数据挖掘与应用

数据挖掘简介

授课教师：周晟

浙江大学　软件学院

2021.09

课程内容

1. 课程简介
   - 课程信息
   - 准备知识
2. 数据挖掘简介
   - 从数据到知识
   - 数据挖掘的定义
   - 数据挖掘的应用
   - 数据挖掘与学术
   - 课程大纲
3. 认识数据
   - 数据类型
   - 数据的相关性度量
   - 数据的统计与可视化
   - 数据质量
4. 深入理解主成分分析 PCA
   - 线性代数基础
   - 主成分分析优化目标
   - 主成分分析实现

课程简介

课程信息

随着信息技术的快速发展，人类生产生活中产生了海量的数据。分析和挖掘数据中的潜在模式以及客观规律，对于提升生活质量，生产效率具有重要的意义。本课程主要讲述数据挖掘相关的基本概念，经典任务，前沿技术以及在实际生产生活中的应用。学习本课程有望在掌握数据挖掘相关知识的同时也培养相关的实践动手能力。

1. 课程主页：https://zhoushengisnoob.github.io/courses/
2. 授课时间：秋学期周四
3. 授课教师：周晟
4. 课程助教：郑卓男，胡伟
5. 考核方式：随堂测试 (3*10%) + 期末报告 (70%)

课程准备

数学基础

1. 线性代数基础（常见的矩阵理论）
2. 概率论基础（贝叶斯理论）
3. 统计学基础（常见的统计方法与指标）

代码基础

1. Python
2. Numpy
3. Pandas
4. Scikit-Learn
5. Pytorch

数据的来源与种类

人类从诞生之日起就不断地产生数据，例如文字，图像以及音乐等

数据的来源与种类

人类从诞生之日起就不断地产生数据，例如文字，图像以及音乐等

随着信息技术的快速发展，人类采集和存储了海量的数据，包括商业，社会，科学，工程等领域

常见的数据种类

目前常见的数据类型主要包括：

1. 数据库数据
2. 图像数据
3. 音频数据
4. 视频数据
5. 网络数据
6. ...

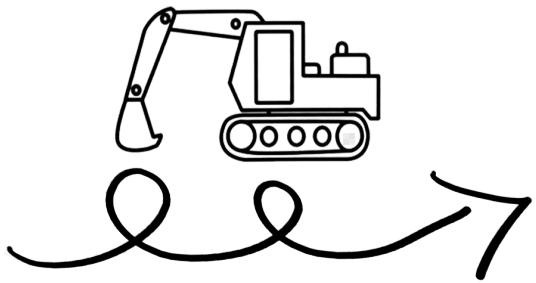
过量信息无法产生价值

从数据到知识

直接使用数据面临的困难

1. 数据量大
2. 噪声和异常多
3. 数据特征高维
4. 模式特征不显著

理解海量数据远远超出了人类的能力，收集在大型数据库中的数据变成了“数据坟墓”，有必要系统地开发数据挖掘工具，将数据坟墓转换成知识“金块”。



We are drowning in data, but starving for knowledge!

数据挖掘的定义

Data Mining(knowledge discovery from data)

- 从大量数据中提取有价值的模式或知识的过程

别名

- 数据库中的知识发现 (Knowledge Discovery in Databases, KDD)
- 知识提取 (Knowledge Extraction)
- 数据/模式分析 (Data/Pattern Analysis)
- data archeology, data dredging, information harvesting, business intelligence...

