной работе изучаются методы построения линейных классификаторов, основанные на критериях минимума суммарной вероятности ошибочной классификации, минимума среднеквадратической ошибки решения, максимума критерия Фишера. Также изучаются итерационные методы настройки классификаторов: алгоритм персептрона и алгоритмы, основанные на методе стохастической аппроксимации.

3.1 Теоретические основы лабораторной работы

3.1.1 Постановка задачи построения линейного классификатора

Классификатор называется *линейным*, если его дискриминантная функция является линейной, то есть допускает представление в виде

$$d(\bar{x}) = \bar{W}^T \bar{x} + w_N, \quad (3.1)$$

где $\bar{x} = (x_0, \dots, x_{N-1})^T$ – вектор признаков, который определяет образ объекта, подлежащего классификации, $\bar{W} = (w_0, \dots, w_{N-1})^T$ – вектор весовых коэффициентов классификатора, w_N – пороговое значение. При этом процесс принятия решения о номере класса текущего объекта производится в соответствии со следующим правилом:

$$d(\bar{x}) = \sum_{i=0}^{N-1} w_i x_i + w_N \begin{matrix} > \\ < \end{matrix} 0 \Rightarrow \bar{X} \in \begin{cases} \Omega_0 \\ \Omega_1 \end{cases}. \quad (3.2)$$

Задача *синтеза линейного классификатора* заключается в поиске вектора коэффициентов \bar{W} и порогового значения w_N оптимальным в каком-либо смысле образом.

Известно [3,5,6,8,9,11,12], что байесовский классификатор в случае нормальных распределений с равными корреляционными матрицами является линейным. Однако часто равенство корреляционных матриц является неприемлемым предположением и, более того, возникает необходимость построения линейного классификатора вообще отказываясь от нормальности распределений.

3.1.2 Линейный классификатор, минимизирующий суммарную вероятность ошибочной классификации

Предположим, что случайная величина $d(\bar{X})$ (как функция случайного вектора \bar{X}) распределена по нормальному или близкому к нему закону распределения в каждом из классов Ω_0 и Ω_1 . В этом случае для вычисления вероятностей ошибок достаточно знать лишь математические ожидания и дисперсии случайной величины $d(\bar{X})$ в каждом из классов.

Действительно, параметры одномерных нормальных распределений случайной величины $d(\bar{X})$ имеют вид

$$\begin{aligned} m_l &= M(d(\bar{X})/\Omega_l) = \bar{W}^T \bar{M}_l + w_N, \\ \sigma_l^2 &= D(d(\bar{X})/\Omega_l) = \bar{W}^T B_l \bar{W}, \quad l = \bar{0}, 1, \end{aligned} \tag{3.3}$$

где

$$\begin{aligned} \bar{M}_l &= M(\bar{X}/\Omega_l), \\ B_l &= M\left((\bar{X} - \bar{M}_l)(\bar{X} - \bar{M}_l)^T / \Omega_l\right), \quad l = \bar{0}, 1 \end{aligned}$$

вектор математических ожиданий и корреляционная матрица случайного вектора \bar{X} соответственно. В силу же (2) вероятности ошибочной классификации первого и второго рода [8,12,6] записываются следующим образом:

$$p_1 = \int_{-\infty}^0 f_d(u/\Omega_1) du, \quad p_0 = \int_0^{\infty} f_d(u/\Omega_0) du, \quad (3.4)$$

где $f_d(u/\Omega_l) \sim N(m_l, \sigma_l^2)$, $(l=0,1)$ – плотности вероятностей нормального закона распределения с соответствующими параметрами.

Зададим критерий оптимальности линейной дискриминантной функции в виде

$$R = P(\Omega_0)p_0 + P(\Omega_1)p_1 \rightarrow \min_{\bar{W}, w_N}, \quad (3.5)$$

где $P(\Omega_0), P(\Omega_1)$ – априорные вероятности появления объектов классов Ω_0 и Ω_1 соответственно. Критерий (3.5), таким образом, определяет оптимальную линейную дискриминантную функцию, которая дает минимальное значение суммарной вероятности ошибочной классификации.

С учетом (3.4) показатель качества критерия (3.5) может быть переписан в следующем виде:

$$R = P(\Omega_0) \int_{\frac{m_0}{\sigma_0}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du + P(\Omega_1) \int_{-\infty}^{\frac{\sigma_1}{m_1}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du. \quad (3.6)$$

Необходимым условием минимума показателя (3.6), как известно, является равенство нулю его частных производных по \bar{W} и w_N :

$$\left\{ \begin{aligned} 0 = \frac{\partial R}{\partial W} &= -P(\Omega_0) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{m_0^2}{2\sigma_0^2}\right) \cdot \frac{\partial}{\partial W} \left(-\frac{m_0}{\sigma_0}\right) + \\ &+ P(\Omega_1) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{m_1^2}{2\sigma_1^2}\right) \cdot \frac{\partial}{\partial W} \left(-\frac{m_1}{\sigma_1}\right), \\ 0 = \frac{\partial R}{\partial w_N} &= -P(\Omega_0) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{m_0^2}{2\sigma_0^2}\right) \cdot \frac{\partial}{\partial w_N} \left(-\frac{m_0}{\sigma_0}\right) + \\ &+ P(\Omega_1) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{m_1^2}{2\sigma_1^2}\right) \cdot \frac{\partial}{\partial w_N} \left(-\frac{m_1}{\sigma_1}\right). \end{aligned} \right. \quad (3.7)$$

Подставляя в (7) выражения для частных производных

$$\begin{aligned} \frac{\partial}{\partial W} \left(-\frac{m_l}{\sigma_l}\right) &= -\frac{\bar{M}_l}{\sigma_l} + \frac{\bar{W}^T B_l m_l}{\sigma_l^3}, \\ \frac{\partial}{\partial w_N} \left(-\frac{m_l}{\sigma_l}\right) &= -\frac{1}{\sigma_l}, \quad l = \overline{0,1}, \end{aligned}$$

окончательно получим следующую систему уравнений:

$$\left\{ \begin{aligned} P(\Omega_1) \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{m_1^2}{2\sigma_1^2}\right) \left(\bar{M}_1 - \frac{m_1}{\sigma_1^2} \bar{W}^T B_1\right) &= \\ = P(\Omega_0) \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{m_0^2}{2\sigma_0^2}\right) \left(\bar{M}_0 - \frac{m_0}{\sigma_0^2} \bar{W}^T B_0\right), \\ P(\Omega_1) \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{m_1^2}{2\sigma_1^2}\right) &= P(\Omega_0) \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{m_0^2}{2\sigma_0^2}\right). \end{aligned} \right. \quad (3.8)$$

При выполнении второго равенства в (8) первое уравнение системы можно переписать в виде

$$\bar{M}_1 - \bar{M}_0 = \left[\frac{m_1}{\sigma_1^2} B_1 - \frac{m_0}{\sigma_0^2} B_0 \right] \bar{W}. \quad (3.9)$$

Решение уравнения (9) может быть получено с помощью следующей итеративной процедуры (процедура Петерсона-Матсона):

$$\bar{W} = [sB_0 + (1-s)B_1]^{-1} (\bar{M}_1 - \bar{M}_0), \quad (3.10)$$

где

$$s = -m_0 \sigma_0^{-2} (m_1 \sigma_1^{-2} - m_0 \sigma_0^{-2})^{-1}, \quad 0 \leq s \leq 1.$$

Из выражения для s следует справедливость равенства $s\sigma_0^2 m_1 + (1-s)\sigma_1^2 m_0 = 0$, откуда, учитывая соотношения (3.3), получаем выражение для w_N :

$$w_N = -\frac{s\sigma_0^2 \bar{M}_1^T + (1-s)\sigma_1^2 \bar{M}_0^T}{s\sigma_0^2 + (1-s)\sigma_1^2} \bar{W}. \quad (3.11)$$

С использованием выражений (3.10) и (3.11) для \bar{W} и w_n можно построить график зависимости суммарной вероятности ошибочной классификации R от единственного параметра s ($0 \leq s \leq 1$) и в качестве окончательного ответа принять то значение параметра, при котором $R(s)$ имеет минимум. Пример подобного решения приведен на рисунке 3.1.

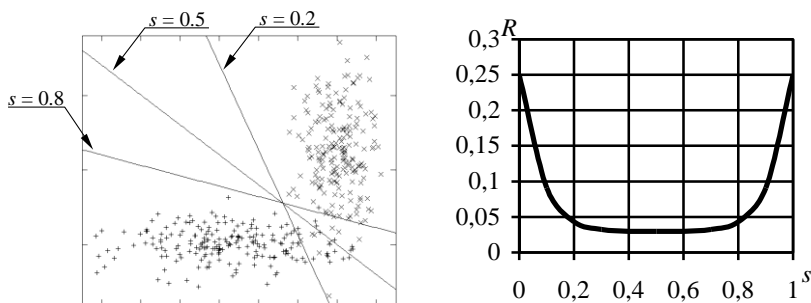


Рисунок 3.1 – Построение линейного классификатора, минимизирующего суммарную вероятность ошибочной классификации

3.1.3 Обобщенная формула построения линейных классификаторов для различных критериев.

Классификатор Фишера

Изложенный выше подход построения линейного классификатора можно обобщить следующим образом. Рассмотрим некоторый критерий $f(m_0, m_1, \sigma_0^2, \sigma_1^2)$, экстремум которого соответствует оптимальному набору параметров линейной дискриминантной функции. Тогда справедливы следующие соотношения:

$$\begin{cases} 0 = \frac{\partial f}{\partial W} = \frac{\partial f}{\partial \sigma_1^2} \cdot \frac{\partial \sigma_1^2}{\partial W} + \frac{\partial f}{\partial \sigma_0^2} \cdot \frac{\partial \sigma_0^2}{\partial W} + \frac{\partial f}{\partial m_1} \cdot \frac{\partial m_1}{\partial W} + \frac{\partial f}{\partial m_0} \cdot \frac{\partial m_0}{\partial W}, \\ 0 = \frac{\partial f}{\partial w_N} = \frac{\partial f}{\partial \sigma_1^2} \cdot \frac{\partial \sigma_1^2}{\partial w_N} + \frac{\partial f}{\partial \sigma_0^2} \cdot \frac{\partial \sigma_0^2}{\partial w_N} + \frac{\partial f}{\partial m_1} \cdot \frac{\partial m_1}{\partial w_N} + \frac{\partial f}{\partial m_0} \cdot \frac{\partial m_0}{\partial w_N}. \end{cases} \quad (3.12)$$

Из соотношений (3) следует, что

$$\begin{aligned} \frac{\partial \sigma_l^2}{\partial W} &= 2B_l \bar{W}, \quad \frac{\partial \sigma_l^2}{\partial w_N} = 0, \\ \frac{\partial m_l}{\partial W} &= \bar{M}_l, \quad \frac{\partial m_l}{\partial w_N} = 1, \quad l = \overline{0,1} \end{aligned}$$

Подставляя эти выражения для частных производных в (3.12), окончательно получаем следующую общую систему уравнений:

$$\begin{cases} 2 \left[\frac{\partial f}{\partial \sigma_0^2} \cdot B_0 + \frac{\partial f}{\partial \sigma_1^2} \cdot B_1 \right] \bar{W} = (\bar{M}_0 - \bar{M}_1) \frac{\partial f}{\partial m_1}, \\ \frac{\partial f}{\partial m_0} = - \frac{\partial f}{\partial m_1}. \end{cases} \quad (3.13)$$

Задавая конкретный вид критерия $f(m_0, m_1, \sigma_0^2, \sigma_1^2)$, из (3.13) можно определить вектор весовых коэффициентов \bar{W} и пороговое

значение w_N линейной дискриминантной функции, оптимальные в смысле этого критерия.

Классификатор Фишера

Выберем в качестве критерия $f(m_0, m_1, \sigma_0^2, \sigma_1^2)$ функцию вида

$$f = \frac{(m_1 - m_0)^2}{\sigma_1^2 + \sigma_0^2}. \quad (3.14)$$

Критерий (3.14) называется *критерием Фишера* и представляет собой меру отличия значений линейной дискриминантной функции в классах Ω_1 и Ω_0 . Для наилучшего разделения классов необходимо определить \bar{W} и w_N , которые доставляли бы этому критерию максимум. Получаемый при этом линейный классификатор называется классификатором Фишера.

Подставляя выражение (14) в общую систему уравнений (13) и игнорируя масштабный множитель линейной функции, получим следующее выражение для вектора весовых коэффициентов дискриминантной функции:

$$\bar{W} = \left(\frac{1}{2}(B_1 + B_2) \right)^{-1} (\bar{M}_1 - \bar{M}_0). \quad (3.15)$$

Выражение (3.10) совпадает с (3.15) при значении параметра $s=0.5$. Делая подстановку этого значения s в (3.11), получим выражение для порогового значения дискриминантной функции w_N :

$$w_N = -\frac{1}{\sigma_1^2 + \sigma_0^2} (\bar{M}_1 - \bar{M}_0)^T \left(\frac{1}{2}(B_1 + B_0) \right)^{-1} (\sigma_1^2 \bar{M}_0 + \sigma_0^2 \bar{M}_1). \quad (3.16)$$

Примечание

Для ситуации равных корреляционных матриц $B \equiv B_1 = B_0$ выражения (15) и (16) преобразуются к следующим:

$$\bar{W} = B^{-1}(\bar{M}_1 - \bar{M}_0), \quad (3.15')$$

$$w_N = -\frac{1}{2}(\bar{M}_1 - \bar{M}_0)^T B^{-1}(\bar{M}_0 + \bar{M}_1). \quad (3.16')$$

Из (3.15') и (3.16') следует, что классификатор Фишера совпадает с байесовским классификатором для нормального закона распределения с равными корреляционными матрицами и равными априорными вероятностями классов:

$$d(\bar{x}) = (\bar{M}_1 - \bar{M}_0)^T B^{-1} \bar{x} - \frac{1}{2}(\bar{M}_1 - \bar{M}_0)^T B^{-1}(\bar{M}_0 + \bar{M}_1).$$

Пример классификатора Фишера приведен на рисунке 3.2.

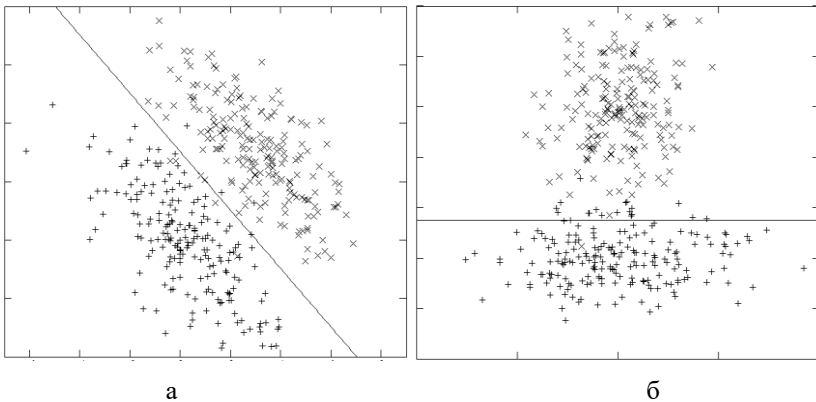


Рисунок 3.2 – Классификатор Фишера: а – равные корреляционные матрицы; б – неравные корреляционные матрицы

3.1.4 Линейный классификатор, минимизирующий СКО решения

При нахождении линейной разделяющей функции, минимизирующей суммарную вероятность ошибочной классификации, предполагалось, что в классах Ω_1 и Ω_0 случайная величина $d(\bar{X})$

имеет нормальный закон распределения. Метод, основанный на минимизации *среднеквадратического отклонения* (СКО) решения, позволяет получить аналогичные результаты без этого предположения.

Пусть при построении классификатора нам для наблюдения доступен набор из K значений векторов–реализаций $\bar{x}^1, \dots, \bar{x}^K$ случайного вектора признаков \bar{X} , относительно каждого из которых известно, какому из классов Ω_1 и Ω_0 он принадлежит. В этом случае говорят, что задана *обучающая выборка*.

