Интеллектуальный анализ данных
(Data Mining)

Шорохов С.Г.

кафедра математического моделирования и искусственного интеллекта

Лекция 1. Препроцессинг данных

БРС по дисциплине

Занятия по дисциплине «Интеллектуальный анализ данных» включают:

- лекции (теоретический материал) – 20 баллов
- 7 лабораторных работ (задания по программированию) – 80 баллов

Для каждого направления подготовки будет создана отдельная команда MS Teams, для каждой лабораторной работы в рамках команды будет создан отдельный канал. Варианты заданий будут направляться через записную книжку команды.

Отчет по лабораторной работы должен быть представлен в виде файла ipynb, представляющего собой файл Jupiter Notebook с титульным листом, вариантом задания, программным кодом, комментариями к программному коду и результатами выполнения программного кода. Отчет в формате ipynb передается как ответ на задание MS Teams. Моментом сдачи задания является дата и время отправки файла путем нажатия кнопки «сдать» в MS Teams.

Лабораторная работа оценивается с максимальной оценкой 10 баллов, если студент присутствовал на занятии, программа корректно решает поставленные задачи и отчет сдан в оговоренный срок. В случае отсутствия на занятии оценка снижается на 1 балл. В случае просрочки сдачи лабораторной работы оценка снижается на 1 балл за каждый день просрочки, но не более чем на 5 баллов. Лабораторные работы с просрочкой более 5 дней оцениваются в конце модуля.

Интеллектуальный анализ данных

Интеллектуальный анализ данных (data mining, добыча данных, майнинг данных) — это совокупность методов обнаружения в данных знаний:

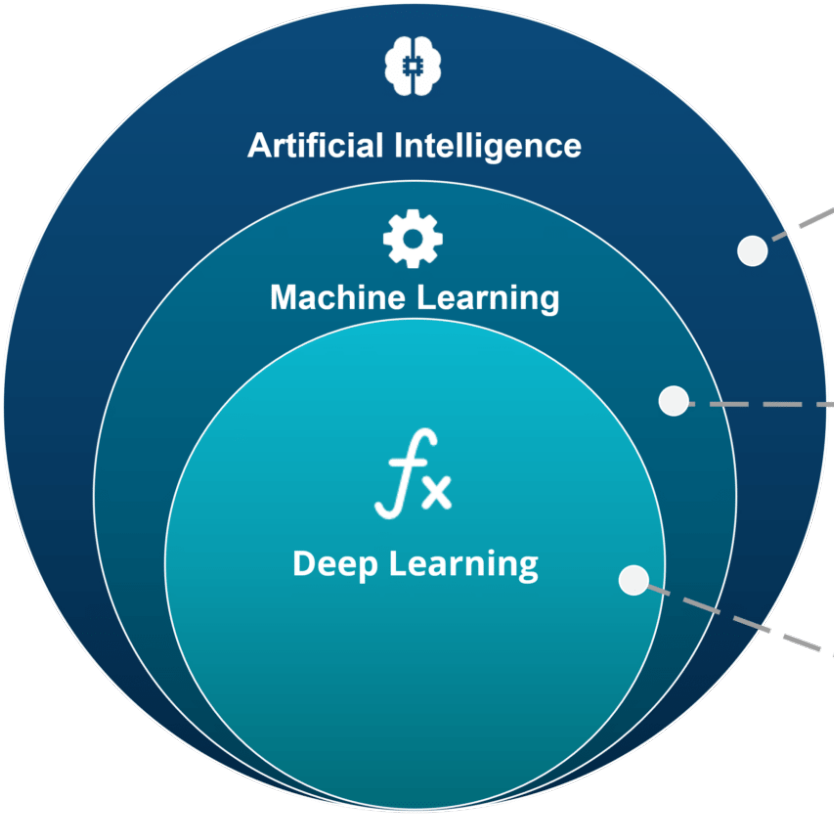
- ранее неизвестных;
- нетривиальных;
- практически полезных;
- доступных интерпретации,

необходимых для принятия решений в различных сферах деятельности.

В интеллектуальном анализе данных основным объектом изучения являются структурированные (табличные) данные.

Основными методами построения моделей интеллектуального анализа данных являются методы статистики и машинного обучения.

Искусственный интеллект и машинное обучение



ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ –
технологии, позволяющие компьютерным системам воспроизводить разумное поведение человека

МАШИННОЕ ОБУЧЕНИЕ –
набор технологий искусственного интеллекта, позволяющий улучшать качество работы компьютерных систем по мере накопления опыта

ГЛУБОКОЕ ОБУЧЕНИЕ –
совокупность методов машинного обучения, основанных на использовании различных архитектур многослойных нейронных сетей

Интеллектуальный анализ данных: темы

1. Препроцессинг данных.
2. Поиск ассоциативных правил.
3. Кластеризация данных.
4. Классификация данных.
5. Бинарная классификация данных.
6. Деревья решений.
7. Регрессия.

Литература

1. M.Zaki, W.Meira, Data Mining and Machine Learning. Fundamental Concepts and Algorithms 2e (2020)
2. К.Мэрфи, Вероятностное машинное обучение: введение (2022)
3. У. Маккинни, Python и анализ данных 3е издание (2023)
4. Дж. Вандер Плас, Python для сложных задач. Наука о данных 2е издание (2024)

Наборы данных

Анализируемые данные часто могут быть представлены в качестве матрицы данных $\mathbf{D}$ размером $n \times d$, имеющей n строк и d столбцов:

$$\mathbf{D} = \left(\begin{array}{c|cccc} & \mathbf{X}_1 & \mathbf{X}_1 & \cdots & \mathbf{X}_1 \\ \hline \mathbf{x}_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \mathbf{x}_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{array}\right)$$

Строки соответствуют сущностям (записям) в наборе данных, а столбцы представляют признаки (атрибуты, свойства) данных. Каждая строка содержит наблюдаемые значения признаков для заданной сущности:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$$

Каждый столбец содержит наблюдаемые значения заданного признака для различных сущностей:

$$\mathbf{X}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})^T$$

Пример: набор данных «Ирисы»

Набор данных «Ирисы» изучался Р. Фишером в 1936 г. Набор состоит из 150 записей и содержит следующие пять признаков:

1. длина чашелистника (sepal length) в см
2. ширина чашелистника (sepal width) в см
3. длина лепестка (petal length) в см
4. ширина лепестка (petal width) в см
5. класс (class) ириса, принимающий значения:
   - Iris-setosa
   - Iris-versicolour
   - Iris-virginica

Набор данных «Ирисы» размещен в репозитарии данных машинного обучения UCI по адресу https://archive.ics.uci.edu/ml/datasets/Iris, количество обращений к набору с 2007 года составляет более 4 миллионов.

Матрица данных набора «Ирисы»

	Sepal length $\mathbf{X}_1$	Sepal width $\mathbf{X}_2$	Petal length $\mathbf{X}_3$	Petal width $\mathbf{X}_4$	Class $\mathbf{X}_5$
$\mathbf{x}_1$	5.9	3.0	4.2	1.5	Iris-versicolor
$\mathbf{x}_2$	6.9	3.1	4.9	1.5	Iris-versicolor
$\mathbf{x}_3$	6.6	2.9	4.6	1.3	Iris-versicolor
$\mathbf{x}_4$	4.6	3.2	1.4	0.2	Iris-setosa
$\mathbf{x}_5$	6.0	2.2	4.0	1.0	Iris-versicolor
$\mathbf{x}_6$	4.7	3.2	1.3	0.2	Iris-setosa
$\mathbf{x}_7$	6.5	3.0	5.8	2.2	Iris-virginica
$\mathbf{x}_8$	5.8	2.7	5.1	1.9	Iris-virginica
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{x}_{149}$	7.7	3.8	6.7	2.2	Iris-virginica
$\mathbf{x}_{150}$	5.1	3.4	1.5	0.2	Iris-setosa

Числовые и категориальные признаки

- Числовые признаки (атрибуты)

Числовой признак принимает вещественные или целочисленные значения. Числовые признаки, принимающие конечное или бесконечное счетное множество значений, называют дискретными. Числовые признаки, принимающие любое вещественное значение, называют непрерывными. Специальным случаем дискретного числового признака является признак со множеством значений $\{0, 1\}$, называемый бинарным.

- Категориальные признаки (атрибуты)

Категориальным является признак, у которого множество значений представляет собой множество символов.
Примерами категориальных признаков являются признаки “Пол” со значениями $\{\text{М}, \text{Ж}\}$ или “Образование” со значениями $\{S, B, M, A, PhD\}$.
Категориальные признаки могут быть двух типов:

- Номинальные – значения не упорядочены, поэтому возможны только сравнения типа равенств (пример: признак “Пол”).
- Ординальные – значения упорядочены, поэтому возможны сравнения типа равенств и типа неравенств (пример: признак “Образование”).

Набор данных как случайная выборка

Для случайной величины X случайная выборка размера n определяется как набор n независимых одинаково распределённых (i.i.d.) случайных величин S_1, S_2, ..., S_n, где S_i имеют то же вероятностное распределение, что и X, и независимы.

Для заданного набора данных $\mathbf{D}$ строки $\mathbf{x}_i$, $i = \overline{1,n}$ могут быть интерпретированы как d-мерная случайная выборка размера n значений векторной случайной величины $\mathbf{X} = (X_1, X_2, ..., X_d)$.

Поэтому при анализе набора данных $\mathbf{D}$ важное значение имеют такие статистические показатели, как:

- выборочные средние признаков
- выборочные дисперсии признаков
- эмпирические функции распределения признаков
- квантили распределения признаков
- выборочные ковариации и корреляции признаков

Формула Байеса в машинном обучении

Условная вероятность $\mathbb{P}[A \mid B]$— это вероятность наступления события A, если известно, что произошло другое событие B.

В машинном обучении формула Байеса записывается в следующем виде:

$$\mathbb{P}[\theta \mid \mathbf{D}] = \frac{\mathbb{P}[\theta]\,\mathbb{P}[\mathbf{D} \mid \theta]}{\mathbb{P}[\mathbf{D}]} = \frac{\mathbb{P}[\theta]\,\mathbb{P}[\mathbf{D} \mid \theta]}{\sum_{\theta' \in \Theta} \mathbb{P}[\mathbf{D} \mid \theta']\,\mathbb{P}[\theta']}$$

Здесь использована общепринятая в машинном обучении терминология:

- $\mathbf{D}$ – набор данных
- θ – параметры модели
- $\mathbb{P}[\theta]$ – априорная вероятность (prior probability)
- $\mathbb{P}[\mathbf{D} \mid \theta]$ – правдоподобие (likelihood)
- $\mathbb{P}[\theta \mid \mathbf{D}]$ – апостериорная вероятность (posterior probability)
- $\mathbb{P}[\mathbf{D}]$ – предельное правдоподобие или обоснованность (evidence)

Максимизация апостериорной вероятности

Большинство задач машинного обучения имеют вид некоторой модели с параметрами θ и задача состоит в том, чтобы по набору данных $\mathbf{D}$ подобрать описывающие эти данные параметры модели θ наилучшим образом.