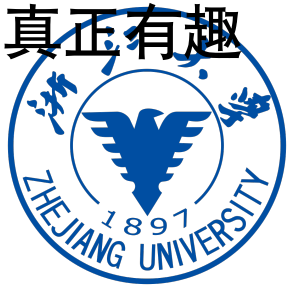
不是所有数据分析都是数据挖掘

- 简单的检索和查询处理
- 演绎专家系统

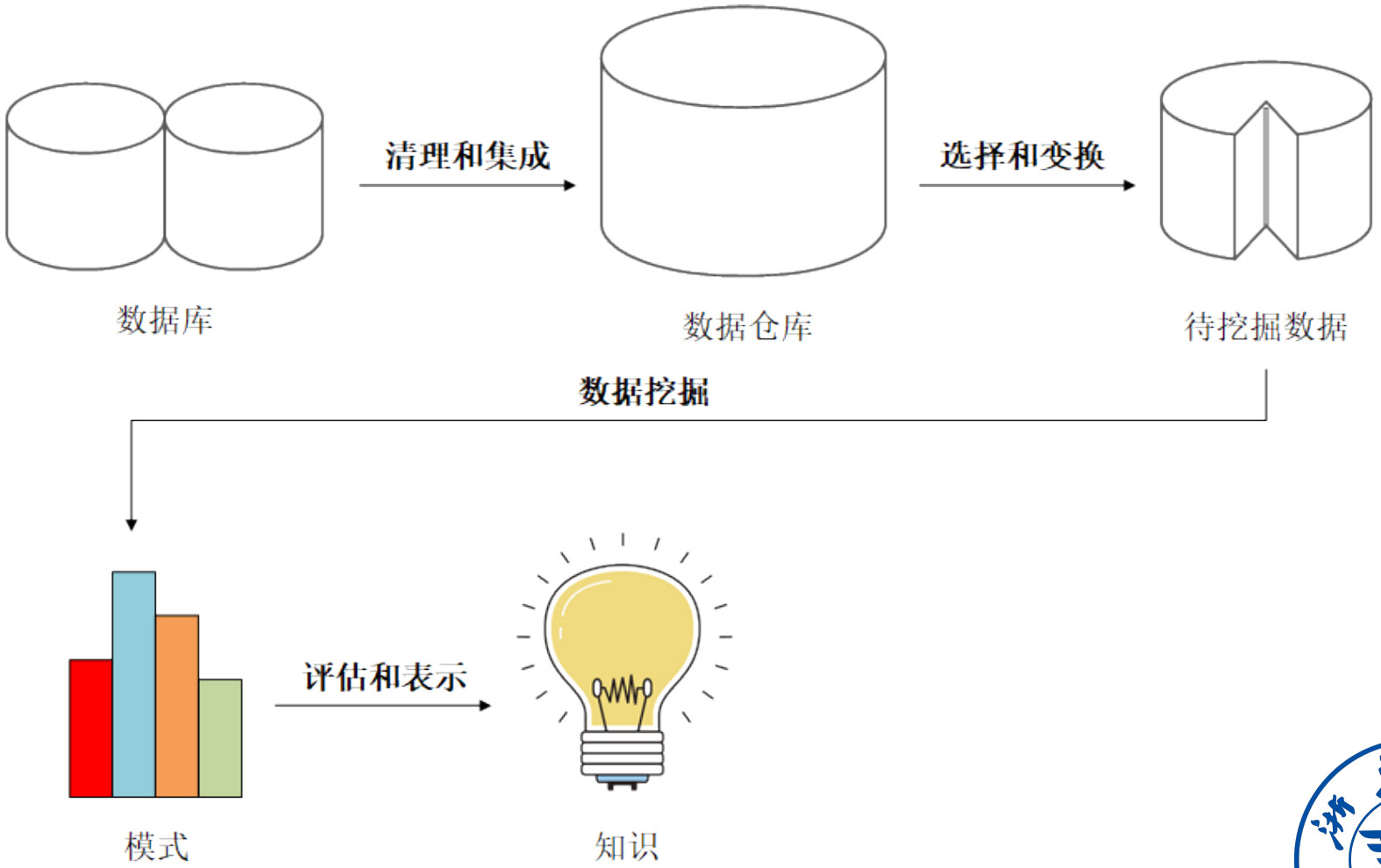
数据挖掘的步骤

数据挖掘是将数据转化为知识的过程，其主要包含如下的步骤：

1. 数据清理（消除噪声和删除不一致数据）
2. 数据集成（多种数据源可以组合在一起）
3. 数据选择（从数据库中提取与分析任务相关的数据）
4. 数据变换（通过汇总或聚集操作，把数据变换和统一成适合挖掘的形式）
5. 数据挖掘（基本步骤，使用智能方法提取数据模式）
6. 模式评估（根据兴趣度度量，识别代表知识的真正有趣的模式）
7. 知识表示（使用可视化和知识表示技术，识别代表知识的真正有趣的模式）