Примем далее следующие обозначения:

- Введем в рассмотрение новый вектор $\bar{z} = (z_0, \dots, z_{N-1}, 1)^T$, формируемый следующим образом. Для объектов класса Ω_1 вектор $\bar{z} = (x_0, \dots, x_{N-1}, 1)^T$, а для объектов класса Ω_0 $\bar{z} = (-x_0, \dots, -x_{N-1}, -1)^T$.
- Дополним вектор весовых коэффициентов значением пороговой величины, то есть перейдем к пополненному вектору весовых коэффициентов:

$$\bar{W} = (w_0, \dots, w_{N-1}, w_N)^T.$$

Проведение таких преобразований позволяет записать линейную разделяющую функцию в виде

$$d(\bar{z}) = \bar{W}^T \bar{z}.$$

Построение классификатора при этом сводится к определению пополненного вектора коэффициентов \bar{W} так, чтобы для любых известных векторов \bar{z}^k ($k = \overline{1, K}$) было справедливо неравенство

$$d(\bar{z}) = \bar{W}^T \bar{z} \geq 0. \quad (3.17)$$

Обозначим $\gamma(\bar{z})$ наилучшую разделяющую функцию, то есть такую функцию, значения которой можно рассматривать в качестве требуемого выхода идеальной дискриминантной функции. Как правило, $\gamma(\bar{z})$ не известна, но ее можно предположительно определить на основе обучающей выборки из условия (3.17). Например, можно взять $\gamma(\bar{z}) \equiv 1$, что соответствует случаю, когда на основе обучающей выборки определяется переменная правильной классификации:

$$r(\bar{x}) = \begin{cases} -1, & \bar{x} \in \Omega_0 \\ 1, & \bar{x} \in \Omega_1 \end{cases}. \quad (3.18)$$

СКО между требуемым и действительным значением разделяющей функции определяется выражением

$$\varepsilon^2 = M \left(\gamma(\bar{Z}) - \bar{W}^T \bar{Z} \right)^2, \quad (3.19)$$

(здесь математическое ожидание соответствует распределению случайного вектора \bar{Z}). Если вместо математического ожидания используется среднее по обучающей выборке, то имеем

$$\varepsilon^2 = \frac{1}{K} \sum_{k=1}^K \left(\bar{W}^T \bar{z}^k - \gamma(\bar{z}^k) \right)^2.$$

Используя матричную форму записи последнего выражения, получаем

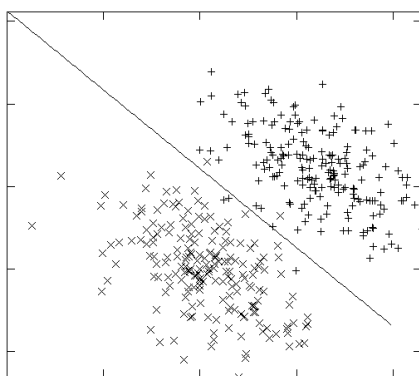
$$\varepsilon^2 = \frac{1}{N} \left(\bar{W}^T U - \bar{\Gamma} \right) \cdot \left(U^T \bar{W} - \bar{\Gamma} \right), \quad (3.20)$$

где $U = (\bar{z}^1, \bar{z}^2, \dots, \bar{z}^K)$ – матрица выборочных данных, а $\bar{\Gamma} = (\gamma(\bar{z}_1), \dots, \gamma(\bar{z}_N))^T$ – вектор требуемых значений выхода. Диф-

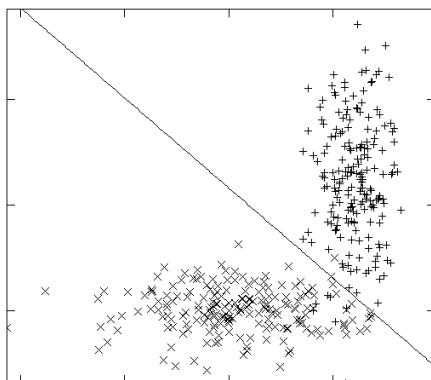
ференцируя выражение (3.20) по дополненному вектору коэффициентов и приравнивая частные производные нулю, получим следующее выражение для \bar{W} :

$$\bar{W} = (UU^T)^{-1} U\bar{G}.$$

Пример классификатора, минимизирующего СКО решения, приведен на рисунке 3.3.



а



б

Рисунок 3.3 – Линейный классификатор, минимизирующий СКО решения:
а – равные корреляционные матрицы;
б – неравные корреляционные матрицы

3.1.5 Последовательная корректировка линейного классификатора: алгоритм перцептрона

Рассмотрим алгоритм, использование которого для расчета вектора параметров линейного классификатора не требует запоминания одновременно всех векторов признаков. Вместо этого в памяти ЭВМ следует хранить только текущие оценки параметров, которые обновляются всякий раз при поступлении очередного вектора наблюдений. Система распознавания такого типа носит название *перцептрон*, а процесс ее итерационной настройки – *обучение*. Основное преимущество подобных алгоритмов в том, что они позволяют использовать бесконечное число наблюдений, располагая конечным объемом памяти.

Итак, пусть решается задача распознавания объектов двух классов, заданных своими пополненными векторами признаками:

$$\bar{x} = (x_0, x_1, \dots, x_{N-1}, 1)^T.$$

Предположим, что в качестве дискриминантной функции выбрана *линейная* и классификатор работает в соответствии со следующим правилом:

$$d(\bar{x}) = \bar{W}^T \bar{x} = \sum_{i=0}^N w_i x_i \begin{cases} < 0 \\ > 0 \end{cases} \Rightarrow \bar{x} \in \begin{cases} \Omega_0 \\ \Omega_1 \end{cases},$$

где $\bar{W} = (w_0, w_1, \dots, w_{N-1}, w_N)^T$ – пополненный вектор весовых коэффициентов.

Пусть $\{\bar{x}^k\}_{k=1}^K$ – обучающая выборка объемом K . Обозначим $\bar{x}(k)$ – элемент выборки, используемый на k -м шаге алгоритма настройки, $\bar{W}(k)$ – оценка искомого вектора \bar{W} на k -м шаге алгоритма.

Классический алгоритм обучения перцептрона, предложенный Ф. Розенблатом в [7], выглядит следующим образом:

$$\bar{W}(k+1) = \begin{cases} \bar{W}(k), & \bar{x}(k) \in \Omega_1 \text{ и } \bar{W}^T(k) \bar{x}(k) > 0 \text{ или} \\ & \bar{x}(k) \in \Omega_0 \text{ и } \bar{W}^T(k) \bar{x}(k) < 0, \\ \bar{W}(k) + c\bar{x}(k), & \bar{x}(k) \in \Omega_1 \text{ и } \bar{W}^T(k) \bar{x}(k) < 0, \\ \bar{W}(k) - c\bar{x}(k), & \bar{x}(k) \in \Omega_0 \text{ и } \bar{W}^T(k) \bar{x}(k) > 0. \end{cases}$$

Выбор параметра c в алгоритме обучения перцептрона производится в соответствии с одним из нижеследующих правил.

Правило 1. Правило фиксированного приращения

Выбирается произвольное постоянное значение $c > 0$.

Правило 2. Правило полной коррекции

Значение параметра c выбирается таким, чтобы текущий вектор признаков был проклассифицирован верно. А именно:

$$c > \frac{|\bar{W}^T(k) \bar{x}(k)|}{\bar{x}^T(k) \bar{x}(k)}.$$

В этом случае параметр c является переменным: $c = c(k)$.

Правило 3. Градиентное правило коррекции

Данное правило используется, если качество линейной дискриминантной функции определяется некоторым функционалом $J(\bar{W})$, минимизацию или максимизацию которого необходимо произвести в процессе настройки классификатора. Экстремум функционала может быть найден, если двигаться вдоль градиента функции качества, то есть использовать градиентное правило коррекции коэффициентов:

$$\bar{W}(k+1) = \bar{W}(k) - \rho \left. \frac{\partial J(\bar{W})}{\partial \bar{W}} \right|_{\bar{W} = \bar{W}(k)},$$

где $\rho > 0$ – параметр градиентного алгоритма. Например, при минимизации СКО, когда

$$J(\bar{W}) = \frac{1}{2} (\gamma(\bar{x}) - \bar{W}^T \bar{x})^2,$$

алгоритм обучения выглядит следующим образом:

$$\bar{W}(k+1) = \bar{W}(k) + \rho \bar{x}(k) (\gamma(\bar{x}(k)) - \bar{W}^T(k) \bar{x}(k)).$$

В этом случае параметр c , очевидно, равен:

$$c = -\rho (\gamma(\bar{x}(k)) - \bar{W}^T(k) \bar{x}(k))$$

и также, как и в предшествующем случае, зависит от k : $c = c(k)$.

Примечание

Алгоритм обучения перцептрона *сходится* за конечное число итераций (то есть строится классификатор, безошибочно классифицирующий все элементы выборочных данных), если классы являются *линейно разделимыми* (рисунок 3.4). В противном случае алгоритм обучения *защелкивается*.

3.1.6 Последовательная корректировка линейного классификатора: стохастическая аппроксимация и процедура Роббинса-Монро

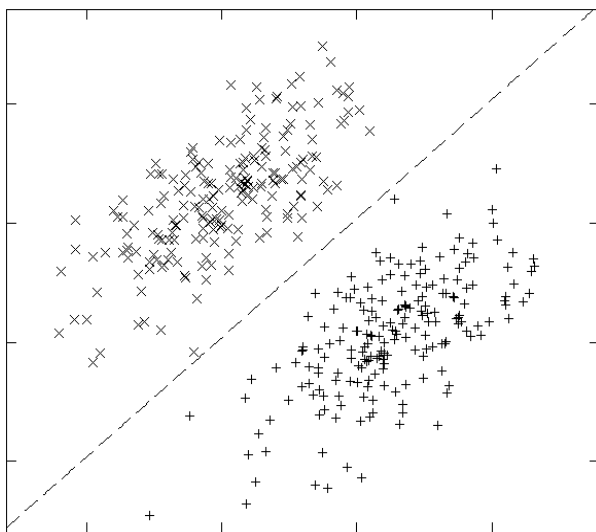
Алгоритм обучения перцептрона не сходится, если классы, заданные обучающими выборками, не являются линейно разделимыми. Этот факт выдвигает задачу построения алгоритма оценивания вектора коэффициентов линейной разделяющей функции

\bar{W} , сходимость которого была бы гарантирована всегда. Такой алгоритм может быть построен с помощью метода *стохастической аппроксимации*.

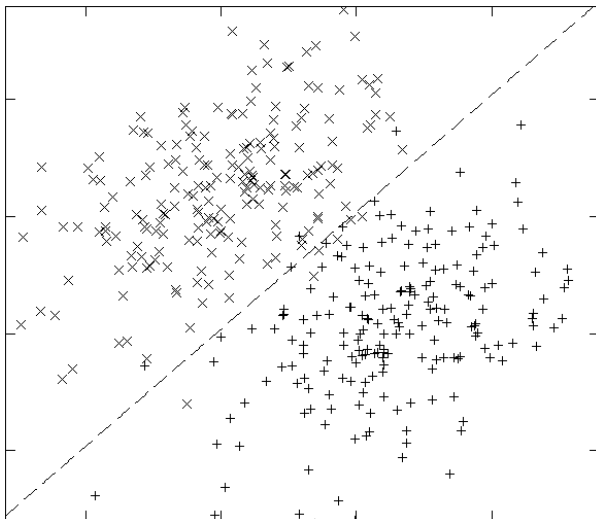
Идея метода состоит в том, что критерий $J(\bar{W})$ рассматривают как функцию регрессии вида $J(\bar{W})=MF(\bar{W},\bar{X})$, где $F(\bar{W},\bar{X})$ – значение функционала качества, наблюдаемого в точке \bar{X} . Метод стохастической аппроксимации позволяет определять по результатам наблюдений корень следующего уравнения:

$$M \frac{\partial F(\bar{W},\bar{X})}{\partial \bar{W}} \equiv Mh(\bar{W},\bar{X})=0,$$

которое называется *уравнением регрессии*.



а



б

Рисунок 3.4 – Иллюстрация линейной разделимости классов:
 а – линейно разделимые классы; б – линейно неразделимые классы

Процедура Роббинса-Монро – это итеративная процедура поиска корня уравнения регрессии. Обозначим $\bar{W}(1)$ произвольную начальную оценку корня \tilde{W} и $\bar{W}(k)$ – оценку этого корня, полученного на k -м шаге итерации. Тогда решение уравнения регрессии может быть получено в результате следующего итерационного процесса:

$$\bar{W}(k+1) = \bar{W}(k) - \alpha_k h(\bar{W}(k), \bar{X}(k)), \quad (3.21)$$

где α_k – элемент последовательности положительных чисел, удовлетворяющий следующим условиям:

$$\lim_{k \rightarrow \infty} \alpha_k = 0, \quad \sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty. \quad (3.22)$$

Утверждение [12]: Если последовательность $\{\alpha_k\}_{k=1}^{\infty}$ удовлетворяет условиям (3.22) и выполнены некоторые дополнительные условия, то оценка (3.21) сходится к корню \tilde{W} в среднеквадратическом и с вероятностью равной единице, то есть

$$\lim_{k \rightarrow \infty} M\left(W(k) - \tilde{W}\right)^2 = 0, \quad \lim_{k \rightarrow \infty} P\left(W(k) = \tilde{W}\right) = 1.$$

Примером последовательности, удовлетворяющей условиям (3.22), является последовательность $\{\alpha_k\}_{k=1}^{\infty}$ вида

$$\alpha_k = \frac{1}{k^\beta}, \quad \frac{1}{2} < \beta \leq 1.$$

3.1.7 Общая схема построения линейных классификаторов, основанная на методе стохастической аппроксимации

Зададимся критерием $J(\bar{W})$ вида

$$J(\bar{W}) = M F(\bar{W}^T \bar{X} - \gamma(\bar{X})),$$

где $F(\dots)$ – некоторая выпуклая функция (например, модуль), $\gamma(\bar{X})$ – требуемый выход разделяющей функции. Дифференцируя по \bar{W} , имеем

$$\frac{\partial J(\bar{W})}{\partial \bar{W}} = M \frac{\partial F(\bar{W}^T \bar{X} - \gamma(\bar{X}))}{\partial \bar{W}} = 0.$$

Полученное уравнение является уравнением регрессии. Воспользовавшись теперь процедурой Роббинса-Монро, можно получить последовательность оценок вектора коэффициентов линейного классификатора, положив

$$h(\bar{W}(k), \bar{X}(k)) = \left. \frac{\partial F(\bar{W}^T \bar{X} - \gamma(\bar{X}))}{\partial \bar{W}} \right|_{\bar{W}=\bar{W}(k), \bar{X}=\bar{X}(k)}$$

и записав алгоритм в виде

$$\bar{W}(k+1) = \bar{W}(k) - \alpha_k \left. \frac{\partial F(\bar{W}^T \bar{X} - \gamma(\bar{X}))}{\partial \bar{W}} \right|_{\bar{W}=\bar{W}(k), \bar{X}=\bar{X}(k)},$$

где начальный вектор $\bar{W}(1)$ выбирается произвольно, а последовательность $\{\alpha_k\}_{k=1}^{\infty}$ удовлетворяет условиям (3.22). Рассмотрим два из возможных алгоритмов, основанных на методе стохастической аппроксимации.

Алгоритм корректирующих приращений (АКП-алгоритм)

В АКП-алгоритме критерий качества классификатора задается в виде

$$J(\bar{W}) = M \left| r(\bar{X}) - \bar{W}^T \bar{X} \right|,$$

где

$$r(\bar{X}) = \begin{cases} 1, & \bar{X} \in \Omega_1 \\ -1, & \bar{X} \in \Omega_0 \end{cases} -$$

случайная переменная правильной классификации. Поскольку производная критерия

$$\frac{\partial J(\bar{W})}{\partial \bar{W}} = -M \bar{X} \operatorname{sgn}(r(\bar{X}) - \bar{W}^T \bar{X}),$$

то в соответствии с общей схемой получаем следующий алгоритм настройки коэффициентов линейной дискриминантной функции:

$$\bar{W}(k+1) = \bar{W}(k) + \alpha_k \bar{x}(k) \operatorname{sgn}\left(r(\bar{x}(k)) - \bar{W}^T(k) \bar{x}(k)\right).$$

Алгоритм можно переписать также в более наглядном виде:

$$\bar{W}(k+1) = \begin{cases} \bar{W}(k) + \alpha_k \bar{x}(k), & r(\bar{x}(k)) \geq \bar{W}^T(k) \bar{x}(k), \\ \bar{W}(k) - \alpha_k \bar{x}(k), & r(\bar{x}(k)) < \bar{W}^T(k) \bar{x}(k). \end{cases}$$

Очевидно, полученный алгоритм корректирует текущую оценку вектора коэффициентов на каждом шаге.