Для этого в машинном обучении ищут апостериорное распределение

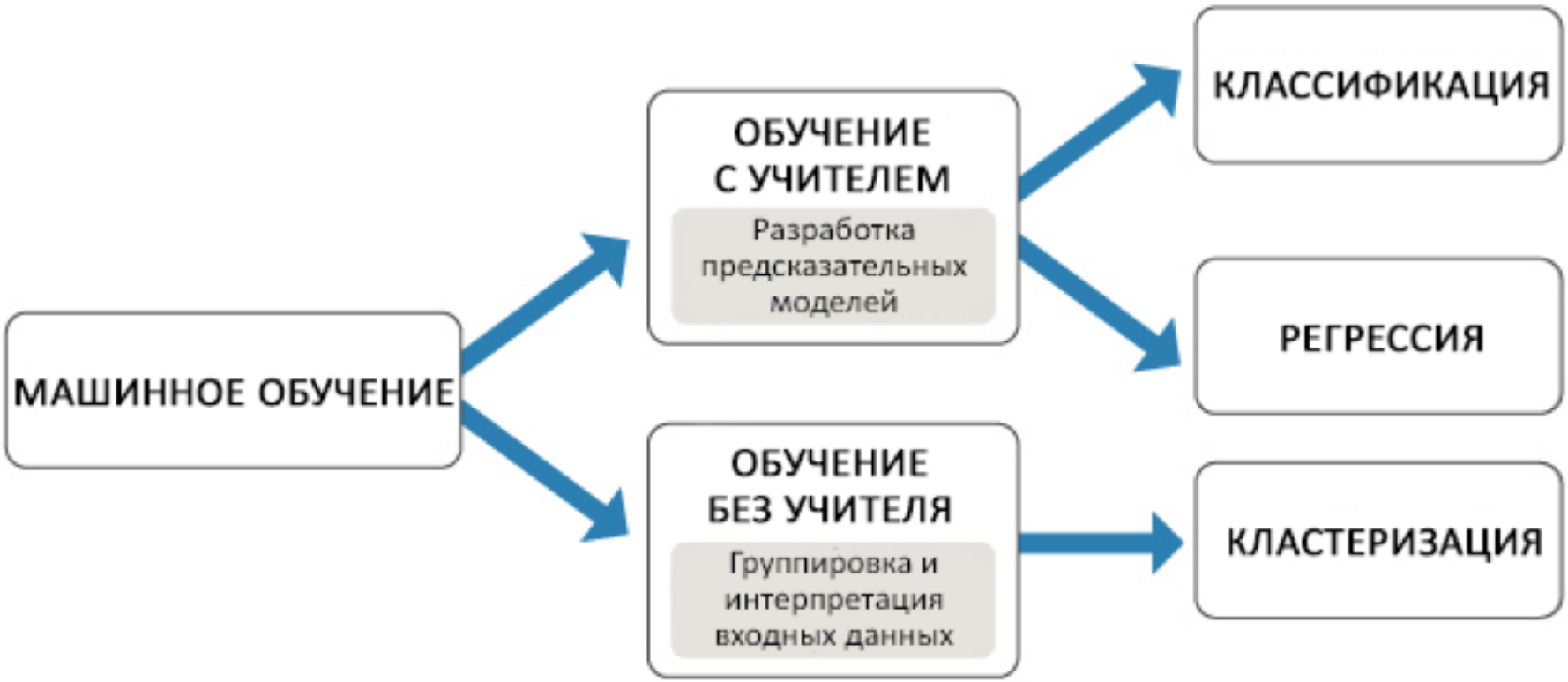
$$\mathbb{P}[\theta \mid \mathbf{D}] \propto \mathbb{P}[\theta]\,\mathbb{P}[\mathbf{D} \mid \theta]$$

(здесь значок $\propto$ означает пропорциональность, т.е. $\mathbb{P}[\theta \mid \mathbf{D}]$ пропорционально, а не равно $\mathbb{P}[\theta]\,\mathbb{P}[\mathbf{D} \mid \theta]$), затем определяют наилучший параметр модели θ^*:

$$\theta^* = \arg\max_\theta \mathbb{P}[\theta \mid \mathbf{D}] = \arg\max_\theta \mathbb{P}[\theta]\,\mathbb{P}[\mathbf{D} \mid \theta]$$

Теорема Байеса позволяет нам перейти от $\mathbb{P}[\theta \mid \mathbf{D}]$, которое обычно непонятно как подсчитать, к вычислению отдельно правдоподобия $\mathbb{P}[\mathbf{D} \mid \theta]$, которое определяется моделью, и априорного распределения $\mathbb{P}[\theta]$, которое можно выбрать удобным образом, например, чтобы упростить вычисление $\mathbb{P}[\theta \mid \mathbf{D}]$.

Основные задачи машинного обучения



Основные задачи машинного обучения

- Задача кластеризации — это задача группировки множества объектов на подмножества (кластеры) таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров по какому-либо критерию.
- В задаче классификации имеется множество объектов, разделённых некоторым образом на классы, и задано конечное подмножество объектов (выборка), для которых известно, к каким классам они относятся (классовая принадлежность остальных объектов неизвестна). Задача классификации – это задача построения алгоритма, способного классифицировать (определять номер или наименование класса) произвольного объекта из исходного множества.
- В задаче регрессии на множестве объектов определена некоторая (вообще говоря) неизвестная функция и задано конечное подмножество объектов (выборка), для которых известны значения функции. Задача регрессии – это задача построения алгоритма, способного находить значение функции для произвольного объекта из исходного множества.

Основные проблемы машинного обучения

Проблемы с данными:

- недостаточный объем обучающих данных
- нерепрезентативные обучающие данные
- данные низкого качества (с ошибками, выбросами, шумами, пропусками)
- нерелевантные признаки

Проблемы с моделями (алгоритмами):

- выбор подходящей модели
- недообучение на обучающих данных
- переобучение на обучающих данных
- тестирование качества модели
- выбор гиперпараметров модели

Препроцессинг данных

- подготовка данных (Data Preparation)
  - очистка данных (Data Cleaning)
  - преобразование данных (Data Transformation)
  - интеграция данных (Data Integration)
  - нормализация данных (Data Normalization)
  - заполнение пропущенных данных (Missing Data Imputation)
  - обработка шума (Noise Identification)
- сокращение данных (Data Reduction)
  - отбор признаков (Feature Selection)
  - извлечение признаков (Feature Extraction)
  - отбор сущностей (Instance Selection)
  - генерация сущностей (Instance Generation)
  - дискретизация (Discretization)

Препроцессинг данных

Препроцессинг (предварительная обработка) данных включает в себя широкий круг методов для очистки, выбора и преобразования данных с целью улучшения качества получаемых в дальнейшем моделей.

Плохое качество данных оказывает негативное воздействие на процесс анализа данных. Наиболее часто встречающиеся проблемы включают шум, выбросы, отсутствующие значения и дублирующиеся данные.

Программные средства для препроцессинга данных имеются как в библиотеке Pandas, так и основной библиотеке машинного обучения scikit-learn (sklearn).

Пропущенные значения

Достаточно часто в записи отсутствуют одно или несколько значений признаков. Иногда не хватает информации, а иногда некоторые значения не подходят для признака. Существуют различные стратегии работы с пропущенными (отсутствующими) значениями:

1. игнорировать запись (сущность) с пропущенным значением
2. ввести пропущенное значение в ручном режиме
3. использовать глобальную константу для заполнения пропущенных значений
4. использовать среднее значение или медиану признака для заполнения пропущенных значений
5. использовать среднее значение или медиану признака, вычисленные по записям, принадлежащим тому же классу
6. использовать наиболее вероятное значение признака для заполнения пропущенных значений

Дублирующиеся данные

Некоторые наборы данных, особенно полученные слиянием данных из нескольких источников, могут содержать дублирующиеся записи.

Из-за ошибок в процессе ввода различия в значениях некоторых атрибутов (например, в значении идентификатора) могут приводить к идентичным повторяющимся записям, считающимися разными. Такие записи не обнаруживаются простым сканированием набора данных на наличие повторяющихся записей.

Анализ сходства между текстовыми признаками весьма нетривиален, так как требуется вводить функции расстояния между текстами. Например, расстояние правки (редактирования, edit distance) – это способ количественной оценки того, насколько две строки отличаются друг от друга, путем подсчета минимального количества операций, необходимых для преобразования одной строки в другую. Обычно рассматриваются три типа операций редактирования: вставка символа, замена символа или удаление символа.

Шум в данных

Для того, чтобы убрать шум, данные подлежат сглаживанию.

1. В методе биннинга (binning method) данные сортируются, отсортированные данные распределяются на несколько корзин (или бинов) и для каждого бина данные заменяются на:
   - среднее значение по бину (bin mean)
   - медиану по бину (bin median)
   - ближайшую границу бина (bin boundary)
2. В методе регрессии исходные данные заменяются на данные, полученные в результате применения линейной, логистической и других видов регрессии.
3. При анализе выбросов (outlier analysis) данные, квалифицированные как выбросы, могут быть заменены на нормальные значения.

Метод биннинга: пример

Отсортированные данные (12 значений) распределены в 3 бина:

$$\underbrace{3, 8, 10, 15}_{Bin\,1}, \underbrace{21, 22, 24, 25}_{Bin\,2}, \underbrace{26, 27, 29, 34}_{Bin\,3}$$

Использование средних значений по бину (bin mean):

$$Bin\,1 : 9, 9, 9, 9$$
$$Bin\,2 : 23, 23, 23, 23$$
$$Bin\,3 : 29, 29, 29, 29$$

Использование медианы по бину (bin median):

$$Bin\,1 : 9, 9, 9, 9$$
$$Bin\,2 : 23, 23, 23, 23$$
$$Bin\,3 : 28, 28, 28, 28$$

Использование границ по бину (bin boundary):

$$Bin\,1 : 3, 3, 15, 15$$
$$Bin\,2 : 21, 21, 25, 25$$
$$Bin\,3 : 26, 26, 26, 34$$

Преобразование (трансформация) данных

1. сглаживание (биннинг, регрессия, кластеризация)
2. построение новых признаков – добавление новых признаков для облегчения майнинга данных
3. агрегирование данных (суммирование)
4. стандартизация и нормализация данных
5. дискретизация данных (значения числового признака заменяются на интервальные метки)
6. замена значений признака на значения более высокого уровня в иерархии (улица заменяется на город или страну)

Нормализация данных

1. нормализация min-max (масштабирование)
   - Пусть min_A и max_A– это минимальное и максимальное значения признака A. Тогда нормализация min-max преобразует значения ν_i признака A в новые значения ν'_i, находящиеся в диапазоне $[min'_A, max'_A]$, при этом
   $$\nu'_i = \frac{\nu_i - min_A}{max_A - min_A}\left(max'_A - min'_A\right) + min'_A$$
2. нормализация z-score (стандартизация)
   - При нормализации z-score значения признака A нормализуются на основе среднего значения (μ_A) и стандартного отклонения (σ_A) признака A, а именно:
   $$\nu'_i = \frac{\nu_i - \mu_A}{\sigma_A}$$
3. нормализация при помощи десятичного масштабирования
   - В зависимости от максимального абсолютного значения A значения признака делятся на 10^j:
   $$\nu'_i = \frac{\nu_i}{10^j},$$
   где j – наименьшее целое число, такое, что $\max|\nu'_i| < 1$.