В случае линейной разделимости классов АКП-алгоритм сходится к точному решению, то есть получающийся в результате классификатор все векторы признаков обучающей выборки классифицирует верно. В случае, когда классы линейно разделимыми не являются, в пределе получается решение, оптимальное в смысле минимизации абсолютной величины его расхождения с переменной правильной классификации.

Алгоритм наименьшей СКО (НСКО-алгоритм)

В НСКО-алгоритме критерий качества классификатора имеет вид

$$J(\bar{W}) = 0.5 \cdot M(r(\bar{X}) - \bar{W}^T \bar{X})^2.$$

Поскольку производная критерия

$$\frac{\partial J(\bar{W})}{\partial \bar{W}} = -M \bar{X} (r(\bar{X}) - \bar{W} \bar{X}),$$

то в этом случае имеется следующий алгоритм настройки коэффициентов линейного классификатора:

$$\bar{W}(k+1) = \bar{W}(k) + \alpha_k \bar{x}(k) \left(r(\bar{x}(k)) - \bar{W}^T(k) \bar{x}(k) \right).$$

Аналогично АКП-алгоритму, в НСКО-алгоритме коррекции также производятся на каждом шаге. Отличие заключается только в величине этих коррекций. Для линейно неразделимых классов алгоритм сходится в смысле минимизации величины СКО между решением и переменной правильной классификации.

Проблемы алгоритмов и способы их разрешения

Одной из основных проблем алгоритмов построения линейных классификаторов на основе метода стохастической аппроксимации является очень медленная скорость сходимости. Основные приемы, используемые для ускорения сходимости, следующие:

- выбор медленно убывающей последовательности $\{\alpha_k\}_{k=1}^{\infty}$,
- изменение значения α_k , если только величина

$$\left(r(\bar{x}(k)) - \bar{W}^T(k)\bar{x}(k) \right)$$

на соседних итерациях изменила знак,

- использование «перцептронного приема», при котором коррекция вектора коэффициентов производится только в случае неверной классификации поступившего вектора признаков.

3.2 Порядок выполнения лабораторной работы

3.2.1 Исходные данные

- Два файла данных, полученных в процессе выполнения первой лабораторной работы (см. раздел 1 настоящего издания) и содержащих наборы двумерных нормально распределенных векторов признаков для ситуации равных корреляционных матриц; параметры этих законов распределения; параметры

байесовского классификатора для ситуации равных корреляционных матриц (см. раздел 2 настоящего издания).

- Два файла данных, полученных в процессе выполнения первой лабораторной работы (см. раздел 1 настоящего издания) и содержащих наборы двумерных нормально распределенных векторов признаков для ситуации неравных корреляционных матриц; параметры этих законов распределения; параметры байесовского классификатора для ситуации неравных корреляционных матриц (см. раздел 2 настоящего издания).

3.2.2 Общий план выполнения работы

Построить линейный классификатор, максимизирующий критерий Фишера, для классов Ω_0 и Ω_1 двумерных нормально распределенных векторов признаков для случаев равных и неравных корреляционных матриц. Сравнить качество полученного классификатора с байесовским классификатором.

1. Построить линейный классификатор, минимизирующий среднеквадратичную ошибку, для классов Ω_0 и Ω_1 двумерных нормально распределенных векторов признаков для случаев равных и неравных корреляционных матриц. Сравнить качество полученного классификатора с классификатором Байеса и классификатором Фишера.
2. Построить линейный классификатор, основанный на процедуре Роббинса-Монро, для классов Ω_0 и Ω_1 двумерных нормально распределенных векторов признаков для случаев равных и неравных корреляционных матриц. Исследовать зависимость скорости сходимости итерационного процесса и качества классификации от выбора начальных условий и выбора последовательности корректирующих коэффициентов. Сравнить качество полученного классификатора с байесовским классификатором.

3.2.3 Содержание отчета

Отчет по работе должен содержать:

1. Аналитические выражения для классификаторов, полученных в результате выполнения пп.1–2 плана, и графическое изображение соответствующих им решающих границ вместе с элементами обучающих выборок.
2. Параметры классификатора, полученного в результате выполнения п.3 плана, и его графическое изображение.
3. Вероятности ошибочной классификации построенных в пп.1–3 плана классификаторов, найденные экспериментально. Результаты сравнения построенных классификаторов с байесовским классификатором.
4. Графическая иллюстрация работы итерационного процесса построения классификатора с помощью процедуры Роббинса-Монро; результаты исследования скорости сходимости этого процесса и качества классификации в зависимости от начальных условий и последовательности корректирующих коэффициентов.

4 ЛИНЕЙНЫЕ КЛАССИФИКАТОРЫ. МЕТОД ОПОРНЫХ ВЕКТОРОВ

4.1 Теоретические основы лабораторной работы

4.1.1 Метод опорных векторов для случая линейно разделимых классов

Рассмотрим *линейный классификатор*, дискриминантная функция которого допускает представление в следующем виде:

$$d(\bar{x}) = \bar{w}^T \bar{x} + w_N \quad (4.1)$$

где $\bar{x} = (x_0, \dots, x_{N-1})^T \in D$ – вектор признаков, который определяет образ объекта ω в пространстве признаков D , подлежащего классификации, $\bar{w} = (w_0, \dots, w_{N-1})^T$ – вектор весовых коэффициентов классификатора, w_N – пороговое значение. Процесс принятия решения о номере класса текущего объекта производится в соответствии со следующим правилом:

$$d(\bar{x}) = \sum_{i=0}^{N-1} w_i x_i + w_N \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \bar{X} \in \begin{cases} D_0 \\ D_1 \end{cases}. \quad (4.2)$$

Здесь D_0 и D_1 – области пространства признаков, соответствующие принятию классификатором решения, о принадлежности объекта-прообраза ω в классе Ω_0 или Ω_1 , соответственно. Определим также *переменную правильной классификации* в виде:

$$r(\bar{x}(\omega)) = \begin{cases} 1, & \omega \in \Omega_1, \\ -1, & \omega \in \Omega_0. \end{cases}$$

Рассмотрим линейно разделимую обучающую выборку $\{\bar{x}_i, r_j\}_{j=0}^{N-1}$ ($r_j \equiv r(\bar{x}_j)$), то есть выборку, для которой найдутся параметры классификатора (1), такие что:

$$\forall j=0, N-1 \quad \overline{w}^T \bar{x}_j + w_N > 0 \wedge r_j = 1 \quad \vee \quad \overline{w}^T \bar{x}_j + w_N < 0 \wedge r_j = -1.$$

Учитывая, что «масштаб» значений дискриминантной функции не влияет на результаты классификации, можно так определить параметры классификатора, чтобы выполнялись следующие ограничения:

$$\forall j=0, N-1 \quad \left| \overline{w}^T \bar{x}_j + w_N \right| \geq 1, \quad (4.3)$$

причем равенство в указанном неравенстве происходит только для ближайших точек к разделяющей гиперплоскости (см. рисунок 4.1).

Идея *метода опорных векторов* (англ.: SVM – Support Vectors Machine) [13, 14] заключается в поиске вектора коэффициентов \overline{w} и порогового значения w_N , удовлетворяющих условию (4.3) и обеспечивающих *максимальную ширину разделяющей полосы* (см. рисунок 4.1):

$$\left| \overline{w}^T \bar{x} + w_N \right| \leq 1. \quad (4.4)$$

Можно показать, что ширина разделяющей полосы в данном случае задается величиной $2 \|\overline{w}\|^{-1}$. Тогда окончательный вид задачи построения линейного классификатора по методу опорных векторов следующий:

$$\begin{cases} \overline{w}^T \bar{w} \rightarrow \min, \\ \left(\overline{w}^T \bar{x}_j + w_N \right) \cdot r_j \geq 1, \quad j=0, N-1. \end{cases} \quad (4.5)$$

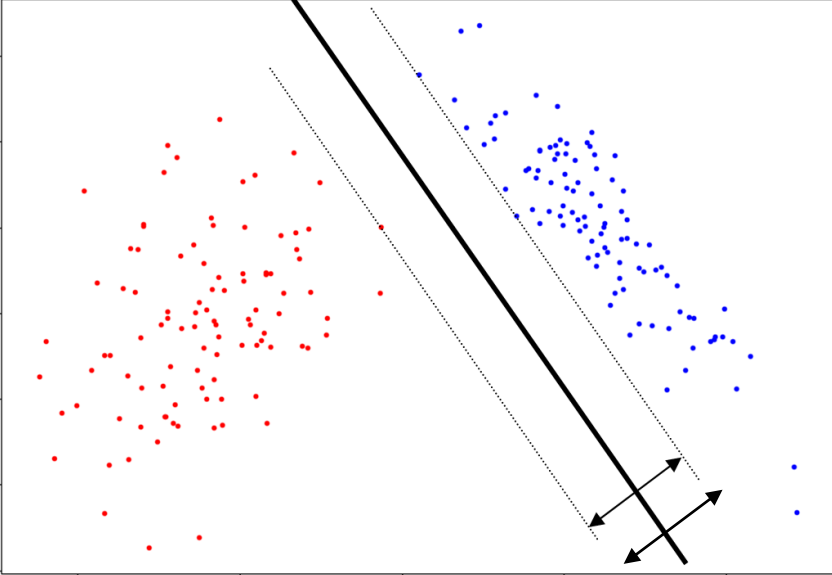


Рисунок 4.1 – Разделяющая гиперплоскость и разделяющая полоса для линейно разделимых классов

По теореме Куна-Такера [15] эта задача квадратичного программирования с линейными ограничениями-неравенствами сводится к задаче поиска седловой точки функции Лагранжа:

$$\left\{ \begin{array}{l} \mathfrak{L}(\bar{w}, w_N, \bar{\lambda}) = \frac{1}{2} \bar{w}^T \bar{w} - \sum_{j=0}^{N-1} \lambda_j \left[r_j \left(\bar{w}^T x_j + w_N \right) - 1 \right] \rightarrow \min_{\bar{w}} \max_{\bar{\lambda}}, \\ \lambda_j \geq 0, \\ \left(\bar{w}^T x_j + w_N = r_j \right) \vee \left(\lambda_j = 0 \right), \quad j = \overline{0, N-1}. \end{array} \right. \quad (4.6)$$

Здесь $\bar{\lambda} = (\lambda_0, \dots, \lambda_{N-1})^T$ – вектор двойственных переменных. Можно показать, что данная задача эквивалентна следующей задаче квадратичного программирования относительно двойственных переменных:

$$\left\{ \begin{array}{l} -\sum_{j=0}^{N-1} \lambda_j + \frac{1}{2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \lambda_j \lambda_i r_j r_i \cdot \bar{x}_j^T \bar{x}_i \rightarrow \min_{\lambda}, \\ \sum_{j=0}^{N-1} \lambda_j r_j = 0; \\ \lambda_j \geq 0, \quad j = \overline{0, N-1}. \end{array} \right. \quad (4.7)$$

Определение. Опорным вектором выборки $\{\bar{x}_j, r_j\}_{j=0}^{N-1}$ называется вектор \bar{x}_j^* , для которого выполняется условие:

$$\left(\bar{w}^T \bar{x}_j + w_N = r_j \right) \wedge (\lambda_j > 0). \quad (4.8)$$

Имея решение задачи (4.7) относительно вектора двойственных переменных, мы находим вектор весов классификатора по формуле:

$$\bar{w} = \sum_{j=0}^{N-1} \lambda_j r_j \cdot x_j. \quad (4.9)$$

Очевидно, что *решение есть линейная комбинация опорных векторов!* Далее имеем:

$$\bar{w}^T \bar{x} = \sum_{j=0}^{N-1} \lambda_j r_j \cdot \bar{x}_j^T \bar{x} \quad (4.10)$$

а пороговое значение может быть определено из условия:

$$\bar{w}^T \bar{x}_j^* + w_N = r_j,$$

где $\lambda_j > 0$, то есть по опорным векторам.

4.1.2. Метод опорных векторов для случая линейно неразделимых классов

Введя дополнительные переменные $s_j \geq 0$, отвечающие за максимально допустимое «нарушение» объектом \bar{x}_j границы разделяющей полосы, задачу (4.5) можно переписать в следующем виде:

$$\begin{cases} \frac{1}{2} \bar{w}^T \bar{w} + C \sum_{j=0}^{N-1} s_j \rightarrow \min_{\bar{w}, s}, \\ \left(\begin{matrix} -T \\ \bar{w} \end{matrix} x_j + w_N \right) \cdot r_j \geq 1 - s_j, \\ s_j \geq 0 \quad j = \overline{0, N-1}. \end{cases} \quad (4.11)$$

Здесь величина C – параметр алгоритма, определяющий компромисс между шириной разделяющей полосы и величиной «нарушений» (ошибок), определяемой суммой $\sum_{j=0}^{N-1} s_j$. Проводя

рассуждения, аналогичные представленным в п.4.1.1, получаем в итоге следующую задачу квадратичного программирования с двойственными переменными:

$$\begin{cases} - \sum_{j=0}^{N-1} \lambda_j + \frac{1}{2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \lambda_j \lambda_i r_j r_i \cdot \bar{x}_j^T \bar{x}_i \rightarrow \min_{\lambda}, \\ \sum_{j=0}^{N-1} \lambda_j r_j = 0; \\ 0 \leq \lambda_j \leq C, \quad j = \overline{0, N-1}. \end{cases} \quad (4.12)$$

Данная задача отличается от задачи (4.7) только допустимым диапазоном двойственных переменных. Определение величины C осуществляется экспериментальным путем.

4.1.3 Использование ядер в методе опорных векторов и переход в пространства высокой размерности

Пусть определено некоторое отображение $\psi: D \rightarrow H$, где H – гильбертово пространство (пространство с определенным в нем скалярным произведением). Используем для классификации образов вместо исходных описаний \bar{x} образы этих векторов признаков в новом пространстве, то есть вектора вида $\psi(\bar{x})$. Тогда задача (4.12) будет иметь следующий вид:

$$\left\{ \begin{array}{l} -\sum_{j=0}^{N-1} \lambda_j + \frac{1}{2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \lambda_j \lambda_i r_j r_i \cdot \psi(\bar{x}_j)^T \psi(\bar{x}_i) \rightarrow \min_{\lambda} \\ \sum_{j=0}^{N-1} \lambda_j r_j = 0; \\ 0 \leq \lambda_j \leq C, \quad j = \overline{0, N-1}. \end{array} \right. \quad (4.13)$$

Здесь $\psi(\bar{x}_j)^T \psi(\bar{x}_i)$ – скалярное произведение в новом гильбертовом пространстве описаний, то есть симметричная и положительно определенная функция. Обозначим $K(\bar{x}_j, \bar{x}_i) \equiv \psi(\bar{x}_j)^T \psi(\bar{x}_i)$, такие функции называют *ядрами*. Тогда задача (4.13) может быть переписана в виде:

$$\left\{ \begin{array}{l} -\sum_{j=0}^{N-1} \lambda_j + \frac{1}{2} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \lambda_j \lambda_i r_j r_i \cdot K(\bar{x}_j, \bar{x}_i) \rightarrow \min_{\lambda} \\ \sum_{j=0}^{N-1} \lambda_j r_j = 0; \\ 0 \leq \lambda_j \leq C, \quad j = \overline{0, N-1}. \end{array} \right. \quad (4.14)$$

Данная задача может быть решена так же, как и задачи (4.12)–(4.13), при этом информация о преобразовании вектора признаков $\psi(\bar{x})$ в явном виде не требуется – достаточно знать только вид ядра $K(\cdot, \cdot)$.

Имея решение задачи (4.14), классификатор можно определить на основании следующих соотношений. Выражение для классификатора (4.10) преобразуется к виду:

$$\bar{w}^T \bar{x} = \sum_{j=0}^{N-1} \lambda_j r_j \cdot K(\bar{x}_j, \bar{x}) \quad (4.15)$$

и очевидным образом вычисляется с учетом известного ядра и найденного вектора двойственных переменных $\bar{\lambda}$. Пороговое значение рассчитывается по тем же правилам, что и в п.4.1.1, что с учетом (4.15) дает выражение:

$$w_N = r_j - \sum_{j=0}^{N-1} \lambda_j r_j \cdot K(\bar{x}_j, \bar{x}_j^*).$$

Примеры некоторых широко используемых ядер представлены в таблице 4.1 ниже. Заметим, что полиномиальное однородное ядро для $d=2$ эквивалентно использованию отображения $\psi(\bar{x})$, формирующего новые признаки вида: $\{x_i x_j\}_{0 \leq i \leq j < n}$ [14].

Таблица 4.1. Примеры ядер

Наименование	Вид ядра
Полиномиальное однородное	$K(\bar{x}, \bar{y}) = (\bar{x}^T \bar{y})^d$
Полиномиальное неоднородное	$K(\bar{x}, \bar{y}) = (\bar{x}^T \bar{y} + 1)^d$

<i>Наименование</i>	<i>Вид ядра</i>
Радиальная функция	$K(\bar{x}, \bar{y}) = \exp(-\gamma \ \bar{x} - \bar{y}\ ^2), \gamma > 0$
Радиальная функция Гаусса	$K(\bar{x}, \bar{y}) = \exp\left(-\frac{\ \bar{x} - \bar{y}\ ^2}{2\sigma^2}\right)$
Сигмоидальная функция	$K(\bar{x}, \bar{y}) = \tanh(\gamma \bar{x}^T \bar{y} + c), \gamma > 0, c < 0$

4.1.4 Достоинства и недостатки метода опорных векторов

Достоинства метода опорных векторов

1. Высокая «надежность» (обобщающая способность) решающего правила.
2. Универсальность, отсутствие формальных ограничений по использованию алгоритма.
3. Возможность неявного выбора пространства классификации путем явного выбора ядер.
4. Возможность неявного перехода (с использованием ядер) в пространство более высокой размерности, где классы оказываются разделимы с большей «вероятностью».

Недостатки метода опорных векторов

1. Сильное влияние шумов, так как классификатор напрямую зависит от выбранных опорных векторов.
2. Необходимость выбора параметра C для случая отсутствия линейной разделимости классов.
3. Отсутствие регулярного метода подбора ядра, связанного с выбором целевого пространства классификации.
4. Вычислительная сложность решения задач квадратичного программирования (4.7), (4.12), (4.14), которая делает затруднительным получения точного решения для больших обучающих выборок.

4.2 Порядок выполнения лабораторной работы

4.2.1 Исходные данные

- Вариант задания.
- Два файла данных, полученных в процессе выполнения первой лабораторной работы (см. раздел 1 настоящего издания) и содержащих наборы двумерных нормально распределенных векторов признаков для ситуации неравных корреляционных матриц; параметры этих законов распределения; параметры байесовского классификатора для ситуации неравных корреляционных матриц (см. раздел 2 настоящего издания).
- Исполняемый в Python файл, необходимый для выполнения пп.1 из плана выполнения лабораторной работы.

4.2.2 Общий план выполнения работы

1. Синтезировать линейно разделимые выборки для двух классов двумерных случайных векторов в количестве $N=100$ штук в каждом классе с использованием предоставленного Python файла.
2. Построить линейный классификатор по методу опорных векторов на выборке с линейно разделимыми классами. Использовать для этого:
 - а) задачу (1.7) и метод решения квадратичных задач:

```
#import qpsolvers  
  
from qpsolvers import solve_qp  
  
alpha = solve_qp(P, q, G, h, A, b)
```
 - б) метод `sklearn.svm.SVC` библиотеки `scikit-learn`.
 - в) сопоставить решения из п.(б) с решением методом `LinearSVC` библиотеки `scikit-learn`.

3. Построить линейный классификатор по методу опорных векторов на выборке с линейно неразделимыми классами. Использовать для этого:
 - а) задачу (4.12) и метод решения квадратичных задач (см. выше). Указать решения для $C=1/10, 1, 10$ и подобранно самостоятельно «лучшим коэффициентом».
 - б) метод `sklearn.svm.SVC` библиотеки `scikit-learn`.
4. Построить линейный классификатор по методу опорных векторов на выборке с линейно неразделимыми классами. Использовать для этого:
 - а) задачу (4.14) и метод решения квадратичных задач (см. выше). Исследовать решение для различных значений параметра $C=1/10, 1, 10$ и вариантов ядер из таблицы 4.1.
 - б) метод `sklearn.svm.SVC` библиотеки `scikit-learn`.

4.2.3 Содержание отчета

Отчет по работе должен содержать:

1. Графическая иллюстрация данных, полученных в результате выполнения п.1 плана.
2. Параметры классификаторов и опорные вектора, полученные в результате выполнения пп.2–4 плана.
3. Вероятности ошибочной классификации построенных в пп.2–4 плана классификаторов, найденные экспериментально. Результаты сравнения построенных классификаторов с байесовским классификатором.
4. Графическая иллюстрация работы итерационного процесса построения классификатора с помощью процедуры Роббинса-Монро; результаты исследования скорости сходимости этого процесса и качества классификации в зависимости от начальных условий и последовательности корректирующих коэффициентов.

5 КЛАССИФИКАТОРЫ, ОСНОВАННЫЕ НА НЕПАРАМЕТРИЧЕСКИХ ОЦЕНКАХ ПЛОТНОСТИ ВЕРОЯТНОСТЕЙ

Цель работы – изучение теоретических основ и экспериментальное исследование классификаторов, основанных на непараметрических методах оценивания плотности вероятностей, для распознавания образов.

В лабораторной работе изучаются методы построения классификаторов, основанных на непараметрических методах оценивания плотности вероятностей, в частности: оценки Парзена, метода ближайшего соседа и K -ближайших соседей.

5.1 Теоретические основы лабораторной работы

Если известны плотности вероятностей (далее – ПВ) случайного вектора признаков для каждого класса и априорные вероятности классов (при простейшей матрице штрафов), то оптимальной стратегией классификации является использование байесовского классификатора в виде:

$$\forall j \neq l \quad P(\Omega_l) f\left(\frac{\bar{x}}{\Omega_l}\right) > P(\Omega_j) f\left(\frac{\bar{x}}{\Omega_j}\right) \Rightarrow \bar{x} \in D_l. \quad (5.1)$$

Если соответствующие плотности и вероятности неизвестны, естественный подход состоит в нахождении их оценок $\hat{f}\left(\frac{\bar{x}}{\Omega_l}\right)$, $\hat{P}(\Omega_l)$ на основе имеющейся обучающей выборки объектов из каждого класса и использовании выражения (5.1) с заменой теоретических величин на эмпирические. Ниже представлены краткие сведения, необходимые для построения и использования классификаторов, использующих два метода непараметрического оценивания ПВ: метод Парзена и метод K ближайших соседей.

5.1.1. Метод Парзена оценивания плотности вероятностей

Пусть $\{\bar{x}_i\}_{i=0}^{N-1}$ – N независимых реализаций случайного вектора \bar{X} размерности n , распределенного по закону, задаваемому неизвестной ПВ $f(\bar{x})$. Метод Парзена получения состоятельных оценок ПВ $f(\bar{x})$ состоит в использовании непрерывных функций вида $\frac{1}{h^n}k\left(\frac{\bar{x}-\bar{x}_i}{h}\right)$, соотнося их с каждым выборочным значением \bar{x}_i , при определенных требованиях к h и функции k . В соответствии с этой идеей приходим к оценке вида

$$\hat{f}_N(\bar{x}) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{1}{h^n} k\left(\frac{\bar{x}-\bar{x}_i}{h}\right), \quad (5.2)$$

где функция $\frac{1}{h^n}k\left(\frac{\bar{y}}{h}\right)$ называется *ядром оценки*. Оценка (5.2) используется в случае «круговых» (изотропных) ядер. Для «некруговых» ядер можно использовать следующую формулу (когда $h_1 \neq h_2 \neq \dots \neq h_N$):

$$\hat{f}_N(\bar{x}) = \frac{1}{N} \cdot \sum_{i=0}^{N-1} \left(\prod_{j=0}^{n-1} h_j \right)^{-1} \cdot k \left[\frac{x_0 - \bar{x}_0^i}{h_0}, \dots, \frac{x_n - \bar{x}_{n-1}^i}{h_{n-1}} \right]. \quad (5.3)$$

Утверждение. Оценка (5.2) является асимптотически несмещенной и состоятельной оценкой функции ПВ $f(\bar{x})$ в точках ее непрерывности, если функция $k\left(\frac{\bar{y}}{h}\right)$ и число h удовлетворяют следующим условиям:

$$\int_{R^n} k\left(\frac{\bar{y}}{h}\right) d\frac{\bar{y}}{h^n} = \int_{R^n} k(\bar{z}) d\bar{z} = 1, \quad (5.4)$$

$$\int \left| k\left(\frac{\bar{y}}{h}\right) \right| d\frac{\bar{y}}{h^n} = \int |k(\bar{z})| d\bar{z} < \infty, \quad (5.5)$$

$$\sup_{\frac{\bar{y}}{h} \in R^n} \left| k\left(\frac{\bar{y}}{h}\right) \right| = \sup_{\bar{z} \in R^n} |k(\bar{z})| < \infty, \quad (5.6)$$

$$\lim_{\frac{\bar{y}}{h} \rightarrow \infty} \left| \frac{\bar{y}}{h} k\left(\frac{\bar{y}}{h}\right) \right| = \lim_{|\bar{z}| \rightarrow \infty} |\bar{z}k(\bar{z})| = 0, \quad (5.7)$$

$$\lim_{n \rightarrow \infty} h^n(N) = 0 \quad (\text{условие асимптотической несмещенности}) \quad (5.8)$$

$$\lim_{n \rightarrow \infty} N \cdot h^n(N) = \infty \quad (\text{условие состоятельности}) \quad (5.9)$$

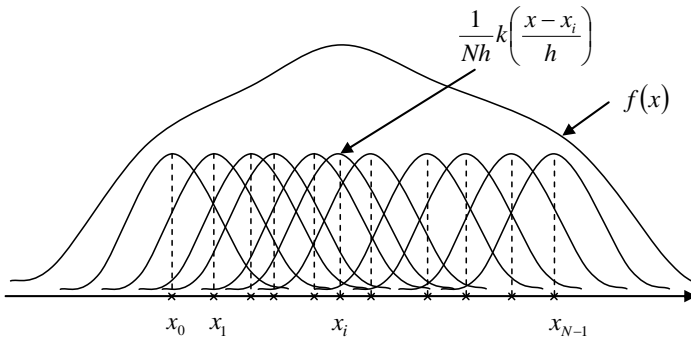


Рисунок 5.1 – Иллюстрация к оценке Парзена ПВ

В случае, если дополнительно к условиям (5.4)–(5.9) выполняется следующее условие

$$\lim_{n \rightarrow \infty} N \cdot h^{2n}(N) = \infty \quad (\text{условия равномерной состоятельности}),$$

то при некоторых дополнительных ограничениях оценка (5.2) является *равномерно состоятельной*:

$$\forall \varepsilon > 0 \quad \lim_{N \rightarrow \infty} \cdot P \left\{ \sup_{-\infty < x < \infty} \left| \hat{f}_N(x) - f(x) \right| > \varepsilon \right\} = 0.$$

Примером многомерного ядра, имеющего вид нормальной ПВ. Это ядро определяется следующим образом:

$$\begin{aligned} \frac{1}{h^n} \cdot k \cdot \left(\frac{\bar{x} - \bar{x}_i}{h} \right) = \\ = (2 \cdot \pi)^{-n/2} \cdot h^{-n} \cdot |B|^{-1/2} \cdot \exp \left[-\frac{1}{2} \cdot h^{-2} \cdot (\bar{x} - \bar{x}_i)' \cdot B^{-1} \cdot (\bar{x} - \bar{x}_i) \right]. \end{aligned} \quad (5.10)$$

Здесь, очевидно:

$$(2 \cdot \pi)^{-n/2} \cdot h^{-n} \cdot |B|^{-1/2} = \frac{1}{(2 \cdot \pi)^{n/2} \cdot \sqrt{|B|} \cdot h^n}.$$

Примечание. Пусть в выражении (5.11) в качестве матрицы B использована выборочная ковариационная матрица \hat{B} , рассчитанная по выборке. Тогда ковариационная матрица оценки Парзена B_p и матрица \hat{B} связаны соотношением:

$$B_p = \frac{1}{N} \cdot \sum_{i=0}^{N-1} \left(\bar{x}_i \cdot \bar{x}_i' + h^2 \cdot \hat{B} \right) \approx (1 + h^2) \cdot \hat{B}.$$

Параметр $h(N)$ удобно использовать в виде:

$$h(N) = N^{-\frac{k}{n}} = \left(N^{-\frac{k}{n}} \right). \quad (5.11)$$

Для выполнения условий (5.8) – (5.10) величина k должна удовлетворять следующим ограничениям: $1 > k > 0$ – условие состоятельности, $\frac{1}{2} > k > 0$ – условие равномерной состоятельности.

Примеры одномерных ядер, используемых в оценке Парзена, приведены ниже.

Прямоугольное ядро

$$\frac{1}{h}k\left(\frac{y}{h}\right) = \begin{cases} \frac{1}{2h}, & \left|\frac{y}{h}\right| \leq 1, \\ 0, & > 1. \end{cases}$$

Треугольное ядро

$$\frac{1}{h}k\left(\frac{y}{h}\right) = \begin{cases} \frac{1}{h}\left(1 - \left|\frac{y}{h}\right|\right), & \left|\frac{y}{h}\right| \leq 1, \\ 0, & > 1. \end{cases}$$

Нормальное ядро

$$\frac{1}{h}k\left(\frac{y}{h}\right) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{y^2}{2h^2}\right),$$

Экспоненциальное ядро

$$\frac{1}{2h} \exp\left(-\frac{y}{h}\right),$$

Рациональное ядро

$$\frac{1}{\pi h} \frac{1}{1 + \left(\frac{y}{h}\right)^2},$$

Ядро "sinc"

$$\frac{1}{\pi h} \left(\frac{\sin \frac{y}{2h}}{\frac{y}{2h}} \right)^2,$$

Достоинства и недостатки оценки Парзена

К достоинствам оценки следует отнести ее универсальность. Недостатками оценки являются ее высокая вычислительная сложность (число слагаемых в выражении (2) соответствует объему обучающей выборки); необходимость хранения всей обучающей выборки; неоднозначность оценки, связанной с произвольностью выбора ядер и величины $h=h(N)$.

5.1.2. Метод K ближайших соседей

Используя выборку $\{\bar{x}_i\}_{i=0}^{N-1}$ объема N , найдем расстояние r от точки \bar{x} до K -го ближайшего к \bar{x} элемента выборки (K -го ближайшего соседа). Для измерения «близости» можно воспользоваться любой подходящей меркой. Тогда в качестве оценки ПВ в точке \bar{x} можно принять:

$$\hat{f}_N(\bar{x}) = \frac{K}{N} \cdot \frac{1}{A(k, N, \bar{x})}, \quad (5.12)$$

где $A(k, N, \bar{x})$ – объем множества всех точек, расстояние которых до \bar{x} меньше, чем r . Величина A является случайной (и зависит фактически от \bar{x}), зависящей от выбранного множества N элементов выборки.

Утверждение. Если параметр $K(N)$, удовлетворяет условиям:

$$\lim_{N \rightarrow \infty} K(N) = \infty,$$

$$\lim_{N \rightarrow \infty} \frac{K(N)}{N} = 0,$$

то оценка (5.12) является асимптотически несмещенной и состоятельной оценкой ПВ $f(\bar{x})$ в точках ее непрерывности.