Преобразование данных

Пусть $\mathbf{A} = \{A_1, ..., A_d\}$ – множество признаков набора данных. Новый производный признак Z может быть получен из существующих признаков путем их преобразования.

- линейное преобразование

$$Z = a_1 A_1 + ... + a_d A_d$$

- квадратичное преобразование

$$Z = a_{1,1} A_1^2 + a_{1,2} A_1 A_2 + ... + a_{d-1,d} A_{d-1} A_d + a_{d,d} A_d^2$$

- полиномиальное преобразование

$$Z = f\left(A_1, ..., A_d\right),$$

где f – полином некоторой степени.

Агрегирование данных

Агрегирование данных – это процесс преобразования данных с высокой степенью детализации к более обобщенному представлению, когда значения двух и более объектов комбинируются в один объект.

Целями агрегирования являются:

1. уменьшение размера обрабатываемых данных
2. изменение (укрупнение) масштабов анализа
3. улучшение стабильности данных

Семплирование данных

Семплирование (от англ. sample — выборка), или методы управления выборкой данных, – это подход, направленный на:

1. сокращение объема данных для анализа данных и масштабирования алгоритмов для приложений с большими данными
2. количественную оценку неопределенностей из-за различного распределения данных

Существуют различные методы выборки данных, такие как выборка без замены, когда каждый выбранный экземпляр удаляется из набора данных, и выборка с заменой, где каждый выбранный экземпляр не удаляется, что позволяет выбирать его более одного раза.

Категориальные признаки – кодирование меток

Наличие признаков с текстовыми значениями может мешать применению некоторых методов машинного обучения (например, SVM и ANN).

Категориальные признаки содержат значения в текстовом формате. Примерами являются цвета (“Red”, “Green”, “Yellow”, “Blue”), размеры (“Small”, “Medium”, “Large”, “Extra Large”), географические обозначения (страны, города и т.п.). Некоторые алгоритмы машинного обучения требуют преобразования текстовых значений в числовые для дальнейшей обработки.

Кодирование меток (label encoding) состоит в преобразовании столбца в категорию и использовании значений категории.

Категориальные признаки – прямое кодирование

Прямое кодирование (One Hot Encoding) состоит в том, чтобы конвертировать каждую категорию в новый столбец, принимающий значения 1 или 0 (True/False). Преимуществом этого подхода является то, что между категориальными значениями не устанавливаются несуществующие связи, а недостатком – что в наборе данных появляются дополнительные столбцы.

Прямое кодирование может приводить к резкому увеличению числа столбцов в наборе, если категориальные признаки имеют большое число различных значений.

Избыточные признаки

Признак является избыточным (redundant), если он может быть производным от другого признака или набора признаков. Избыточность признаков приводит к увеличению размера набора данных, а это означает, что время работы алгоритмов также увеличивается, а также может вызвать переобучение в полученной модели.

Избыточность признаков можно обнаружить с помощью корреляционного анализа, который измеряет, насколько сильно влияние одного признака на другой.

Когда данные являются категориальными и набор значений, таким образом, конечен, обычно применяется критерий χ^2 (хи-квадрат). Для числовых признаков типично использование коэффициента корреляции и ковариации.

Критерий χ^2 для категориальных признаков

Пусть два категориальных признака A и B принимают l и m различных значений $a_1, ..., a_l$ и $b_1, ..., b_m$ соответственно. Построим таблицу сопряженности с совместными событиями (A_i, B_j), в которых признак A принимает значение a_i, а признак B принимает значение b_j. Значение χ^2 (или статистика Пирсона χ^2) вычисляется по формуле:

$$\chi^2 = \sum_{i=1}^{l} \sum_{j=1}^{m} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

где o_{ij} – наблюдаемая частота совместного события (A_i, B_j), а e_{ij} – ожидаемая частота (A_i, B_j), равная

$$e_{ij} = \frac{count\,(A = a_i)\, count\,(B = b_j)}{n},$$

где n – количество сущностей (записей) в наборе данных, $count\,(A = a_i)$ – число сущностей с признаком A, равным a_i, $count\,(B = b_j)$ – число сущностей с признаком B, равным b_j.

Избыточность категориальных признаков

Критерий (тест) χ^2 для категориальных признаков A и B проверяет (нулевую) гипотезу, что признаки A и B независимы (тогда расхождения между наблюдаемыми и ожидаемыми частотами несущественны). Альтернативная гипотеза состоит в том, что признаки A и B зависимы. Цель состоит в том, чтобы определить, какая гипотеза лучше подтверждается имеющимся набором данных.

Статистика χ^2 для набора данных $\mathbf{D}$ имеет распределение χ^2 с $(l-1)(m-1)$ степенями свободы, поэтому вычисленное значение сравнивается с таблицей значений функции распределения χ^2 с соответствующим числом степеней свободы или значением функции распределения χ^2, полученного при помощи программного обеспечения, для соответствующего уровня значимости. Если вычисленное значение χ^2 выше необходимого значения согласно таблице, можно сказать, что нулевая гипотеза отклонена и, следовательно, признаки A и B не являются независимыми и один из них может быть удален из набора данных. В противном случае оснований отвергать гипотезу о независимости признаков нет.

Избыточность числовых признаков

Для проверки зависимости (корреляции) числовых признаков A и B можно вычислить коэффициент корреляции (Пирсона):

$$\rho_{A,B} = \frac{1}{n\,\sigma_A \sigma_B} \sum_{i=1}^{n} \left(a_i - \bar{A}\right)\left(b_i - \bar{B}\right) = \frac{1}{n\,\sigma_A \sigma_B} \left(\sum_{i=1}^{n} a_i b_i - m\bar{A}\bar{B} \right),$$

где n – количество записей, a_i и b_i - значения признаков A и B в i-й записи, $\bar{A}$ и $\bar{B}$ – средние значения A и B соответственно, а σ_A и σ_B – стандартные отклонения A и B.

Для коэффициента корреляции $-1 \leqslant \rho_{A,B} \leqslant +1$. Когда $\rho_{A,B} > 0$, это означает, что два признака A и B положительно коррелированы: когда значения A увеличиваются, значения B также увеличиваются. Чем выше коэффициент $\rho_{A,B}$, тем выше корреляция между признаками. Высокое значение $\rho_{A,B}$ может указывать на то, что один из двух признаков A и B является избыточным и может быть удален.

Отбор признаков

Набор признаков, используемых для обучения модели, оказывает значительное влияние на качество результатов. Присутствие в наборе данных малоинформативных признаков приводит к снижению точности многих моделей, особенно моделей регрессии.

Отбор признаков (feature selection) – это процесс выбора признаков, обеспечивающий более высокое качество модели машинного обучения или оптимальный набор признаков в соответствии с определенным критерием.

Отбор признаков перед построением модели обеспечивает следующие преимущества:

- Уменьшение переобучения – чем меньше избыточных данных, тем меньше возможностей для модели принимать решения на основе «шума» в данных.
- Повышение точности – чем меньше противоречивых данных, тем выше точность.
- Сокращение времени обучения – чем меньше данных, тем быстрее обучается модель.

Поиск набора признаков (алгоритм SFG)

Последовательная прямая генерация (Sequential Forward Generation, SFG): начинается с пустого списка признаков S. Признаки добавляются в S в соответствии с критерием U, который отличает лучший признак от других. Список S растет, пока не достигнет полного набора исходных признаков. Критерием остановки может быть пороговое значение для количества признаков или просто генерациия всех возможных подмножеств полным перебором.

Algorithm 1 Sequential forward feature set generation - SFG.

function SFG(F - full set, U - measure)
  initialize: $S = \{\}$ ▷ S stores the selected features
  repeat
    $f = \text{FINDNEXT}(F)$
    $S = S \cup \{f\}$
    $F = F - \{f\}$
  until S satisfies U *or* $F = \{\}$
  return S
end function

Поиск набора признаков (алгоритм SBG)

Последовательная обратная генерация (Sequential Backward Generation, SBG): начинается с полного набора признаков и, итеративно, признаки удаляются по одному. Здесь критерий U должен указывать на худший или наименее важный признак. В конце набор признаков состоит только из одного признака, который считается наиболее информативным из всего набора. Как и в предыдущем случае, могут использоваться другие критерии остановки.

Algorithm 2 Sequential backward feature set generation - SBG.

function SBG(F - full set, U - measure)
  initialize: $S = \{\}$ ▷ S holds the removed features
  repeat
    $f = \text{GETNEXT}(F)$
    $F = F - \{f\}$
    $S = S \cup \{f\}$
  until S does not satisfy U *or* $F = \{\}$
  return $F \cup \{f\}$
end function

Поиск набора признаков (алгоритм BG)

Двунаправленная генерация (Bidirectional Generation, BG): начинает поиск в обоих направлениях, одновременно выполняя алгоритмы SFG и SBG. Алгоритмы останавливаются в двух случаях: (1) когда один алгоритм находит лучший набор, состоящий из m признаков, прежде чем он достигает точной середины, или (2) оба алгоритма достигают середины пространства поиска. Алгоритм использует преимущества как SFG, так и SBG.

Algorithm 3 Bidirectional feature set generation - BG.

function BG(F_f, F_b - full set, U - measure)
  initialize: $S_f = \{\}$ ▷ S_f holds the selected features
  initialize: $S_b = \{\}$ ▷ S_b holds the removed features
  repeat
    f_f = FINDNEXT(F_f)
    f_b = GETNEXT(F_b)
    $S_f = S_f \cup \{f_f\}$
    $F_b = F_b - \{f_b\}$
    $F_f = F_f - \{f_f\}$
    $S_b = S_b \cup \{f_b\}$
  until (a) S_f satisfies U *or* $F_f = \{\}$ *or* (b) S_b does not satisfy U *or* $F_b = \{\}$
  return S_f if (a) *or* $F_b \cup \{f_b\}$ if (b)
end function

Поиск набора признаков (алгоритм RG)

Случайная генерация (Random Generation, RG): запускает случайный поиск. Выбор добавления или удаления признаков является случайным решением. Алгоритм RG пытается избежать застревания в точках локального минимума, не следуя фиксированному пути генерации подмножеств. В отличие от SFG или SBG, размер подмножества функций не может быть оговорен.

Algorithm 4 Random feature set generation - RG.

function RG(F - full set, U - measure)
  initialize: $S = S_{best} = \{\}$ ▷ S - subset set
  initialize: $C_{best} = \#(F)$ ▷ # - cardinality of a set
  repeat
    $S = \text{RANDGEN}(F)$
    $C = \#(S)$
    if $C \leq C_{best}$ *and* S satisfies U **then**
      $S_{best} = S$
      $C_{best} = C$
    end if
  until some stopping criterion is satisfied
  return S_{best} ▷ Best set found so far
end function

Критерий выбора – информационная мера

Дана функцию неопределенности U и априорные вероятности классов $\mathbb{P}[c_i]$, где $i = 1, ..., C$, C – количество классов. Информационный выигрыш $IG(A)$ признака A определяется как разница между априорной неопределенностью и ожидаемой апостериорной неопределенностью с использованием A:

$$IG(A) = \sum_{i=1}^{C} U(\mathbb{P}[c_i]) - \sum_{i=1}^{C} U(\mathbb{P}[c_i \mid A])$$

По теореме Байеса

$$\mathbb{P}[c_i \mid A] = \frac{\mathbb{P}[c_i]\,\mathbb{P}[A \mid c_i]}{\mathbb{P}[A]},\ \mathbb{P}[A] = \sum_{i=1}^{C} \mathbb{P}[c_i]\,\mathbb{P}[A \mid c_i]$$

Согласно модели оценки признаков, основанной на концепции информационного выигрыша, если $IG(A_i) > IG(A_j)$, то признак A_i выбирается вместо признака A_j, так как A_i уменьшает больше неопределенности, чем A_j.