5.1.2.1 Решающее правило K ближайших соседей

Полученная оценка $\hat{f}_N(\bar{x})$ ПВ может использоваться следующим образом. Когда требуется классифицировать неизвестный объект (вектор признаков) \bar{x} , среди имеющихся N объектов (уже проклассифицированных), из которых N_l объектов из класса

$\Omega_l \left(\sum_{l=0}^{L-1} N_l = N \right)$, находят K ближайших к точке \bar{x} объектов.

Пусть $k_l \left(l = \overline{0, L-1} \right)$ – число объектов из класса Ω_l среди этих K ближайших соседей $\left(\sum_{l=0}^{L-1} K_l = K \right)$. Тогда оценка ПВ принимает вид

(в l -м классе):

$$\hat{f}_{N_i}^i \left(\bar{x} / \Omega_l \right) = \frac{K_l}{N_l} \cdot \frac{1}{V}, \quad l = \overline{0, L-1}. \quad (5.13)$$

Поскольку независимо от номера класса, K_l объектов извлечены из одной и той же области A , то величина объема V оказывается одинаковой для всех классов. Следовательно, байесовский классификатор (5.1), минимизирующий общий риск, будет иметь вид (оценку априорной вероятности берем в виде (5.15):

$$\hat{P}(\Omega_l) = \frac{N_l}{N};$$

$$\forall j \neq l \quad \frac{N_l}{N} \cdot \hat{f}_N \left(\bar{x} / \Omega_l \right) \geq \frac{N_j}{N} \cdot \hat{f}_N \left(\bar{x} / \Omega_j \right) \Rightarrow \bar{x} \in D_l.$$

Подставляя в это выражение значение из (13), получим:

$$\forall j \neq l \quad \frac{N_l}{N} \cdot \frac{K_l}{N_l \cdot V} \geq \frac{N_j}{N} \cdot \frac{K_j}{N_j \cdot V} \Rightarrow \bar{x} \in D_l$$

или

$$\forall j \neq l \quad K_l \geq K_j \Rightarrow \bar{x} \in D_l. \quad (5.14)$$

Окончательно, решающее правило можно записать в виде:

$$l = \arg \max_{j=0, L-1} K_j, \quad \bar{x} \in D_l.$$

5.1.2.2 Решающее правило K ближайших соседей для двух классов

Из соотношения (14) непосредственно следует:

$$K_0 > K_1 \Rightarrow \bar{x} \in \begin{cases} D_0 \\ D_1 \end{cases}$$

Таким образом, решение о принадлежности вектора \bar{x} принимается в тот класс, соседей которого больше. Для устранения возможных проблем с «равенством» числа соседей, число K в случае двух классов следует брать нечетным.

5.1.2.3 Решающее правило ближайшего соседа

В случае $K=1$ решающее правило называют *правилом ближайшего соседа*. В соответствие с этим правилом решение о принадлежности вектора \bar{x} принимается в тот класс, у которого оказался ближайший к \bar{x} сосед.

Достоинства и недостатки решающего правила K соседей

Достоинство: решающее правило, основанное на методе K ближайших соседей, является очень простым и не требует знания (явного построения) ПВ (или ее оценки).

Его *недостаток* заключается в необходимости хранить в памяти машины все объемы (всю обучающую выборку) и сравнивать каждый из них с неизвестным объектом.

5.1.3 Оценка априорных вероятностей

Оценка величин $P(\Omega_l)$ для больших объемов обучающей выборки может быть выполнена следующим образом:

$$\hat{P}(\Omega_l) = \frac{N_l}{N}, \quad N = N_0 + \dots + N_{L-1}, \quad (5.15)$$

Здесь N – общее количество элементов обучающей выборки во всех классах; N_l – количество элементов выборки в l -м классе.

5.2 Порядок выполнения лабораторной работы

5.2.1 Исходные данные

- Два файла данных, полученных в процессе выполнения первой лабораторной работы (см. раздел 1 настоящего издания) и содержащих наборы двумерных нормально распределенных векторов признаков для ситуации равных корреляционных матриц; параметры этих законов распределения; параметры байесовского классификатора для ситуации равных корреляционных матриц (см. раздел 2 настоящего издания).
- Два файла данных, полученных в процессе выполнения первой лабораторной работы (см. раздел 1 настоящего издания) и содержащих наборы двумерных нормально распределенных векторов признаков для ситуации неравных корреляционных матриц; параметры этих законов распределения; параметры байесовского классификатора для ситуации неравных корреляционных матриц (см. раздел 2 настоящего издания).
- Исполняемый в Python файл, необходимый для выполнения п.2 из плана выполнения лабораторной работы.

5.2.2 Общий план выполнения работы

1. Синтезировать дополнительные реализации для каждого из классов (использовать результаты и файлы лабораторной работы № 1), соответствующих существующим исходным данным (см. 5.2.1).

2. Построить классификатор, основанный на непараметрической оценке Парзена, используя сгенерированные на предшествующем шаге данные как обучающие выборки, а данные из первой лабораторной работы – как тестовые. В качестве ядра взять гауссовское (5.10), величину h взять в виде (5.11). Оценить эмпирический риск – оценку суммарной вероятности ошибочной классификации.

3. Построить классификатор, основанный на методе K ближайших соседей (для $K=1,3,5$), используя сгенерированные в п.1 данные как обучающие выборки, а данные из первой лабораторной работы – как тестовые. Оценить эмпирический риск – оценку суммарной вероятности ошибочной классификации. Для построения классификатора использовать `sklearn.neighbors.KNeighborsClassifier` библиотеки библиотеки `scikit-learn`.

4. Сравнить полученные в пп.2–3 классификаторы по качеству их работы между собой и с байесовским классификатором из л.р.№2.

5.2.3 Содержание отчета

Отчет по работе должен содержать:

1. Графическая иллюстрация данных, полученных в результате выполнения п.1 плана и результатов их классификации из пп.2–3.

2. Вероятности ошибочной классификации построенных в пп.2–3 плана классификаторов, найденные экспериментально. Результаты сравнения построенных классификаторов с байесовским классификатором.

6 АВТОМАТИЧЕСКАЯ КЛАССИФИКАЦИЯ

Цель работы – изучение теоретических основ и экспериментальное исследование методов автоматической классификации.

В лабораторной работе изучаются методы автоматической классификации в распознавании образов. Рассматривается постановка задачи автоматической классификации, используемая при автоматической классификации меры сходства, критерии кластеризации. Рассматриваются три известных алгоритма: простой алгоритм выделения кластеров, максиминный и K внутригрупповых средних.

6.1 Теоретические основы лабораторной работы

6.1.1 Постановка задачи автоматической классификации

Пусть классификации подлежат N объектов, каждый из которых характеризуется n -мерным вектором признаков \bar{x} , то есть дано множество векторов $\{\bar{x}_i\}_{i=0}^{N-1}$. Эти вектора рассматриваются как фиксированные. Каждый объект должен быть отнесен к одному из L классов: $\Omega_0, \Omega_1, \dots, \Omega_{L-1}$, где число классов L может быть известно заранее или может быть не известно. Таким образом, основной вопрос задачи автоматической классификации (АК), так же как и в других задачах классификации, это вопрос об определении класса¹.

Один из возможных подходов к определению класса состоит в его понимании как *кластера (таксона)*, то есть компактной в некотором смысле области в признаковом пространстве. В этом

¹ Речь идет не о задании самого класса, а о задании областей признакового пространства, которые «соответствуют» классам.

случае задача АК является задачей *кластер-анализа* и представляет собой задачу идентификации групп схожих образов в анализируемом множестве данных (или задачу выделения кластеров).

Процесс выделения кластеров является искусством весьма «эмпирическим», так как работа конкретного алгоритма зависит не только от характера анализируемых данных, но в значительной степени и от выбранной меры подобия образов и метода, используемого для идентификации кластеров, и даже от последовательности просмотра образов.

6.1.2 Меры сходства

Для того чтобы определить на множестве данных кластеры, необходимо *в первую очередь ввести меру сходства (подобия) образов* (векторов признаков), которая может быть положена в основу правила отнесения образов характеризуемой некоторым центром кластера. В качестве мер сходства широко используются следующие:

– *евклидово расстояние* $d(\bar{x}, \bar{z}) = \|\bar{x} - \bar{z}\|$ (меньше d – больше сходство);

– *расстояние Махаланобиса* $d(\bar{x}, \bar{z}) = (\bar{x} - \bar{z})^T B^{-1} (\bar{x} - \bar{z})$ (меньше d – больше сходство). Используется, когда известны статистические характеристики кластеров (матрицы разброса);

– *косинус угла между векторами* $d(\bar{x}, \bar{z}) = \frac{\bar{x}^T \bar{z}}{\|\bar{x}\| \cdot \|\bar{z}\|}$. Данную

меру удобно использовать, когда кластеры имеют тенденцию располагаться вдоль главных осей, как, например, изображено на рисунке 6.1;

– *мера Такимото* $d(\bar{x}, \bar{z}) = \frac{\bar{x}^T \bar{z}}{\bar{x}^T \bar{x} + \bar{z}^T \bar{z} - \bar{x}^T \bar{z}}$.

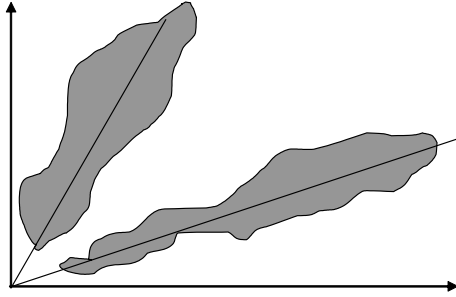


Рисунок 5.1 – Иллюстрация к мерам сходства

В дальнейшем ограничимся евклидовой мерой подобия.

6.1.3 Критерии кластеризации

После выбора меры сходства необходимо определить *критерий кластеризации*. Критерий кластеризации может либо воспроизводить некие эвристические соображения, либо основываться на минимизации (или максимизации) какого-нибудь показателя качества.

При *эвристическом подходе* решающую роль играют интуиция и опыт. Он предусматривает задание набора правил, которые обеспечивают использование выбранной меры сходства для отнесения образов к одному из кластеров.

Подход к кластеризации, предусматривающий *использование показателя качества*, связан с разработкой процедур, которые обеспечат минимизацию или максимизацию выбранного показателя качества. Одним из наиболее популярных показателей является сумма квадратов ошибки:

$$\mathfrak{J} = \sum_{l=0}^{L-1} \sum_{\bar{x} \in S_l} \|\bar{x} - \bar{M}_l\|^2, \quad (4.1)$$

где L – число кластеров, S_l – множество образов (векторов признаков), относящихся к l -му кластеру, а $\bar{M}_l = \frac{1}{N_l} \sum_{\bar{x} \in S_l} \bar{x}$ – вектор вы-

борочных средних значений для множества образов S_l , величина N_l характеризует количество образов, входящих во множество S_l . Как видно, показатель качества (1) определяет общую сумму квадратов отклонений характеристик всех образов, входящих в некоторый кластер, от соответствующих средних значений по этому кластеру. Алгоритм, основанный на этом показателе качества, рассматривается ниже. Естественно, существует масса показателей качества, помимо рассмотренного. Нередко применяются алгоритмы отыскания кластеров, основанные на совместном использовании эвристического подхода и показателя качества. Примером является алгоритм ISODATA [8].

Ниже рассмотрены три наиболее известных алгоритма [8], которые являются примерами как эвристического подхода, так и подхода, использующего показатель качества.

6.1.4 Простой алгоритм выделения кластеров

Пусть задано множество N образов $\{\bar{x}_0, \bar{x}_1, \dots, \bar{x}_{N-1}\}$. Пусть также центр первого кластера \bar{M}_0 совпадает с любым из заданных образов и определена произвольная неотрицательная пороговая величина T ; для удобства можно считать, что $\bar{M}_0 = \bar{x}_0$. После этого вычисляется расстояние $d(\bar{M}_0, \bar{x}_1)$ между существующим центром кластера \bar{M}_0 и образом \bar{x}_1 . Если это расстояние больше значения пороговой величины T ($d(\bar{M}_0, \bar{x}_1) > T$), то учреждается новый кластер с центром $\bar{M}_1 = \bar{x}_1$. В противном случае образ \bar{x}_1 включается в кластер, центром которого является \bar{M}_0 . Пусть условие $d(\bar{M}_0, \bar{x}_1) > T$ выполнено, и \bar{M}_1 – центр нового кластера.

На следующем шаге вычисляются расстояния $d(\bar{M}_0, \bar{x}_2)$ и $d(\bar{M}_1, \bar{x}_2)$ до образа \bar{x}_2 от центров кластеров \bar{M}_0 и \bar{M}_1 . Если

оба расстояния оказываются больше порога T ($d(\bar{M}_0, \bar{x}_2) > T$ и $d(\bar{M}_1, \bar{x}_2) > T$), то учреждается новый кластер с центром $\bar{M}_2 = \bar{x}_2$. В противном случае образ \bar{x}_2 зачисляется в тот кластер l ($l=0,1$), чей центр \bar{M}_l к нему ближе.

Подобным же образом расстояния от каждого нового образа \bar{x}_j ($j=1, N-1$) до каждого известного центра кластера \bar{M}_l ($l=0, L-1$) вычисляются и сравниваются с пороговой величиной. Если все эти расстояния превосходят значение порога T ($\forall l=0, L-1 \ d(\bar{M}_l, \bar{x}_j) > T$), то учреждается новый кластер с центром $\bar{M}_L = \bar{x}_j$ (и число кластеров увеличивается на единицу). В противном случае образ зачисляется в кластер с самым близким к нему центром.

Результаты описанной процедуры определяются выбором первого центра кластера, порядком просмотра образов, значением пороговой величины T и, конечно, геометрическими характеристиками данных.

6.1.5 Алгоритм максиминного расстояния

Алгоритм, основанный на принципе *максиминного* (максимально-минимального) *расстояния*, представляет собой еще одну простую эвристическую процедуру, использующую евклидово расстояние. Этот алгоритм в принципе аналогичен схеме из п.1.4, за исключением того обстоятельства, что в первую очередь он выявляет наиболее удаленные кластеры.

Алгоритм состоит из нескольких шагов.

Шаг 1. Произвольным образом выбирается центр первого кластера \bar{M}_0 . Удобно выбирать в качестве центра кластера \bar{M}_0 тот вектор признаков $\bar{x} \in \{\bar{x}_j\}_{j=0}^{N-1}$, который обладает некоторыми

«экстремальными» свойствами. Например, удобным является тот вектор, удаление которого от среднего всех векторов из выборки максимально:

$$\bar{M}_0 = \arg \max_{\bar{x} \in \{\bar{x}_j\}_{j=0}^{N-1}} d \left(\bar{x}, \frac{1}{N} \sum_{i=0}^{N-1} \bar{x}_i \right).$$

Шаг 2. Выбирается центр второго кластера \bar{M}_1 . В качестве центра используется тот вектор $\bar{x} \in \{\bar{x}_j\}_{j=0}^{N-1}$, который наиболее удален от первого центра кластера:

$$\bar{M}_1 = \arg \max_{\bar{x} \in \{\bar{x}_j\}_{j=0}^{N-1}} d(\bar{M}_0, \bar{x}).$$

Шаг L ($L > 2$). Выбирается центр кластера \bar{M}_{L-1} . Для этого вычисляются все расстояния между оставшимися образами (векторами признаков) $\bar{x} \in \{\bar{x}_j\}_{j=0}^{N-1} \setminus \{\bar{M}_l\}_{l=0}^{L-2}$ и уже существующими центрами кластеров $\bar{M}_0, \bar{M}_1, \dots, \bar{M}_{L-2}$. Для каждого оставшегося образа \bar{x} находится тот центр кластера l , расстояние до которого минимально: $l = \arg \min_{l=0, L-2} d(\bar{M}_l, \bar{x})$ (вектора распределяются по кластерам по критерию близости к их центру). В качестве претендента на новый центр кластера \bar{M}_{L-1} берется тот вектор признаков \bar{x} , у которого это минимальное расстояние (расстояние до центра «своего» кластера) максимально:

$$\tilde{\bar{M}}_{L-1} = \arg \max_{\bar{x} \in \{\bar{x}_j\}_{j=0}^{N-1}} \min_{l=0, L-2} d(\bar{M}_l, \bar{x}).$$

Полученное для выбранного вектора расстояние $d_{\min} = \min_{l=0, L-2} d(\bar{M}_l, \tilde{\bar{M}}_{L-1})$ от него до ближайшего кластера сравнивается с некоторым «типичным» расстоянием $d_{typical}$ между кластерами. Если полученное расстояние больше «типичного», то этот вектор становится новым центром кластера:

$$d_{\min} > d_{typical} \Rightarrow \bar{M}_{L-1} = \tilde{\bar{M}}_{L-1}.$$

В противном случае – процесс выделения кластеров останавливается.

Выбор типичного расстояния может быть осуществлен различными способами. Один из часто используемых способов является задание типичного расстояния равного некоторой части η от среднего расстояния между уже существующими кластерами:

$$d_{typical} = \eta \left(\frac{2}{(L-1)(L-2)} \right) \sum_{l=0}^{L-2} \sum_{j=l+1}^{L-2} d(\bar{M}_l, \bar{M}_j).$$

Величина η выбирается из условия $0 < \eta < 1$. Типичное значение:

$$\eta = \frac{1}{2}.$$

6.1.6 Алгоритм K внутригрупповых средних

Алгоритмы, рассмотренные в пп. 6.1.4–6.1.5, являются, в сущности, эвристическими процедурами. Алгоритм, представленный ниже, минимизирует показатель качества, заданный как сумма квадратов расстояний всех точек, входящих в кластерную область, до центра кластера. Эта процедура, которую часто называют алгоритмом, основанным на вычислении K внутригрупповых средних, состоит из следующих шагов.

Описание алгоритма

Шаг 1. Выбираются K исходных центров кластеров на первой итерации ($r=1$): $\bar{M}_0(r), \bar{M}_1(r), \dots, \bar{M}_{K-1}(r)$. Этот выбор производится произвольно, и обычно в качестве исходных центров используются первые K образов из заданного множества $\{\bar{x}_j\}_{j=0}^{N-1}$ ($N > K$):

$$\bar{M}_0(1) = \bar{x}_0, \bar{M}_1(1) = \bar{x}_1, \dots, \bar{M}_{K-1}(1) = \bar{x}_{K-1}.$$

Номер итерации увеличивается: $r := r + 1$.

Шаг 2. На r -й итерации ($r=2, 3, \dots$) исходное множество образов $\{\bar{x}_j\}_{j=0}^{N-1}$ распределяется по K кластерам по правилу близости.

То есть некоторый образ $\bar{x} \in \{\bar{x}_j\}_{j=0}^{N-1}$ относят в кластер $S_k(r)$ ($k=0, K-1$) с центром $\bar{M}_k(r-1)$, рассчитанным на предыдущем шаге, если этот центр – ближайший:

$$\bar{x} \in S_k(r), \quad k = \arg \min_{j=0, K-1} d(\bar{x}, \bar{M}_j(r-1)).$$

Таким образом, $S_k(r)$ – множество образов, входящих в кластер с номером k на r -й итерации алгоритма. В случае равенства расстояний от некоторого образа до нескольких центров кластеров решение об отнесении этого образа к одному из них принимается произвольным образом.

Шаг 3. На основе результатов шага 2 определяются новые центры кластеров $\bar{M}_0(r), \bar{M}_1(r), \dots, \bar{M}_{K-1}(r)$ на r -й итерации алгоритма. Они определяются из условия, чтобы сумма квадратов расстояний между всеми образами, принадлежащими кла-

стеру $S_k(r)$, и новым центром кластера $\bar{M}_k(r)$ должна быть минимальной. Другими словами, новые центры кластеров выбираются таким образом, чтобы минимизировать частные показатели

$$J_k(r) = \sum_{\bar{x} \in S_k(r)} d(\bar{x}, \bar{M}_k(r))^2, \quad k=0,1,\dots,K-1$$

и, следовательно, интегральный показатель качества кластеризации на r -м шаге:

$$J(r) = \sum_{k=0}^{K-1} J_k(r) = \sum_{k=0}^{K-1} \sum_{\bar{x} \in S_k(r)} d(\bar{x}, \bar{M}_k(r))^2.$$

Для случая евклидова расстояния новый центр кластера $\bar{M}_k(r)$, обеспечивающий минимизацию соответствующего частного показателя $J_k(r)$, является, в сущности, выборочным средним, определенным по множеству образов в кластере $S_k(r)$:

$$\bar{M}_k(r) = \frac{1}{N_k} \sum_{x \in S_k(r)} \bar{x}, \quad k=0,1,\dots,K-1,$$

где $N_k = |S_k(r)|$ – число образов, входящих в кластер $S_k(r)$ на r -м шаге. Как видно, название алгоритма « K внутригрупповых средних» определяется способом, принятым для последовательной коррекции назначения центров кластеров.

Шаг 4. Равенство центров кластеров на соседних шагах $\bar{M}_k(r) = \bar{M}_k(r-1)$ ($k=0, \overline{K-1}$) является условием сходимости алгоритма и при его достижении выполнение алгоритма заканчивается. В противном случае алгоритм повторяется с шага 2 с новым номером итерации $r := r+1$.

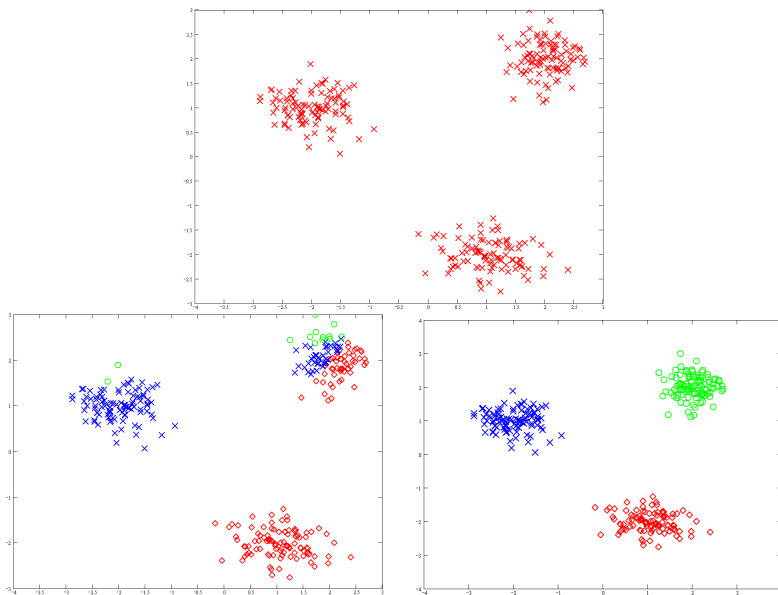


Рисунок 6.1 – Пример кластеризации с использованием алгоритма K внутригрупповых средних (2 шага алгоритма)

Комментарии

Качество работы алгоритма K внутригрупповых средних зависит от числа выбираемых центров кластеров K , от выбора центров кластеров на первой итерации $\bar{M}_0(1), \bar{M}_1(1), \dots, \bar{M}_{K-1}(1)$ и, естественно, от геометрических особенностей данных. От последовательности просмотра данных результаты не зависят.

Хотя для этого алгоритма общее доказательство сходимости не известно, получения приемлемых результатов можно ожидать в тех случаях, когда данные образуют характерные гроздья, отстоящие друг от друга достаточно далеко. В большинстве случаев практическое применение этого алгоритма потребует проведения экспериментов, связанных с выбором различных значений параметра K , и расположения первоначальных центров кластеров.

6.3 Порядок выполнения лабораторной работы

6.3.1 Исходные данные

– Математические ожидания для пяти случайных векторов признаков, задаются по номеру варианта учащегося (см. раздел 7, часть 2).

6.3.2 Общий план выполнения работы

1. Смоделировать и изобразить графически обучающие выборки объема $N=50$ для пяти нормально распределенных двумерных случайных векторов с заданными математическими ожиданиями и самостоятельно подобранными корреляционными матрицами, которые обеспечивают линейную разделимость классов.
2. Объединить пять выборок в одну. Общее количество векторов в объединенной выборке должно быть 250. Полученная объединенная выборка используется для выполнения пунктов 3 и 4 настоящего плана.
3. Разработать программу кластеризации данных с использованием минимаксного алгоритма. В качестве типичного расстояния взять половину среднего расстояния между существующими кластерами. Построить отображение результатов кластеризации для числа кластеров, начиная с двух. Построить график зависимости максимального (из минимальных) и типичного расстояний от числа кластеров.
4. Разработать программу кластеризации данных с использованием алгоритма K внутригрупповых средних для числа кластеров равного 3 и 5. Для ситуации 5 кластеров подобрать начальные условия так, чтобы получить два результата: а) чтобы кластеризация максимально **соответствовала** первоначальному разбиению на классы («правильная» кластеризация); б) чтобы кластеризация максимально **не соответствовала**

ла первоначальному разбиению на классы («неправильная» кластеризация). Для всех случаев построить графики зависимости числа векторов признаков, сменивших номер кластера, от номера итерации.

6.3.3 Содержание отчета

Отчет по работе должен содержать:

- исходные данные генерируемых векторов признаков – средние и ковариационные матрицы;
- для минимаксного алгоритма: полученное число классов, график зависимости максимального и типичного расстояний от числа кластеров;
- для алгоритма K -внутригрупповых средних: график зависимости числа векторов признаков, сменивших номер кластера, от итерации алгоритма.

7 СПРАВОЧНЫЕ СВЕДЕНИЯ О ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON И СРЕДСТВАХ МАШИННОГО ОБУЧЕНИЯ

7.1 Язык Python

7.1.1 Общие сведения

Python – высокоуровневый язык программирования общего назначения [16]. «Python поддерживает структурное, обобщенное, объектно-ориентированное, функциональное и аспектно-ориентированное программирование. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты» [17]. Подробную информацию о работе с Python на английском и русском языках можно найти на специализированных сайтах [16] и [18]. Ниже представлены краткие сведения о языке и основных используемых библиотеках.

Переменные

Имя (идентификатор) переменной должно начинаться с латинской буквы любого регистра или подчеркивания, в имени можно использовать цифры. В качестве имени нельзя использовать ключевые слова.

Комментарии

Текст строки, расположенный справа от символа "#", считается комментарием.

Операторы

Оператор присвоения:

```
stringVariable = "world"
```

```
integerVariable = 5
```

realVariable = 5.2
logicalVariable = integerVariable > realVariable

Оператор условия

if условие_1:
операторы_1
elif условие_2:
операторы_2
...
elif условие_N:
операторы_N
else:
операторы

Операторы цикла

Python поддерживает два способа организации циклов.

Цикл по условию:

while условие:
операторы

Цикл по элементам последовательности:

for элемент in последовательность:
операторы

Пустой оператор

pass

Определение функций и классов

Определение функций и процедур

Для определения функции следует использовать ключевое слово *def*. Типовой вид функции следующий

@декоратор
def НазваниеФункции(parameter_list)
тело функции
return возвращаемоеФункциейЗначение

Здесь:

```
parameter_list ::= defparameter ( "," defparameter)* "," "/" [" ;"  
                        [parameter_list_no_posonly]]  
                        | parameter_list_no_posonly  
parameter_list_no_posonly ::= defparameter ( "," defparameter)* [" ;"  
                        [parameter_list_starargs]]  
                        | parameter_list_starargs  
parameter_list_starargs ::= "*" [parameter] ( "," defparameter)* [" ;" ["**" pa-  
parameter [ "," ;"]]]  
                        | "*" parameter [" ;"]  
parameter ::= identifier [":" expression]  
defparameter ::= parameter ["=" expression]
```

Пример:

```
@staticmethod  
def myFunction( inputValue ):  
    return inputValue + 1
```

Определение классов

Для определения класса следует использовать ключевое слово *class*. Методы класса также могут определяться с декораторами. Пример простейшего класса приведен ниже.

```
class CMyClass:  
    m_ClassMember = 'PythonClass'  
    def SetClassMemberValue( self, newName ):  
        return self.m_ClassMember = newName
```

7.1.2 Установка интерпретатора языка и пакетов расширений

Установка интерпретатора языка

Загрузка инсталлятора языка Python производится с сайта разработчика [16]. На указанном сайте выберете вкладку "Download", появится информация о последней версии языка и доступности ее в различных ОС, как на рисунке 7.1 ниже.

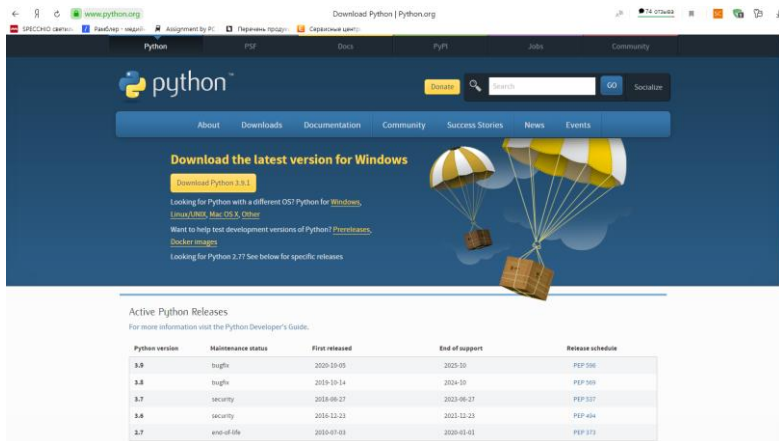


Рисунок 7.1 – Сайт разработчика языка Python

Скачайте инсталлятор, нажав на кнопку "Download Python 3...." и запустите его от имени Администратора для установки Python. Следуйте указаниям инсталлятора. Для проверки успешности установки Python можно проверить его наличие, версию и окружение, набрав и выполнив команду "python" из командной строки. Пример показан на рисунке 7.2 ниже.

```
C:\>python
Python 3.6.10 |Anaconda custom (64-bit)| (default, Mar 23 2020, 17:58:33) [MSC v
.1916 64 bit (AMD64)] on win32
```

Рисунок 7.2 – Информация об установленной версии Python и использованном окружении

В Python нет предустановленной среды разработки – ее установка и использование не является обязательной. Можно написать текст программы на Python в любом текстовом редакторе (Notepad++, gedit и др.), а запускать ее консолью, набрав:

```
python ProgramFileNameWithPath.py
```

В то же время для написания, отладки и «контролируемого» исполнения программ существенно удобнее пользоваться специализированными средами разработки, подобными PyCharm (см. подраздел 7.2 настоящего издания).

Установка пакетов расширений

Для установки пакетов расширений используйте в командной строке вызов программы "pip" (программа pip вышла вместе с Python 3.4). Список команд данной программы представлен ниже на рисунке 7.3.



```
cmd
Командная строка

C:\>pip
Usage:
  pip <command> [options]

Commands:
  install             Install packages.
  download            Download packages.
  uninstall           Uninstall packages.
  freeze             Output installed packages in requirements format.
  list                List installed packages.
  show                Show information about installed packages.
  check              Verify installed packages have compatible dependencies.
  config             Manage local and global configuration.
  search              Search PyPI for packages.
  wheel              Build wheels from your requirements.
  hash                Compute hashes of package archives.
  completion         A helper command used for command completion.
  debug              Show information useful for debugging.
  help                Show help for commands.

General Options:
  -h, --help          Show help.
  --isolated           Run pip in an isolated mode, ignoring environment variables and user configuration.
  -v, --verbose       Give more output. Option is additive, and can be used up to 3 times.
  -U, --version       Show version and exit.
  -q, --quiet         Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).
  --log <path>       Path to a verbose appending log.
  --proxy <proxy>     Specify a proxy in the form [user:passwd@]proxy.server:port.
  --retries <retries> Maximum number of retries each connection should attempt (default 5 times).
  --timeout <sec>    Set the socket timeout (default 15 seconds).
  --exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.
  --trusted-host <hostname> Mark this host or host:port pair as trusted, even though it does not have valid or any HTTPS.
  --cert <path>      Path to alternate CA bundle.
  --client-cert <path> Path to SSL client certificate, a single file containing the private key and the certificate in PEM format.
  --cache-dir <dir>  Store the cache data in <dir>.
  --no-cache-dir     Disable the cache.
  --disable-pip-version-check Don't periodically check PyPI to determine whether a new version of pip is available for download. Implied with --no-index.
  --no-color         Suppress colored output
  --no-python-version-warning Silence deprecation warnings for upcoming unsupported Pythons.

C:\>_
```

Рисунок 7.3 – Команда pip

Установка, обновление и просмотр списка установленных пакетов расширений производится, соответственно, следующим образом:

```

pip install PackageName
pip install -U PackageName
pip list

```

Убедитесь, что указанные в таблице 7.1 ниже пакеты у вас установлены, они необходимы для выполнения настоящего издания.