Критерий выбора – энтропия Шеннона

Так как слагаемое $\sum_{i=1}^{C} U\left(\mathbb{P}\left[c_i\right]\right)$ не зависит от выбора признака, можно переписать правило выбора так: признак A_i выбирается вместо признака A_j, если

$$U'\left(A_i\right) < U'\left(A_j\right), U'\left(A\right) = \sum_{i=1}^{C} U\left(\mathbb{P}\left[c_i \mid A\right]\right)$$

Наиболее часто используемой функцией неопределенности является функция

$$U\left(x\right) = -x \log_2 x,$$

при которой выбор признака основывается на энтропии Шеннона и

$$IG\left(A\right) = -\sum_{i=1}^{C} \mathbb{P}\left[c_i\right] \log_2 \mathbb{P}\left[c_i\right] + \sum_{i=1}^{C} \mathbb{P}\left[c_i \mid A\right] \log_2 \mathbb{P}\left[c_i \mid A\right]$$

Тогда признак A_i выбирается вместо признака A_j, если

$$\sum_{k=1}^{C} \mathbb{P}\left[c_k \mid A_i\right] \log_2 \mathbb{P}\left[c_k \mid A_i\right] < \sum_{k=1}^{C} \mathbb{P}\left[c_k \mid A_j\right] \log_2 \mathbb{P}\left[c_k \mid A_j\right]$$

Критерий выбора – меры расстояния

Наиболее типичная мера определяется расстоянием между функциями условной плотности классов. Например, в задаче с двумя классами, если $D(A)$ – это расстояние между условными плотностями $p(x, A \mid c_1)$ и $p(x, A \mid c_2)$, то правило оценки признаков, основанное на расстоянии $D(A)$, гласит, что выбирается признак A_i вместо A_j, если $D(A_i) > D(A_j)$.

При отборе признаков используются две популярные меры расстояния: направленная дивергенция DD и дисперсия V:

$$DD(A_j) = \int \left(\sum_i p(x, c_i \mid A_j = a) \log \frac{p(x, c_i \mid A_j = a)}{p(x, c_i)} \right) p(x, A_j = a)\, dx$$

$$V(A_j) = \int \left(\sum_i p(x, c_i) \left(p(x, c_i \mid A_j = a) - p(x, c_i) \right)^2 \right) p(x, A_j = a)\, dx$$

Критерий выбора – меры расстояния

Объяснением использования мер расстояния является то, что мы пытаемся найти лучший признак, чтобы разделить два класса как можно лучше. Функции расстояния между априорной и апостериорной вероятностями классов аналогичны подходу на основе информационного выигрыша, за исключением того, что функции основаны на расстояниях, а не на неопределенности. Меры расстояния также известны как меры разделения, дискриминации или расхождения.

Расстояние	Математическая форма
Евклидово расстояние	$D_e = \left(\sum_{i=1}^{d} (x_i - y_i)^2\right)^{\frac{1}{2}}$
Расстояние городских кварталов	$D_{cb} = \sum_{i=1}^{d} \lvert x_i - y_i \rvert$
Расстояние Чебышева	$D_{ch} = \max_i \lvert x_i - y_i \rvert$
Расстояние Минковского	$D_M = \left(\sum_{i=1}^{d} (x_i - y_i)^p\right)^{\frac{1}{p}}$
Квадратичное расстояние	$D_q = \sum_{i,j=1}^{d} (x_i - y_i) Q_{ij} (x_j - y_j)$
Канберрское расстояние	$D_{ca} = \sum_{i=1}^{d} \frac{\lvert x_i - y_i \rvert}{\lvert x_i \rvert + \lvert y_i \rvert}$
Угловое расстояние	$D_{as} = \frac{\sum_{i=1}^{d} x_i\, y_i}{\left(\sum_{i=1}^{d} x_i^2 \sum_{i=1}^{d} y_i^2\right)^{\frac{1}{2}}}$

Критерии выбора – меры зависимости

Меры зависимости также известны как меры ассоциации или корреляции. При оценке признаков общая процедура состоит в измерении корреляции между любым признаком и классом. Обозначая через $R(A)$ меру зависимости между признаком A и классом C, мы выбираем признак A_i вместо признака A_j, если $R(A_i) > R(A_j)$. Другими словами, выбирается признак, наиболее коррелированный с классом. Если A и C статистически независимы, то они не коррелируют, и удаление признака A не должно влиять на разделимость классов при помощи остальных признаков. В противном случае следует оставить признак, потому что он может в некоторой степени объяснить тенденцию в классе.

Одним из наиболее часто используемых показателей зависимости является коэффициент корреляции Пирсона, который измеряет степень линейной корреляции между двумя переменными. Для двух переменных X и Y коэффициент корреляции определяется по формуле ($\bar{x}$ и $\bar{y}$ – средние значения):

$$\rho(X,Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\left(\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2\right)}$$

Одномерное ранжирование признаков

Основная идея состоит в том, чтобы оценить каждый признак с помощью меры и привязать результат оценки к каждому признаку. Затем признаки сортируются по значениям оценки в убывающем порядке. Когда известно наиболее подходящее число признаков, то для выполнения отбора достаточно выбрать нужное число первых отранжированных признаков. Однако не существует простой процедуры для получения количества признаков для отбора.

Algorithm 5 A univariate feature ranking algorithm.

function RANKING ALGORITHM(x - features, U - measure)
  initialize: list $L = \{\}$ ▷ L stores ordered features
  for each feature x_i, $i \in \{1, \ldots, M\}$ **do**
    $v_i =$ COMPUTE(x_i, U)
    position x_i into L according to v_i
  end for
  return L in decreasing order of feature relevance.
end function

Техника минимального набора признаков

Количество соответствующих признаков – это параметр, который часто не известен. Поэтому существуют алгоритмы, ориентированные на получение минимально возможного подмножества признаков без их упорядочивания.

В алгоритме построения минимального набора признаков функция subsetGenerate(), использующая одну из схем генерации подмножеств, возвращает подмножество признаков и устанавливает критерий остановки stop в значение true. Функция legitimacy() возвращает истину, если подмножество признаков S_k удовлетворяет требованиям к значению меры U.

Algorithm 6 A minimum subset algorithm.

function MIN- SET ALGORITHM(x - features, U - measure)
  initialize: $L = \{\}$, stop = *false* ▷ S holds the minimum set
  repeat
    S_k = SUBSETGENERATE(x) ▷ **stop** can be set here
    if LEGITIMACY(S_k, U) is *true and* $\#(S_k) < \#(S)$ **then**
      $S = S_k$ ▷ S is replaced by S_k
    end if
  until stop = *true*
  return S - the minimum subset of features
end function

Снижение размерности

Цель снижения размерности состоит в том, чтобы найти представление матрицы данных $\mathbf{D}$ в виде матрицы более низкой размерности.

Если дана матрица данных размером $n \times d$, то каждая запись (строка) вида $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{id})^T$ является вектором в d-мерном пространстве, образованном стандартными базисными векторами $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_d$.

Если дано другое множество из d ортонормальных векторов $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_d$, то можно выразить каждую точку (вектор) $\mathbf{x}$ в виде

$$\mathbf{x} = a_1\mathbf{u}_1 + a_2\mathbf{u}_2 + ... + a_d\mathbf{u}_d,$$

где вектор $\mathbf{a} = (a_1, a_2, ..., a_d)^T$ представляет собой вектор координат $\mathbf{x}$ в новом базисе. В матричном виде имеем:

$$\mathbf{x} = \mathbf{U}\,\mathbf{a},$$

где $\mathbf{U}$ – ортогональная $d \times d$-матрица, в которой i-й столбец является i-ым базисным вектором $\mathbf{u}_i$. Тогда $\mathbf{U}^{-1} = \mathbf{U}^T$, поэтому

$$\mathbf{a} = \mathbf{U}^T\mathbf{x}$$

Оптимальный базис: проекция в подпространство

Задача состоит в том, чтобы найти оптимальный базис, сохраняющей существенную информацию о матрице данных $\mathbf{D}$, а именно, требуется найти оптимальное r-мерное представление для $\mathbf{D}$, где r значительно меньше d.

Проекция $\mathbf{x}$ на первые r базисных векторов равна

$$\mathbf{x}' = a_1\mathbf{u}_1 + a_2\mathbf{u}_2 + ... + a_r\mathbf{u}_r = \sum_{i=1}^{r} a_i\mathbf{u}_i = \mathbf{U}_r\,\mathbf{a}_r,$$

где $\mathbf{U}_r$ и $\mathbf{a}_r$ представляют собой r базисных векторов и координат соответственно ($\mathbf{U}_r = (\mathbf{u}_1, ..., \mathbf{u}_r)$, $(\mathbf{a}_r = a_1, ..., a_r)$). Ограничиваясь в равенстве $\mathbf{a} = \mathbf{U}^T\mathbf{x}$ первыми r компонентами, получим

$$\mathbf{a}_r = \mathbf{U}_r^T\mathbf{x}$$

Поэтому r-мерная проекция $\mathbf{x}$ задается формулой

$$\mathbf{x}' = \mathbf{U}_r\,\mathbf{U}_r^T\mathbf{x} = \mathbf{P}_r\mathbf{x},$$

где $\mathbf{P}_r = \mathbf{U}_r\,\mathbf{U}_r^T = \sum_{i=1}^{r}\mathbf{u}_i\mathbf{u}_i^T$ – это матрица ортогональной проекции для подпространства, порожденного первыми r базисными векторами.

Оптимальный базис: вектор ошибки

Пусть проекция на первые r базисных векторов равна $\mathbf{x}' = \mathbf{P}_r\mathbf{x}$, тогда соответствующий вектор ошибки $\boldsymbol{\varepsilon}$ – это проекция на оставшиеся $d - r$ базисных векторов

$$\boldsymbol{\varepsilon} = \sum_{i=r+1}^{d} a_i \mathbf{u}_i = \mathbf{x} - \mathbf{x}'$$

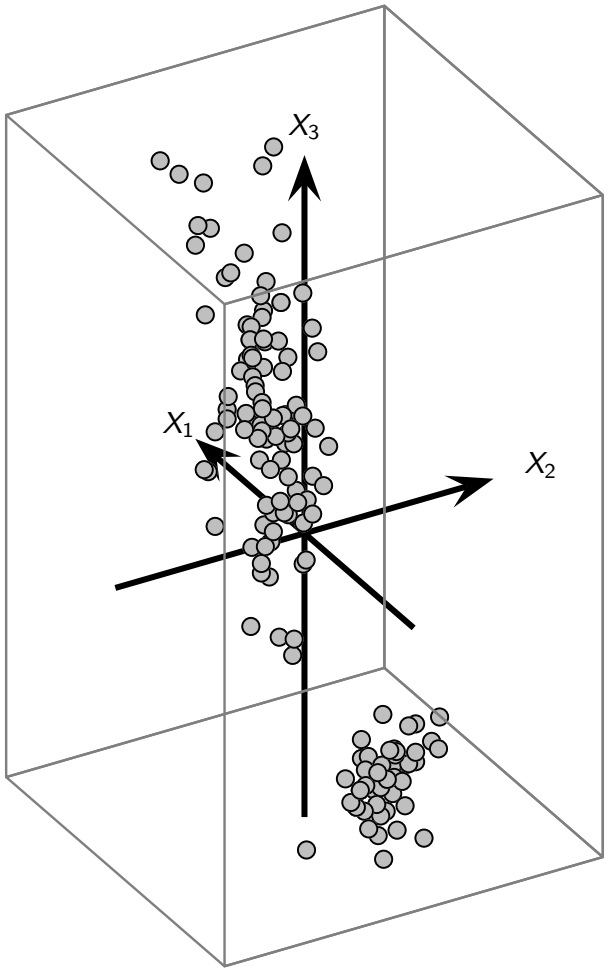
Вектор ошибки $\boldsymbol{\varepsilon}$ ортогонален вектору $\mathbf{x}'$.

Задача состоит в том, чтобы найти оптимальный базис, сохраняющей существенную информацию о матрице данных $\mathbf{D}$, а именно, требуется найти оптимальное r-мерное представление для $\mathbf{D}$, где r значительно меньше d.

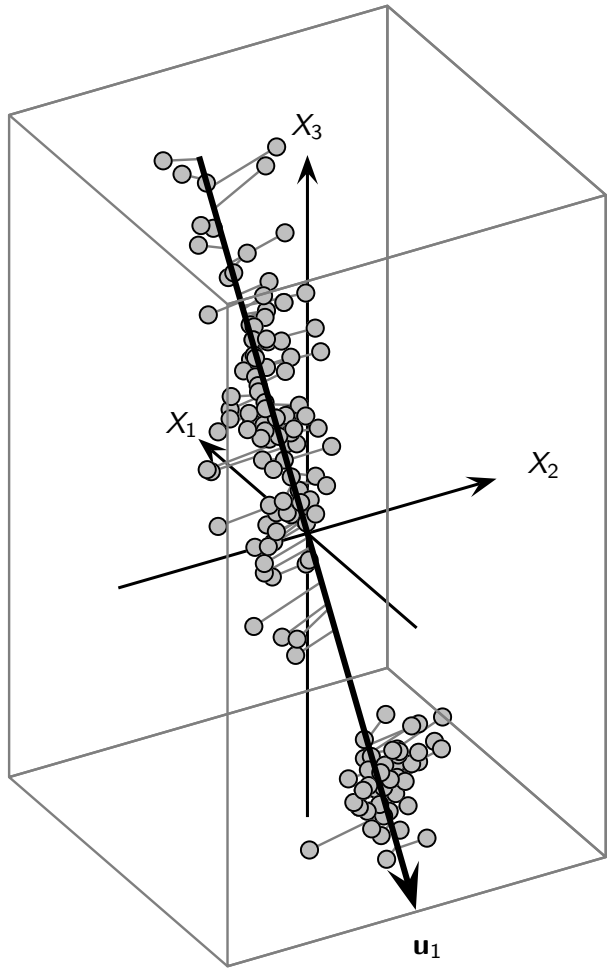
Цель сокращения размерности состоит в том, чтобы найти r-мерный базис, который дает наилучшую аппроксимацию $\mathbf{x}'_i$ для всех точек $\mathbf{x}_i \in \mathbf{D}$.

Другими словами, требуется минимизировать ошибку $\boldsymbol{\varepsilon} = \mathbf{x}_i - \mathbf{x}'_i$ по всем точкам набора данных $\mathbf{D}$.

Набор данных «Ирисы»: оптимальный базис $r = 1$

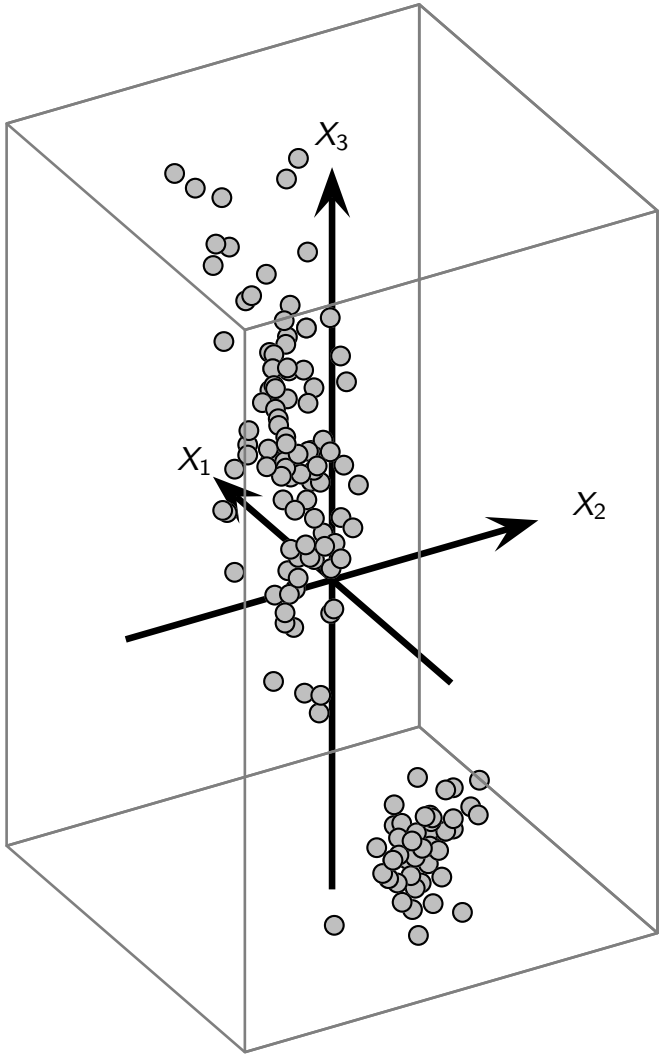


Набор данных "Ирисы": 3D

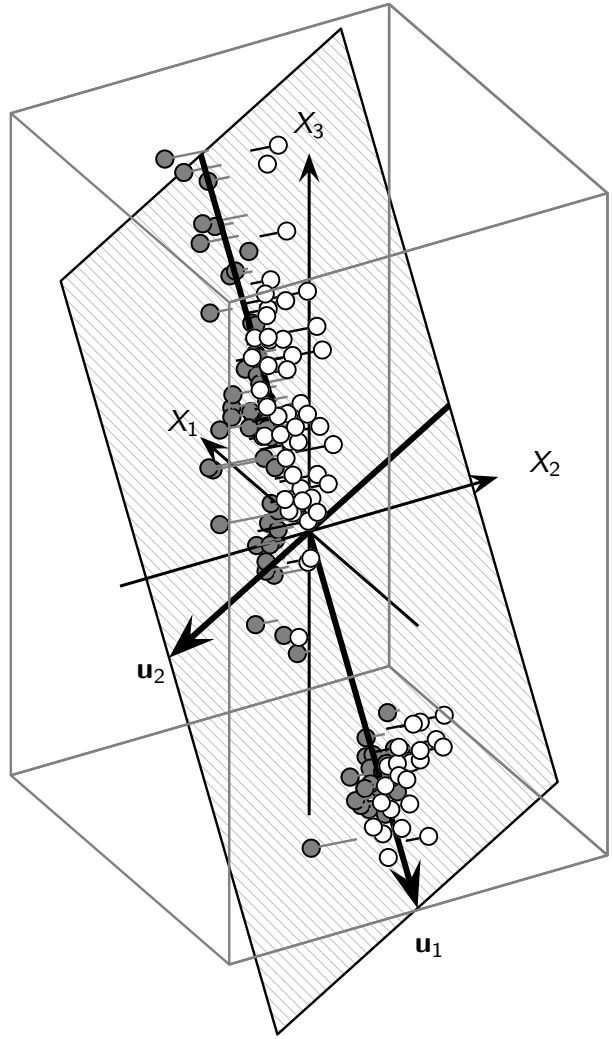


Оптимальный базис: 1D

Набор данных «Ирисы»: оптимальный базис $r = 2$



Набор данных "Ирисы": 3D



Оптимальный базис: 2D

Метод главных компонент (PCA)

Метод главных компонент (Principal Component Analysis, PCA) – это метод, позволяющий найти r-мерный базис, который наилучшим образом отражает дисперсию в данных.

Направление с максимальной дисперсией называется первой главной компонентой.

Направление с максимальной дисперсией при условии ортогональности направлению первой главной компоненты называется второй главной компонентой.

И т.д.: направление с максимальной дисперсией при условии ортогональности направлениям первых $k-1$ главных компонент называется k-й главной компонентой.

Направление, которое максимизирует дисперсию, одновременно является направлением, которое минимизирует среднеквадратичную ошибку.

Будем считать, что матрица данных $\mathbf{D}$ уже центрирована, и пусть $\mathbf{\Sigma}$ – это ее ковариационная матрица.

PCA: направление с максимальной дисперсией (1)

Требуется найти единичный вектор $\mathbf{u}$, который максимизирует дисперсию проекций точек набора данных.

Проекция $\mathbf{x}_i$ на $\mathbf{u}$ равна

$$\mathbf{x}'_i = \left(\frac{\mathbf{u}^T \mathbf{x}_i}{\mathbf{u}^T \mathbf{u}} \right) \mathbf{u} = \left(\mathbf{u}^T \mathbf{x}_i \right) \mathbf{u} = a_i \mathbf{u}$$

По всем точкам набора данных дисперсия проекций на $\mathbf{u}$ равна

$$\sigma^2_{\mathbf{u}} = \frac{1}{n} \sum_{i=1}^{n} \left(a_i - \mu_{\mathbf{u}} \right)^2 = \frac{1}{n} \sum_{i=1}^{n} \mathbf{u}^T \left(\mathbf{x}_i \mathbf{x}_i^T \right) \mathbf{u} = \mathbf{u}^T \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \right) \mathbf{u} = \mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u}$$

Требуется найти оптимальный базисный вектор $\mathbf{u}$, который максимизирует дисперсию проекций $\sigma^2_{\mathbf{u}} = \mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u}$ при условии, что $\mathbf{u}^T \mathbf{u} = 1$. Максимизируемая целевая функция равна

$$J\left(\mathbf{u}\right) = \mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u} - \alpha \left(\mathbf{u}^T \mathbf{u} - 1 \right)$$

PCA: направление с максимальной дисперсией (2)

Если дана целевая функция $J(\mathbf{u}) = \mathbf{u}^T\mathbf{\Sigma}\mathbf{u} - \alpha\left(\mathbf{u}^T\mathbf{u} - 1\right)$, то решаем задачу максимизации $J(\mathbf{u})$, вычисляя производную (градиент) $J(\mathbf{u})$ по $\mathbf{u}$ и приравнивая ее нулю:

$$\frac{\partial}{\partial \mathbf{u}}\left(\mathbf{u}^T\mathbf{\Sigma}\mathbf{u} - \alpha\left(\mathbf{u}^T\mathbf{u} - 1\right)\right) = \mathbf{0} \Rightarrow 2\mathbf{\Sigma}\mathbf{u} - 2\alpha\mathbf{u} = \mathbf{0} \Rightarrow \mathbf{\Sigma}\mathbf{u} = \alpha\mathbf{u}$$