Таблица 7.1. Список требуемых пакетов расширения языка Python

<i>Наименование пакета</i>	<i>Назначение</i>
<i>matplotlib</i>	Пакет для работы с графиками и диаграммами, для графического представления результатов научных исследований: https://matplotlib.org
<i>numpy</i>	Пакет для работы с массивами, матрицами и тензорами (многомерными матрицами): https://numpy.org/
<i>qpsolvers</i>	Пакет для решения задач квадратичного программирования: https://pypi.org/project/qpsolvers/
<i>pillow</i>	Пакет для работы с цифровыми изображениями: https://python-pillow.org/
<i>Scipy</i>	Пакет для научных вычислений (быстрые преобразования Фурье и т.п.): https://www.scipy.org/
<i>scikit-learn</i>	Пакет машинного обучения (см. раздел 7.3.2)
<i>csv</i>	Пакет для записи/чтения данных в файлы <i>csv</i> формата: https://docs.python.org/3/library/csv.html

Для использования соответствующего установленного пакета и находящихся в нем функций/классов необходимо в начале программы указать¹:

¹ Обратите внимание, что имена пакетов при их инсталляции и при их использовании внутри Python могут отличаться. Смотрите справочную документацию по используемым пакетам.

import *packageName*

или¹

import *packageName as packageShortName*

Тогда доступ к конкретной функции или классу производится в виде

packageName.FuncOrClassName

Если из всего пакета достаточно использование заданной функции/класса, то используйте следующую конструкцию:

from *packageName import funcOrClassName*

В такой ситуации доступ к конкретной функции осуществляется напрямую по ее имени, то есть без имени пакета.

На состав используемых пакетов указывают стартовые строки программы на Python, как показано ниже².

```
import csv
import numpy
import random
import math
import matplotlib as plt
from PIL import Image
# import scipy
from scipy.fftpack import dct, idct
#import sklearn
from sklearn import linear_model
#import qpsolvers
from qpsolvers import solve_qp
```

¹ Многие сокращенные названия пакетов стали, де факто, стандартными. Например, для пакета matplotlib часто используется сокращение plt, для tensorflow – tf и др. Посмотрите на профессиональных сайтах и в руководствах примеры сокращений.

² Пакеты math, random и некоторые другие обычно входят в состав базового пакета, устанавливаемого вместе с Python.

7.2 PyCharm – интегрированная среда разработки для языка программирования Python

Загрузка инсталлятора интегрированной среды разработки для языка программирования Python – PyCharm – производится с сайта разработчика [20]. На указанном сайте, представленном на рисунке 7.4, выберете кнопку «Скачать».

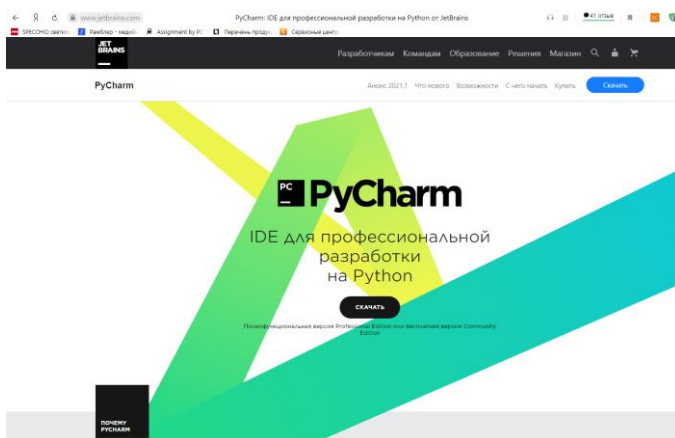


Рисунок 7.4 – Сайт разработчика интегрированной среды разработки PyCharm

Браузер перейдет на страницу выбора версии, показанной на рисунке 7.5.

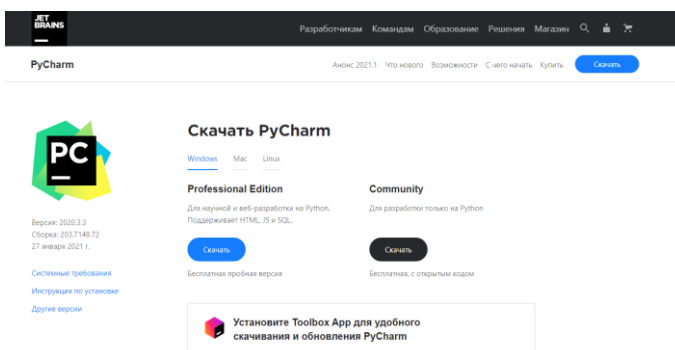


Рисунок 7.5 – Выбор версии PyCharm

Нажмите кнопку «Скачать» для бесплатной версии "Community" для получения соответствующего инсталлятора. Скачав инсталлятор, запустите его от имени Администратора для установки IDE PyCharm, следуйте указаниям инсталлятора.

Для работы в IDE запустите PyCharm из списка установленных



программ или с помощью иконки . Появится окно запуска среды разработки, показанное на рисунке 7.6. Здесь вы можете создать новый проект или открыть существующий, указав место его расположения.

В результате создания нового или открытия существующего проекта будет появление основного окна среды разработки, аналогичного показанному на рисунке 7.7.

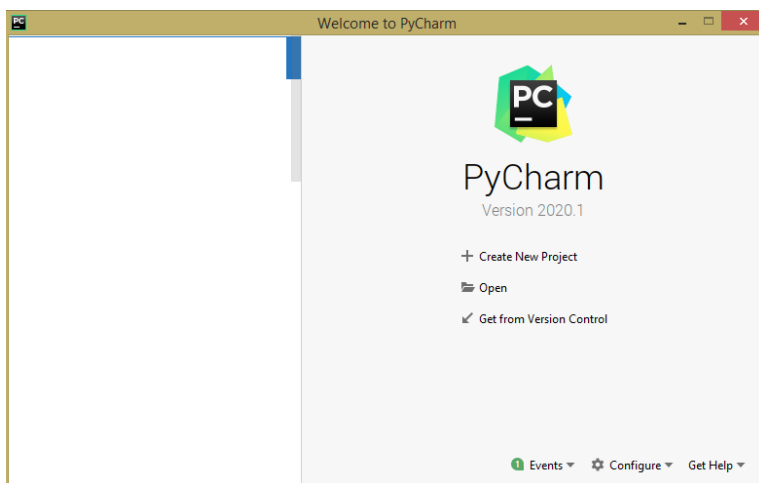



Рисунок 7.6 – Запуск среды разработки

Для запуска программы используются команды меню "Run\Run" или "Run\Debug", а также клавиши , отвечающие, соответственно, за запуск или отладку. При первом запуске, возможно, команды запуска или отладки надо выполнить из контекстного меню, вызванного на форме конкретного файла проекта

(рисунок 7.8). Результат работы программы показан в нижней части окна PyCharm, как показано на рисунке 7.9.



Рисунок 7.7 – Окно среды разработки

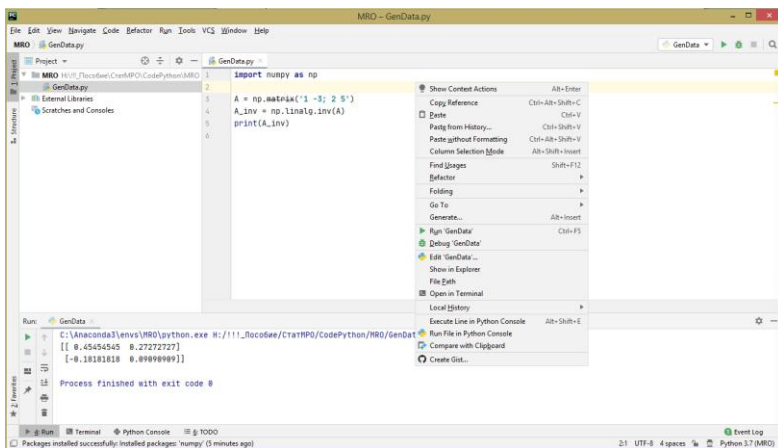


Рисунок 7.8 – Контекстное меню для текущего файла Python



Рисунок 7.9 – Результат запуска файла исполнения Python

7.3 Библиотека машинного обучения scikit-learn (machine learning in Python)

Библиотека scikit-learn – одна из наиболее известных и наиболее часто используемых открытых библиотек по машинному обучению, распознаванию образов и анализу данных [21,22]. Данная библиотека включает в себя средства решения задач:

- управляемой классификации,
- автоматической классификации (кластеризации),
- регрессии,
- снижения размерности данных,
- подбора модели данных,
- предобработки и отбора признаков.

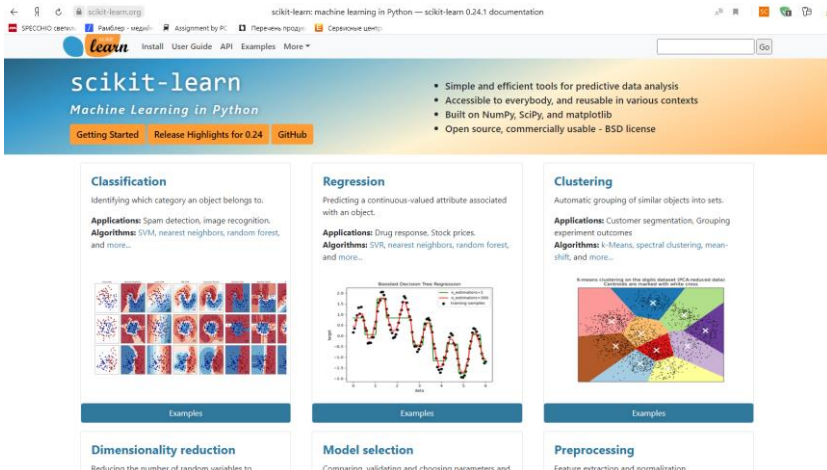


Рисунок 7.10 Сайт библиотеки машинного обучения scikit-learn

Установка библиотеки scikit-learn в Python производится командой pip (см. раздел 7.1):

```
pip install -U scikit-learn
```

В таблице 7.2 представлено соответствие между методами классификации, изложенными в настоящем издании, и функциях библиотеки `scikit-learn`.

Ниже, в подразделе 7.3 приведены примеры синтеза данных и их классификации с использованием библиотеки `scikit-learn`.

Для работы с методами машинного обучения, использующими нейронные сети глубокого обучения, используется библиотека `tensorflow`. Описание этой библиотеки выходит за рамки настоящего издания.

Таблица 7.2 Соответствие разделов издания и руководства библиотеки `scikit-learn`

<i>Раздел издания</i>	<i>Раздел UserGuide сайта scikit-learn [21]</i>
2 – Оптимальные стратегии...	1.9. Naive Bayes <code>from sklearn.naive_bayes import GaussianNB</code>
3 – Линейные классификаторы	1.1 Linear Models <code>from sklearn import linear_model</code>
4 – Линейные классификатора. Метод опорных векторов	1.4. Support Vector Machines <code>from sklearn import svm</code>
5 – Классификаторы, основанные на непараметрических оценках...	1.6. Nearest Neighbors <code>from sklearn.neighbors import NearestNeighbors</code>
6 – Автоматическая классификация	2.3. Clustering <code>from sklearn.cluster import KMeans</code>

7.4 Прикладная программа синтеза и классификации данных

7.4.1 Синтеза данных для классификации

Код программы синтеза данных с комментариями и результаты его запуска представлены на рисунке 7.11 ниже.

```

1 import numpy as np
2 import math
3 import random
4 import matplotlib.pyplot as plt
5
6 def GenerateRandomVectors(meanVector, covarianceMatrix, N, savedFileName):
7     A = np.zeros((2,2))
8     A[0,0] = math.sqrt(covarianceMatrix[0, 0])
9     A[1,0] = covarianceMatrix[1, 0] / math.sqrt(covarianceMatrix[0, 0])
10    A[1,1] = math.sqrt(covarianceMatrix[1, 1] - covarianceMatrix[0, 1] / covarianceMatrix[0, 0])
11    #print(A) # transformation matrix for generation
12
13    vectorsNorm01 = np.random.randn(2, N) # (0,1) - normal vectors
14    #plt.plot(vectorsNorm01[0,:], vectorsNorm01[1,:], color='green', marker='o', linestyle='none')
15
16    x = np.matmul(A, vectorsNorm01) + np.repeat(meanVector, N, 1) # generated vectors with given mean M and covariance B
17    np.save(savedFileName, x)
18    #plt.plot(x[0,:], x[1,:], color='green', marker='X', linestyle='none') # plot generated data to check
19    #plt.show() # show plot
20
21
22
23 N = 100
24 M = np.array([[5],[4]]) # mean vector
25 B = np.array([[5, 2],[2, 1]]) # covariance matrix
26 fileName2save = 'arrayx.npy'
27 GenerateRandomVectors(M, B, N, fileName2save)
28 # check save data
29 z = np.load(fileName2save)
30 plt.plot(z[0, :], z[1, :], color='red', marker='.', linestyle='none') # plot saved and loaded generated data
31 plt.show()

```

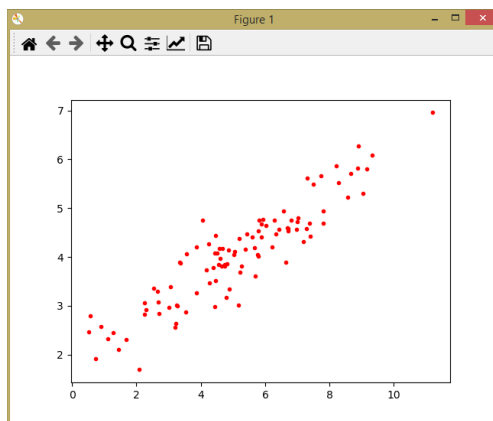


Рисунок 7.11 – Листинг программы генерации данных и результат ее работы

7.4.2 Классификации данных с использованием метода опорных векторов на *scikit-learn*

Код программы классификации данных с комментариями и результаты его запуска представлены на рисунке 7.12 ниже.

```

import numpy as np
import math

```

```

import matplotlib.pyplot as plt
from sklearn import svm

def GenerateRandomVectors(...):
    ...
    ...

N = 100
M = np.array([[4],[1]]) # mean vector
B = np.array([[5, 1], [1, 1]]) # covariance matrix
fileName2Save0 = 'array0.npy'
GenerateRandomVectors( M, B, N, fileName2Save0 )

M = np.array([[4],[0]]) # mean vector
B = np.array([[5, -2], [-2, 1]]) # covariance matrix
fileName2Save1 = 'array1.npy'
GenerateRandomVectors( M, B, N, fileName2Save1 )

# check save data
z0 = np.load( fileName2Save0 )
z1 = np.load( fileName2Save1 )
plt.plot(z0[0, :], z0[1, :], color='red', marker='.', linestyle='none') # plot generated data
plt.plot(z1[0, :], z1[1, :], color='blue', marker='.', linestyle='none') # plot generated data
plt.show()

X = np.concatenate((z0, z1), axis=1)
X = X.transpose()
yldeal = np.zeros((2*N))
yldeal[N:2*N] = 1
clf = svm.LinearSVC() # linear version of SVM. Instead of svm.SVC()
clf.fit(X, yldeal)

# division boundary for linear classifier
x_1 = -10
y_1 = - ( clf.coef_[0,0] * x_1 + clf.intercept_ ) / clf.coef_[0,1]

```

```

x_2 = 10
y_2 = - ( clf.coef_[0,0] * x_2 + clf.intercept_ ) / clf.coef_[0, 1]

plt.plot(z0[0, :], z0[1, :], color='red', marker='.', linestyle='none') # plot gener-
ated data
plt.plot(z1[0, :], z1[1, :], color='blue', marker='.', linestyle='none') # plot gener-
ated data
plt.plot( [x_1,x_2], [ y_1, y_2], color='green', marker='X', linestyle='solid') #
plot classifier
plt.show()

yPredicted = clf.predict(X)
yDif = np.abs( yIdeal - yPredicted )
Nerr = np.sum(yDif)
yDif01 = yDif[0:N]
yDif10 = yDif[N:2*N]
N01 = np.sum(yDif01)
N10 = np.sum(yDif10)
print(Nerr/N)
print(N01/N)
print(N10/N)

```

Результат запуска программы показан ниже и отражает значения суммарной ошибки (0.02) и ошибок первого и второго рода (0.02 и 0, соответственно) для построенного классификатора, что можно проверить по рисунку 7.12.