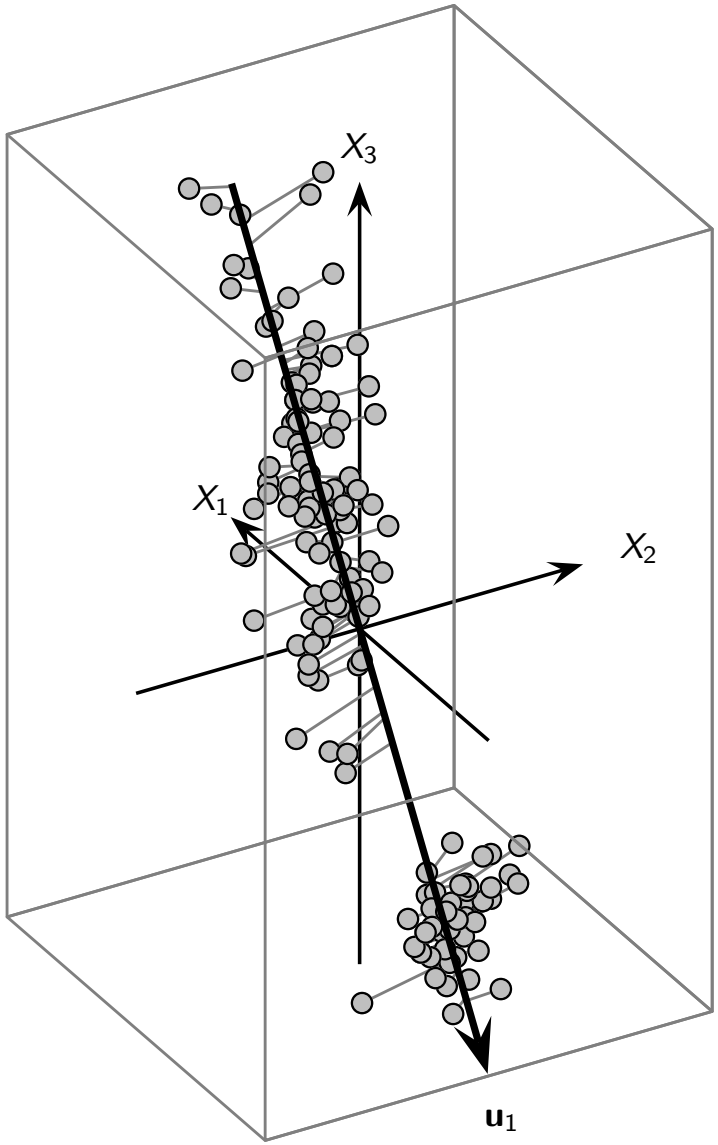
Таким образом, α – это собственное значение ковариационной матрицы $\mathbf{\Sigma}$ с собственным вектором $\mathbf{u}$.

Так как

$$\sigma^2_{\mathbf{u}} = \mathbf{u}^T\mathbf{\Sigma}\mathbf{u} = \mathbf{u}^T\alpha\mathbf{u} = \alpha\mathbf{u}^T\mathbf{u} = \alpha$$

Чтобы максимизировать дисперсию проекций $\sigma^2_{\mathbf{u}}$, следует выбрать в качестве α наибольшее собственное значение λ_1 матрицы $\mathbf{\Sigma}$, тогда собственный вектор $\mathbf{u}_1$ задает направление с максимальной дисперсией проекций, называемое первой главной компонентой.

Набор «Ирисы»: первая главная компонента



Минимальная среднеквадратичная ошибка (MSE)

Направление, которое максимизирует дисперсию проекций, также минимизирует среднюю квадратичную ошибку.

Среднеквадратичная ошибка (mean squared error, MSE) равна

$$MSE\left(\mathbf{u}\right) = \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{\varepsilon}_i\right\|^2 = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i - \mathbf{x}_i'\right\|^2 = \sum_{i=1}^{n}\frac{\left\|\mathbf{x}_i\right\|^2}{n} - \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u}$$

Так как первое слагаемое фиксировано для набора данных $\mathbf{D}$, очевидно, что направление $\mathbf{u}_1$, которое максимизирует дисперсию, также минимизирует MSE.

Далее имеем

$$\sum_{i=1}^{n}\frac{\left\|\mathbf{x}_i\right\|^2}{n} = \mathrm{var}\left(\mathbf{D}\right) = \mathrm{tr}\left(\boldsymbol{\Sigma}\right) = \sum_{i=1}^{d}\sigma_i^2$$

Поэтому для направления $\mathbf{u}_1$, которое минимизирует MSE, имеем

$$MSE\left(\mathbf{u}_1\right) = \mathrm{var}\left(\mathbf{D}\right) - \mathbf{u}_1^T\boldsymbol{\Sigma}\mathbf{u}_1 = \mathrm{var}\left(\mathbf{D}\right) - \lambda_1$$

Наилучшая двумерная аппроксимация

Двумерное подпространство, которое наилучшим образом отражает дисперсию в $\mathbf{D}$, образовано собственными векторами $\mathbf{u}_1$ и $\mathbf{u}_2$, соответствующими двум наибольшим собственным числам λ_1 и λ_2 соответственно.

Пусть $\mathbf{U}_2 = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 \end{pmatrix}$ – это матрица, которая соответствует двум главным компонентам. Тогда координаты проекции точки $\mathbf{x}_i \in \mathbf{D}$ вычисляются следующим образом:

$$\mathbf{a}_i = \mathbf{U}_2^T \mathbf{x}_i$$

Пусть $\mathbf{A}$ обозначает двумерную проекцию точек $\mathbf{D}$. Общая дисперсия точек в проекции $\mathbf{A}$ равна

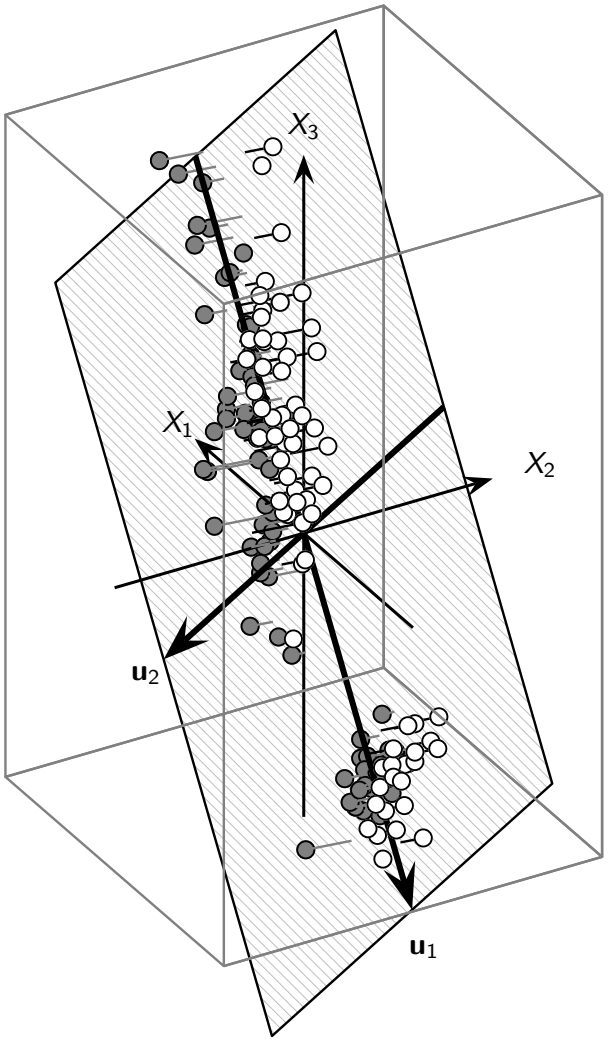
$$\text{var}\,(\mathbf{A}) = \mathbf{u}_1^T \mathbf{\Sigma} \mathbf{u}_1 + \mathbf{u}_2^T \mathbf{\Sigma} \mathbf{u}_2 = \mathbf{u}_1^T \lambda_1 \mathbf{u}_1 + \mathbf{u}_2^T \lambda_2 \mathbf{u}_2 = \lambda_1 + \lambda_2$$

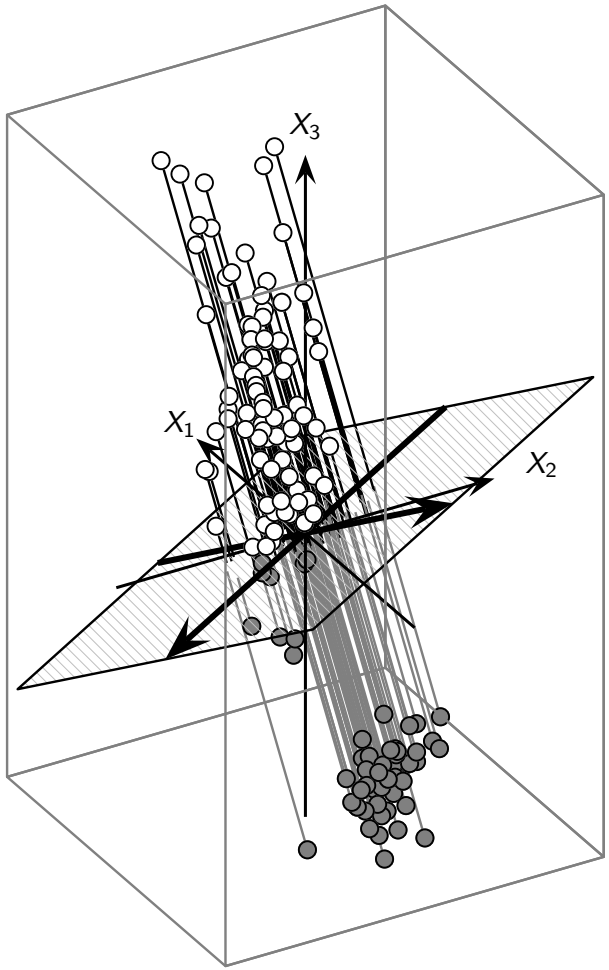
Первые две главные компоненты также минимизируют целевую функцию среднеквадратичной ошибки

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \mathbf{x}_i' \right\|^2 = \text{var}\,(\mathbf{D}) - \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^T \mathbf{P}_2 \mathbf{x}_i = \text{var}\,(\mathbf{D}) - \text{var}\,(\mathbf{A})$$

Оптимальная двумерная аппроксимация

Оптимальное подпространство максимизирует дисперсию и минимизирует квадратичную ошибку.





Наилучшая r-мерная аппроксимация

Для построения наилучшей r-мерной аппроксимации $\mathbf{D}$ вычислим собственные значения ковариационной матрицы $\mathbf{\Sigma}$. Так как $\mathbf{\Sigma}$ положительно полуопределена ($\mathbf{x}^T\mathbf{\Sigma}\,\mathbf{x} \geqslant 0\ \forall \mathbf{x} \in \mathbb{R}^d$), все ее собственные числа неотрицательные: $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_r \geqslant \lambda_{r+1} \geqslant \ldots \geqslant \lambda_d \geqslant 0$.