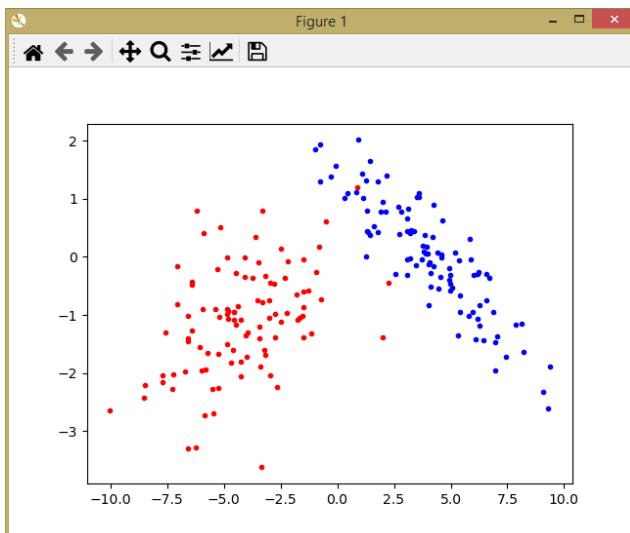
```
C:\Anaconda3\envs\MRO\python.exe C:/MRO/GenData.py
```

```
0.02
```

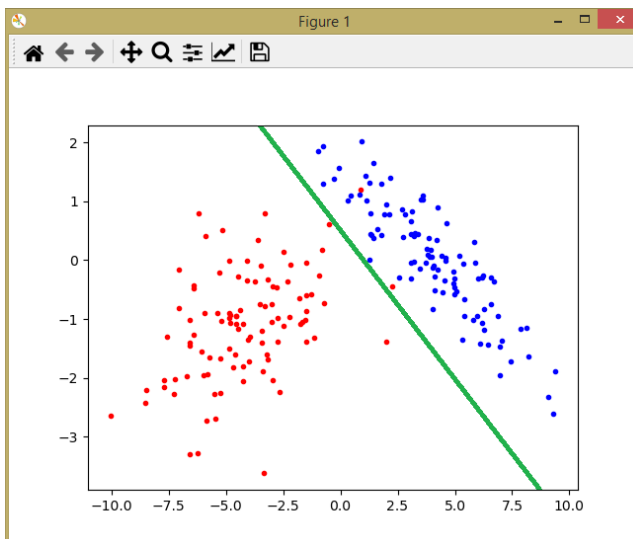
```
0.02
```

```
0.0
```

```
Process finished with exit code 0
```

а



б

Рисунок 7.12 – Исходные данные двух классов (а) и их разделение с использованием линейной разделяющей границы (б), построенной с использованием метода опорных векторов

8 КОНТРОЛЬНЫЕ ВОПРОСЫ

8.1 Контрольные вопросы к лабораторной работе № 1

1. Алгоритм моделирования нормально распределенного случайного вектора.
2. Вид матрицы линейного преобразования, используемой для моделирования нормально распределенного случайного вектора.
3. Оценивание параметров нормального закона распределения.
4. Выражения для рекуррентного оценивания параметров нормального закона распределения.
5. Меры близости нормальных распределений.
6. Инвариантность расстояний к линейным преобразованиям.
7. Характер линейного преобразования, обеспечивающего инвариантность евклидова расстояния.
8. Алгоритм моделирования бинарного случайного вектора с независимыми координатами.
9. Структура прикладной программы на языке Python.

8.2 Контрольные вопросы к лабораторной работе № 2

1. Постановка задачи классификации.
2. Определение классификатора.
3. Способы задания классификатора.
4. Качество классификатора.
5. Байесовский классификатор.
6. Минимаксный классификатор.
7. Классификатор Неймана-Пирсона.
8. Отношение правдоподобия.
9. Байесовский классификатор для нормально распределенных признаков.

10. Байесовский классификатор для распознавания бинарных признаков.

11. Вычисление вероятностей ошибочной классификации.

8.3 Контрольные вопросы к лабораторной работе № 3

1. Процедура Петерсона-Матсона построения линейной дискриминантной функции, минимизирующей суммарную вероятность ошибочной классификации.

2. Классификатор Фишера.

3. Линейная разделяющая функция, минимизирующая СКО решения.

4. Алгоритм перцептрона.

5. Стохастическая аппроксимация. Идея метода.

6. Процедура Роббинса-Монро.

7. Схема построения линейных классификаторов, основанная на методе стохастической аппроксимации.

8. АКП-алгоритм.

9. НСКО-алгоритм.

10. Способы ускорения сходимости алгоритмов построения линейных классификаторов, основанных на методе стохастической аппроксимации.

8.4 Контрольные вопросы к лабораторной работе № 4

1. Понятие разделяющей гиперплоскости и разделительной полосы при классификации линейно-разделимых классов.

2. Идея метода опорных векторов для линейно-разделимых классов. Формулировка первичного критерия.

3. Формулировка задачи построения оптимальной разделяющей гиперплоскости с использованием задачи квадратичного программирования.

4. Понятие опорного вектора.
5. Метод опорных векторов для случая линейно неразделимых классов. В чем отличие постановки задачи от случая линейно-разделимых классов.
6. Использование ядер в методе опорных векторов и переход в пространства высокой размерности.
7. Примеры ядер в методе опорных векторов.
8. Вид классификатора при использовании ядер в методе опорных векторов.
9. Достоинства метода опорных векторов.
10. Недостатки метода опорных векторов.

8.5 Контрольные вопросы к лабораторной работе № 5

1. Вид классификатора Байеса для простейшей матрицы штрафа. Варианты построения классификаторов, основанных на оценивании плотности вероятностей.
2. Метод Парзена оценивания плотности вероятностей.
3. Требования к ядрам и параметру оценки.
4. Утверждение о свойствах оценки Парзена плотности вероятностей.
5. Примером многомерного ядра, имеющего вид нормальной ПВ.
6. Связь ковариационной матрицы оценки Парзена и эмпирической оценки матрицы по выборке.
7. Виды в примеры ядер оценки Парзена.
8. Достоинства и недостатки оценки Парзена при построении классификаторов.
9. Метод К ближайших соседей.
10. Утверждение о свойствах оценки плотности вероятностей методом К ближайших соседей.
11. Решающее правило К ближайших соседей.

12. Решающее правило К ближайших соседей для двух классов.
13. Решающее правило ближайшего соседа.
14. Достоинства и недостатки решающего правила К соседей.
15. Оценка априорных вероятностей.

8.6 Контрольные вопросы к лабораторной работе № 6

1. Меры сходства, используемые при автоматической классификации.
2. Критерии, используемые при автоматической классификации.
3. Простой алгоритм выделения кластеров.
4. Достоинства и недостатки простого алгоритма выделения кластеров.
5. Максиминный алгоритм.
6. Достоинства и недостатки максиминного алгоритма.
7. Алгоритм К внутригрупповых средних.
8. Достоинства и недостатки алгоритма К внутригрупповых средних.
9. Какие из представленных алгоритмов являются империческими, а какие используют для кластеризации критерий качества?

9 ВАРИАНТЫ ЗАДАНИЙ

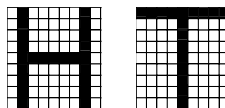
9.1 Варианты заданий к лабораторным работам № 1–5

Вариант Математические ожидания трех наборов нормально распределенных случайных векторов

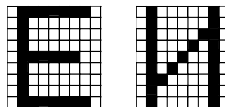
Представители бинарных случайных векторов,

□ ~ "0", ■ ~ "1"

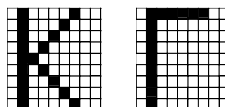
1. $\bar{M}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$



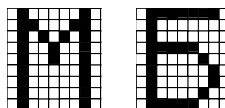
2. $\bar{M}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$



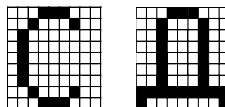
3. $\bar{M}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$



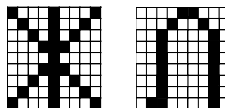
4. $\bar{M}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$



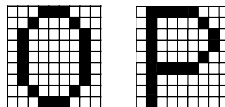
5. $\bar{M}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$



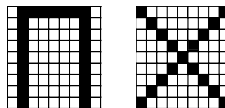
6. $\bar{M}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$



7. $\bar{M}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$



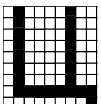
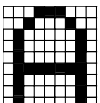
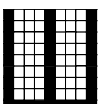
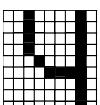
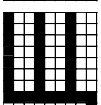
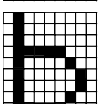
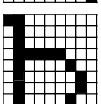
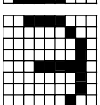
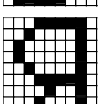
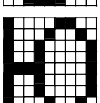
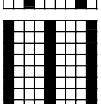
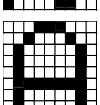
8. $\bar{M}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$



Вариант Математические ожидания трех наборов нормально распределенных случайных векторов

Представители бинарных случайных векторов,

□ ~ "0", ■ ~ "1"

9. $\bar{M}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}.$  
10. $\bar{M}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$  
11. $\bar{M}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$  
12. $\bar{M}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 1 \\ -2 \end{pmatrix}.$  
13. $\bar{M}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$  
14. $\bar{M}_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}.$  

9.2 Варианты заданий к лабораторной работе 6

№ Математические ожидания пяти наборов нормально распределенных случайных векторов

1. $\bar{M}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$
2. $\bar{M}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}.$

№

Математические ожидания пяти наборов
нормально распределенных случайных векторов

3.
$$\bar{M}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

4.
$$\bar{M}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

5.
$$\bar{M}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} -1 \\ -2 \end{pmatrix}.$$

6.
$$\bar{M}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} -2 \\ -1 \end{pmatrix}.$$

7.
$$\bar{M}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

8.
$$\bar{M}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 0 \\ -2 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 0 \\ -3 \end{pmatrix}.$$

9.
$$\bar{M}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -2 \\ 1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

10.
$$\bar{M}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

11.
$$\bar{M}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}.$$

12.
$$\bar{M}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

13.
$$\bar{M}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

14.
$$\bar{M}_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \bar{M}_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \bar{M}_3 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}, \bar{M}_4 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \bar{M}_5 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

ЗАКЛЮЧЕНИЕ

Настоящее издание является сборником методических указаний к лабораторным работам по статистической теории распознавания образов, для каждой из которых приводятся описание теоретических основ, задание и список контрольных вопросов.

Издание покрывает вопросы построения и использования ставших классическими методов и алгоритмов статистической теории распознавания образов. В частности:

- методы оптимальной классификации (классификатор Байеса, минимаксный классификатор и классификатор Неймана-Пирсона);

- методы построения и настройки линейных классификаторов и классификаторов, основанных на оценивании плотностей вероятностей;

- методы автоматической классификации – кластеризации и таксономии.

Заключительная часть издания посвящена вопросам установки Python и иного программного обеспечения, необходимого для его установки, конфигурирования и использования при выполнении этого издания.

Важно отметить, что данный практику посвящен основным (статистическим) методам машинного обучения и распознавания образов, которые стали к настоящему времени уже классическими и реализованы в рамках библиотеки машинного обучения `scikit-learn`. Поэтому многие вопросы современного машинного обучения, в частности, нейронные сети и глубокое обучение остались за рамками рассмотренного материала.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Анисимов, Б.В. Распознавание и цифровая обработка изображений / Б. В. Анисимов, В. Д. Курганов, В. К. Злобин. – Москва: Высшая школа, 1983. – 295 с.
2. Распознавание образов: состояние и перспективы: перевод с английского / К. Верхаген, Р. Дейн, Ф. Грун [и др.]. – Москва: Радио и связь, 1985. – 103 с.
3. Горелик, А.Л. Методы распознавания / А.Л. Горелик, В. А. Скрипкин. – Москва: Высшая школа, 1984. – 208 с.
4. Горелик, А.Л. Современное состояние проблемы распознавания / А. Л. Горелик, И. Б. Гуревич, В. А. Скрипкин. – Москва: Высшая школа, 1985. – 160 с.
5. Дуда, Р. Распознавание образов и анализ сцен: перевод с английского / Р. Дуда, П. Харг. – Москва: Мир, 1976. – 512 с.
6. Методы компьютерной обработки изображений / под редакцией В.А. Сойфера. – Москва: Физматлит, 2003. – 784 с.
7. Розенблатт, Ф. Принципы нейродинамики / Ф. Розенблатт. – Москва: Мир, 1965. – 480 с.
8. Ту, Дж. Принципы распознавания образов: перевод с английского / Дж. Ту, Р. Гонсалес. – Москва: Мир, 1978. – 412 с.
9. Фомин, Я.А. Статистическая теория распознавания образов / Я.А. Фомин, Г.Р. Тарловский. – Москва: Радио и связь, 1986. – 264 с.
10. Фор, А. Восприятие и распознавание образов: перевод с английского / А. Фор. – Москва: Машиностроение, 1989. – 272 с.
11. Фу, К. Последовательные методы в распознавании образов и обучении машин: перевод с английского / К. Фу. – Москва: Наука, 1971. – 256 с.

12. Фукунага, К. Введение в статистическую теорию распознавания образов: перевод с английского / К. Фукунага. – Москва: Наука, 1979. – 368 с.
13. Вапник, В.Н. Восстановление зависимостей по эмпирическим данным / В.Н. Вапник. – Москва: Наука, 1979. – 448 с.
14. Воронцов, К.В. Машинное обучение. Курс лекций [сайт]. – 2023. – URL: http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_%28курс_лекций%2C_К.В.Воронцов%29 (дата обращения: 01.03.2021).
15. Мину, М. Математическое программирование. Теория и алгоритмы: перевод с французского и предисловие А.И. Штерна / М. Мину. – Москва: Наука, 1990.
16. Python [сайт]. – 2023. – URL: <https://www.python.org/> (дата обращения: 01.03.2021).
17. Python. Типы_и_структуры_данных [сайт]. – 2023. – URL: https://ru.wikipedia.org/wiki/Python#Типы_и_структуры_данных (дата обращения: 01.03.2021).
18. Pythonworld [сайт]. – 2023. – URL: <https://pythonworld.ru> (дата обращения: 01.03.2021).
19. Прохоренок, Н. Введение в Python 3. Самое необходимое / Н. Прохоренок, В. Дронов. – 2-е изд. – БХВ-Петербург, 2019. – 608 с.
20. Pycharm [сайт]. – 2023. – URL: <https://www.jetbrains.com/ru-ru/pycharm/> (дата обращения: 01.03.2021).
21. SciKit-Learn. Machine Learning in Python [сайт]. – 2023. – URL: <https://scikit-learn.org/> (дата обращения: 01.03.2021).
22. Scikit-learn: Machine Learning in Python / Fabian Pedregosa; Gaël Varoquaux; Alexandre Gramfort [etc.] // Journal of Machine Learning Research. – 2011. – № 12. – P. 2825–2830.

Учебное издание

Мясников Владислав Валерьевич

**ОСНОВЫ СТАТИСТИЧЕСКОЙ ТЕОРИИ
РАСПОЗНАВАНИЯ ОБРАЗОВ
И МАШИННОГО ОБУЧЕНИЯ**

Практикум

Редакционно-издательская обработка А.С. Никитиной

Подписано в печать 27.06.2023. Формат 60x84 1/16.

Бумага офсетная. Печ. л. 7,75.

Тираж 27 экз. Заказ . Арт. – .

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)
443086, САМАРА, МОСКОВСКОЕ ШОССЕ, 34.

Издательство Самарского университета.
443086, Самара, Московское шоссе, 34.