Выбираем r наибольших собственных значений и соответствующие им собственные вектора для построения наилучшей r-мерной аппроксимации.

Общая дисперсия: пусть $\mathbf{U}_r = \begin{pmatrix} \mathbf{u}_1 & \ldots & \mathbf{u}_r \end{pmatrix}$ – матрица из r базисных векторов с проекционной матрицей $\mathbf{P}_r = \mathbf{U}_r\mathbf{U}_r^T = \sum_{i=1}^r \mathbf{u}_i\mathbf{u}_i^T$.

Пусть $\mathbf{A}$ обозначает набор данных, сформированный из проекций точек $\mathbf{D}$ на r-мерное подпространство. Дисперсия проекции $\mathbf{A}$ равна

$$\operatorname{var}(\mathbf{A}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^T\mathbf{P}_r\,\mathbf{x}_i = \sum_{i=1}^{r} \mathbf{u}_i^T\mathbf{\Sigma}\,\mathbf{u}_i = \sum_{i=1}^{r} \lambda_i$$

Среднеквадратичная ошибка:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{x}_i'\|^2 = \operatorname{var}(\mathbf{D}) - \sum_{i=1}^{r}\lambda_i = \sum_{i=1}^{d}\lambda_i - \sum_{i=1}^{r}\lambda_i = \sum_{i=r+1}^{d}\lambda_i$$

Выбор размерности аппроксимации

Одним из критериев выбора размерности r является доля общей дисперсии, соответствующая первым r главным компонентам и вычисляемая по формуле

$$f(r) = \frac{\lambda_1 + \lambda_2 + ... + \lambda_r}{\lambda_1 + \lambda_2 + ... + \lambda_d} = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} = \frac{\sum_{i=1}^{r} \lambda_i}{\text{var}(\mathbf{D})}$$

Если задан определенный желаемый уровень доли дисперсии α, то начинаем с первой главной компоненты, продолжаем добавлять дополнительные главные компоненты и останавливаемся на наименьшем r, для которого $f(r) \geqslant \alpha$.

Другими словами, выбираем наименьшую размерность r, для которой подпространство главных компонент этой размерности имеет дисперсию проекции набора данных, составляющую по меньшей мере долю α общей дисперсии (например, $\alpha = 0.9$).

Алгоритм PCA (метода главных компонент)

Algorithm 1: Алгоритм PCA

Data: $\mathbf{D}$ – исходный набор данных, α – доля общей дисперсии

Result: $\mathbf{A}$ – набор данных уменьшенной размерности

1 $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ // вычислить средние значения

2 $\mathbf{Z} = \mathbf{D} - \mathbf{1}\cdot\boldsymbol{\mu}^T$ // центрировать данные

3 $\boldsymbol{\Sigma} = \frac{1}{n}\left(\mathbf{Z}^T\mathbf{Z}\right)$ // вычислить матрицу ковариации

4 $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_d)$ // вычислить собственные значения $\boldsymbol{\Sigma}$

5 $\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & ... & \mathbf{u}_d \end{pmatrix}$ // вычислить собственные векторы $\boldsymbol{\Sigma}$

6 $f(r) = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$, $r = 1, 2, ..., d$ // вычислить доли общей дисперсии

7 Выбрать наименьшее r, такое, что $f(r) \geqslant \alpha$ // выбрать размерность

8 $\mathbf{U}_r = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & ... & \mathbf{u}_r \end{pmatrix}$ // сокращенный базис

9 $\mathbf{A} = \left\{\mathbf{a}_i \mid \mathbf{a}_i = \mathbf{U}_r^T\mathbf{x}_i,\ i = \overline{1,n}\right\}$ // уменьшенная размерность данных

Главные компоненты набора данных «Ирисы»

Матрица ковариации (для первых 3 признаков):

$$\mathbf{\Sigma} = \begin{pmatrix} 0.681 & -0.039 & 1.265 \\ -0.039 & 0.187 & -0.320 \\ 1.265 & -0.320 & 3.092 \end{pmatrix}$$

Собственные значения и собственные вектора $\mathbf{\Sigma}$

$$\lambda_1 = 3.662,\ \lambda_2 = 0.239,\ \lambda_3 = 0.059$$

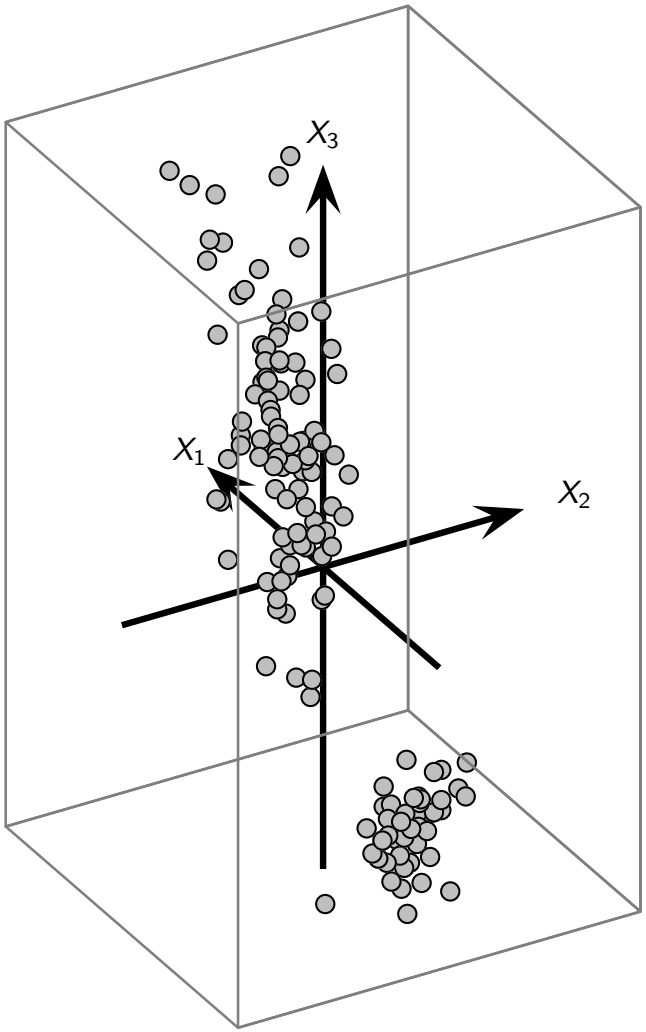
$$\mathbf{u}_1 = \begin{pmatrix} -0.390 \\ 0.089 \\ -0.916 \end{pmatrix},\ \mathbf{u}_2 = \begin{pmatrix} -0.639 \\ -0.742 \\ 0.200 \end{pmatrix},\ \mathbf{u}_3 = \begin{pmatrix} -0.663 \\ 0.664 \\ 0.346 \end{pmatrix}$$

Общая дисперсия равна $\lambda_1 + \lambda_2 + \lambda_3 = 3.662 + 0.239 + 0.059 = 3.96$. Доля общей дисперсии для различных значений r равна

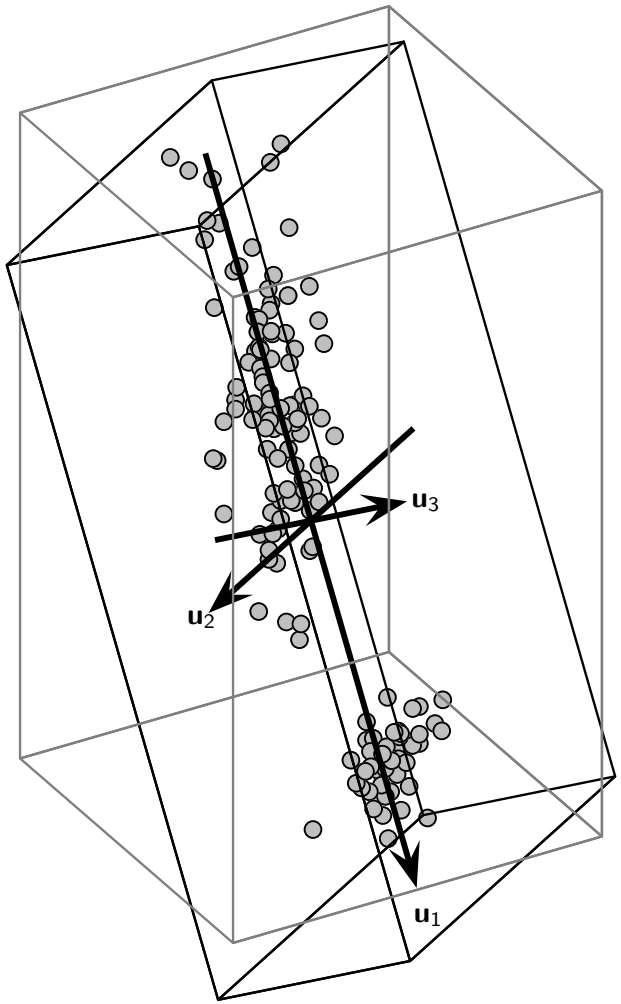
r	1	2	3
$f(r)$	0.925	0.985	1.0

Достаточно двух главных компонент ($r = 2$), чтобы охватить долю дисперсии $\alpha = 0.95$

Оптимальный 3D базис из главных компонент

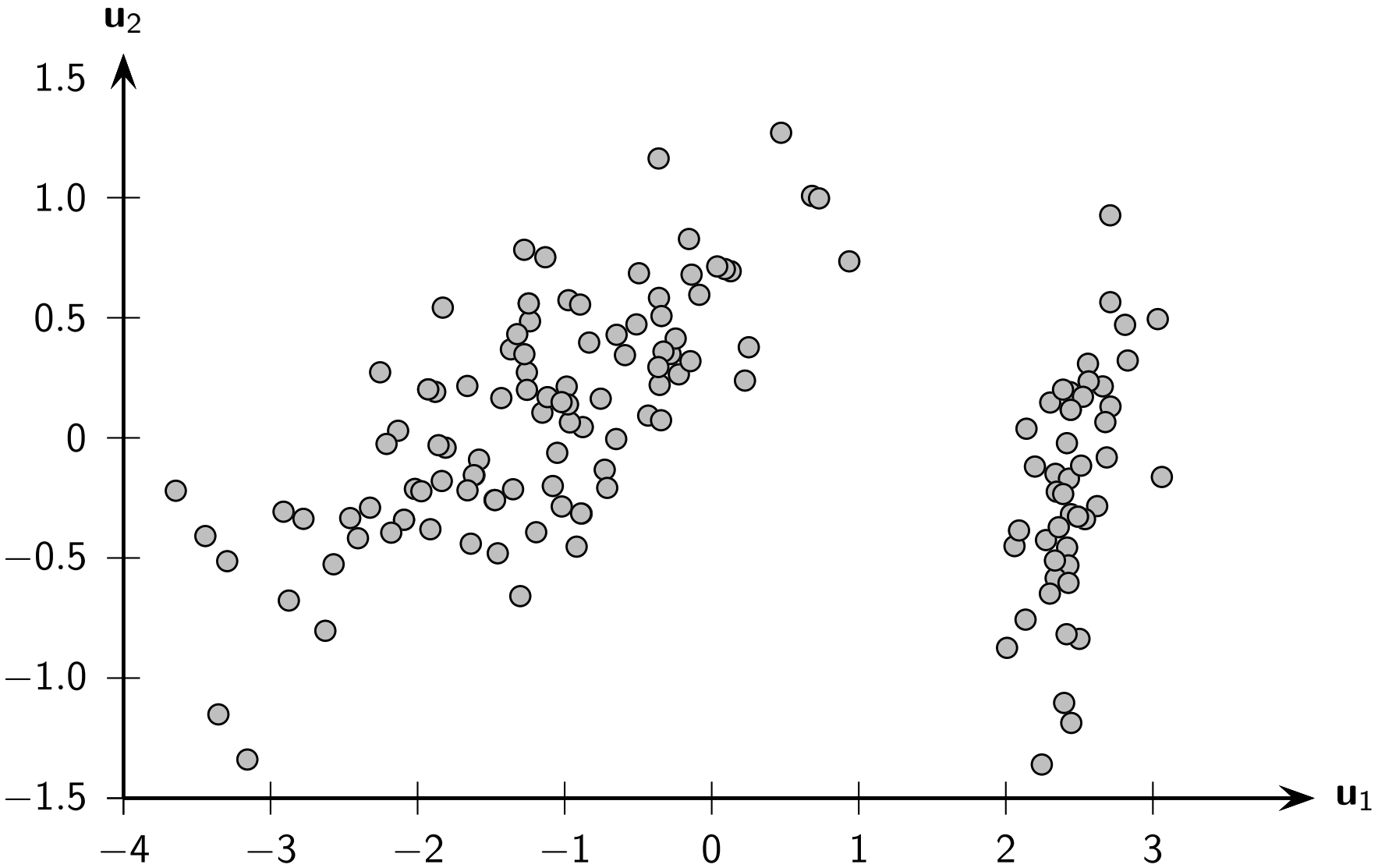


Набор данных "Ирисы": 3D



Оптимальный базис: 3D

Главные компоненты: 2D проекция



Геометрия метода главных компонент

Геометрически, когда $r = d$, метод главных компонент соответствует ортогональной замене базиса, такой, что общая дисперсия равна сумме дисперсия вдоль каждого направления $\mathbf{u}_1, ..., \mathbf{u}_d$ и все ковариации нулевые.
Пусть $\mathbf{U}$ – ортогональная матрица размера $d \times d$, $\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & ... & \mathbf{u}_d \end{pmatrix}$, причем $\mathbf{U}^{-1} = \mathbf{U}^T$. Пусть $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, ..., \lambda_d)$ – диагональная матрица собственных значений. Каждая главная компонента $\mathbf{u}_i$ соответствует собственному вектору ковариационной матрицы $\mathbf{\Sigma}$

$$\mathbf{\Sigma}\,\mathbf{u}_i = \lambda_i \mathbf{u}_i,\ i = \overline{1, d}$$

или в компактных матричных обозначениях

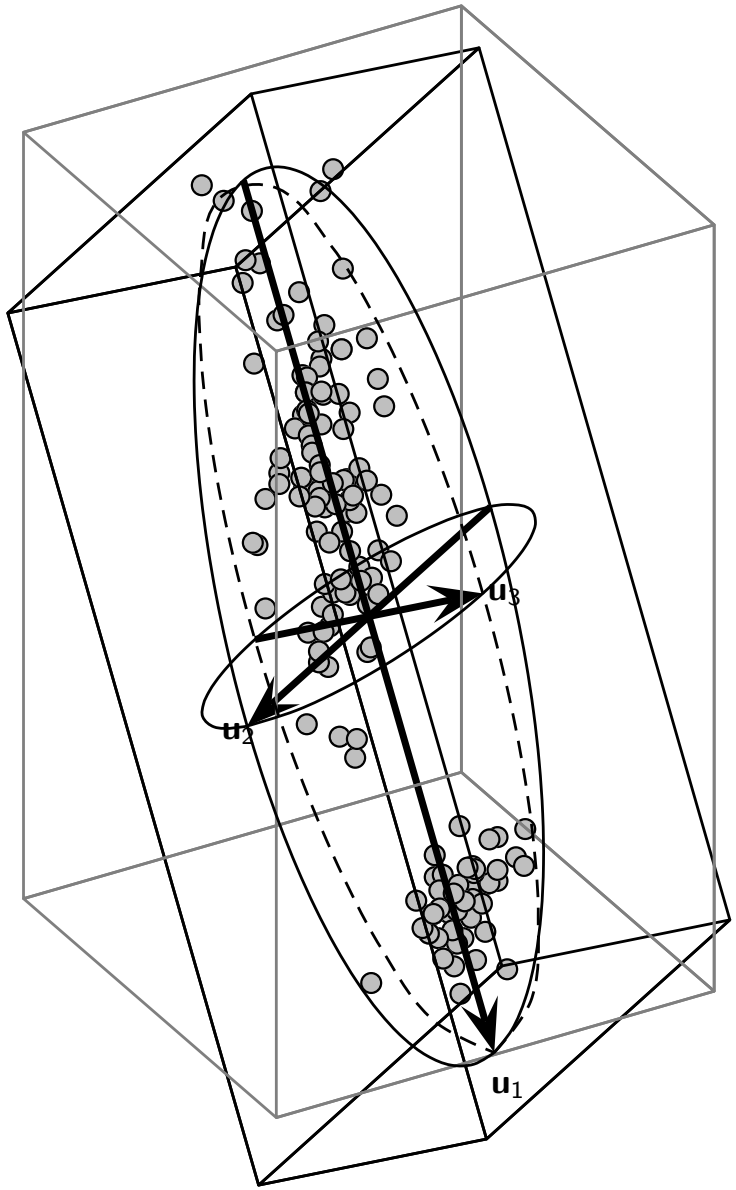
$$\mathbf{\Sigma}\,\mathbf{U} = \mathbf{U}\,\mathbf{\Lambda} \Rightarrow \mathbf{\Sigma} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T$$

Поэтому $\mathbf{\Lambda}$ является ковариационной матрицей в новом базисе из главных компонент. В новом базисе из главных компонент уравнение

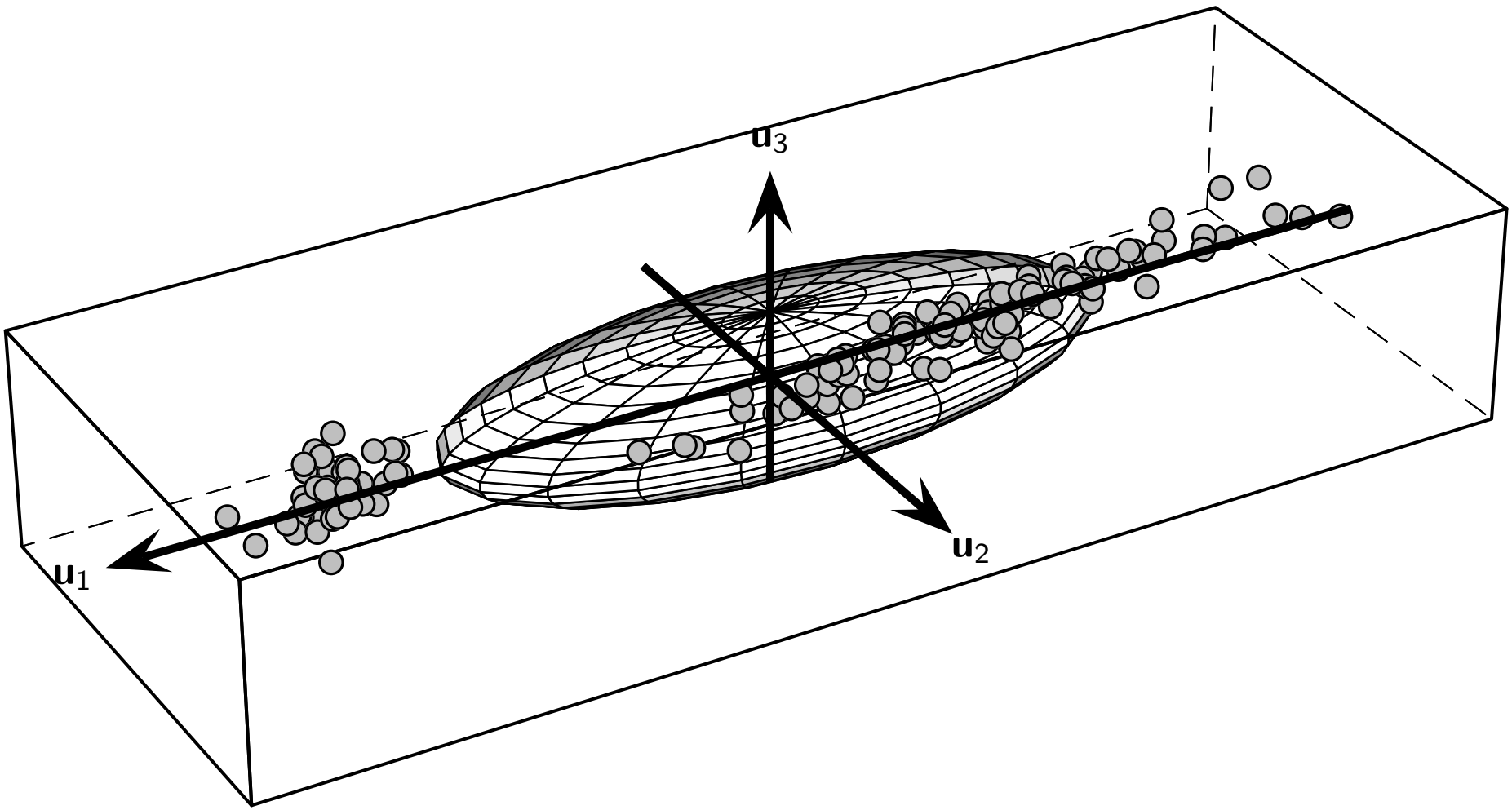
$$\mathbf{x}^T \mathbf{\Sigma}^{-1} \mathbf{x} = 1$$

определяет d-мерный эллипсоид. Собственные вектора $\mathbf{u}_i$ матрицы $\mathbf{\Sigma}$, т.е. главные компоненты, являются направлениями осей эллипсоида. Квадратные корни собственных значений $\sqrt{\lambda_i}$ дают длины полуосей эллипсоида.

Контуры эллипсоида в стандартном базисе



Контуры эллипсоида в базисе PCA



Применение PCA к графическим данным (1)

Рассмотрим в качестве примера применение метода PCA к тестовому изображению Lenna (TIFF 512x512 24bit RGB color).

TIFF (Tagged Image File Format) – это формат хранения растровых графических изображений.

При использовании формата TIFF без сжатия появляются графические файлы большого размера.

Применение PCA к графическим данным (2)

Применяя к набору данных тестового изображения метод PCA и оставляя только первую главную компоненту, а именно, используя представление:

$$\begin{bmatrix} R_{i,j} \\ G_{i,j} \\ B_{i,j} \end{bmatrix} = \begin{bmatrix} 0.767785 \\ 0.45439 \\ 0.4517034 \end{bmatrix} Y_{i,j},$$

где $Y_{i,j}$ – значения первой главной компоненты, соответствующие пикселю (i, j), получим следующее изображение