数据挖掘的步骤



数据挖掘的主要步骤

从数据中我们可以挖掘什么知识

描述性（descriptive）知识：

1. 类/概念描述
2. 频繁模式（频繁项集，频繁序列，频繁结构）
3. 关联分析，因果推断
4. 分类（离散标号）与回归（连续数值）
5. 聚类分析（数据内在关系，不考虑标号。最大化类内相似性，最小化类间相似性）
6. 异常检测

预测性（predictive）知识：
在当前数据上进行归纳，对未来进行预测

知识——有一定价值的模式

数据挖掘系统具有产生数以千计，甚至数以万计模式或规则的潜在能力。然而，并不是所有模式都是有用的。

有价值的模式：

1. 易于被人理解
2. 在某种确信度上，对于新的或检验数据是有效的
3. 是潜在有用的
4. 是新颖的

模式价值度量

1. 客观度量：基于所发现模式的结构和相关的统计量
2. 主观度量：基于用户对数据的观念，反映用户需要和兴趣

数据挖掘的常用技术

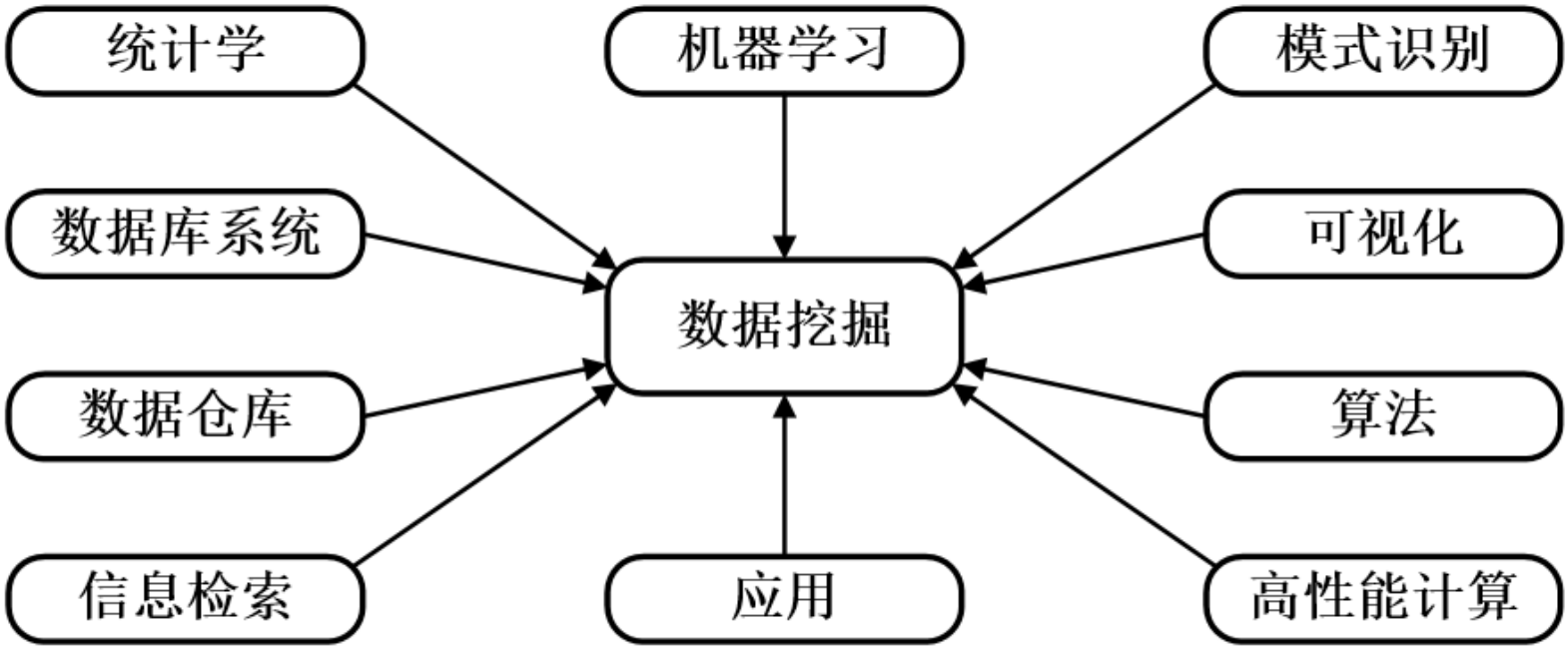
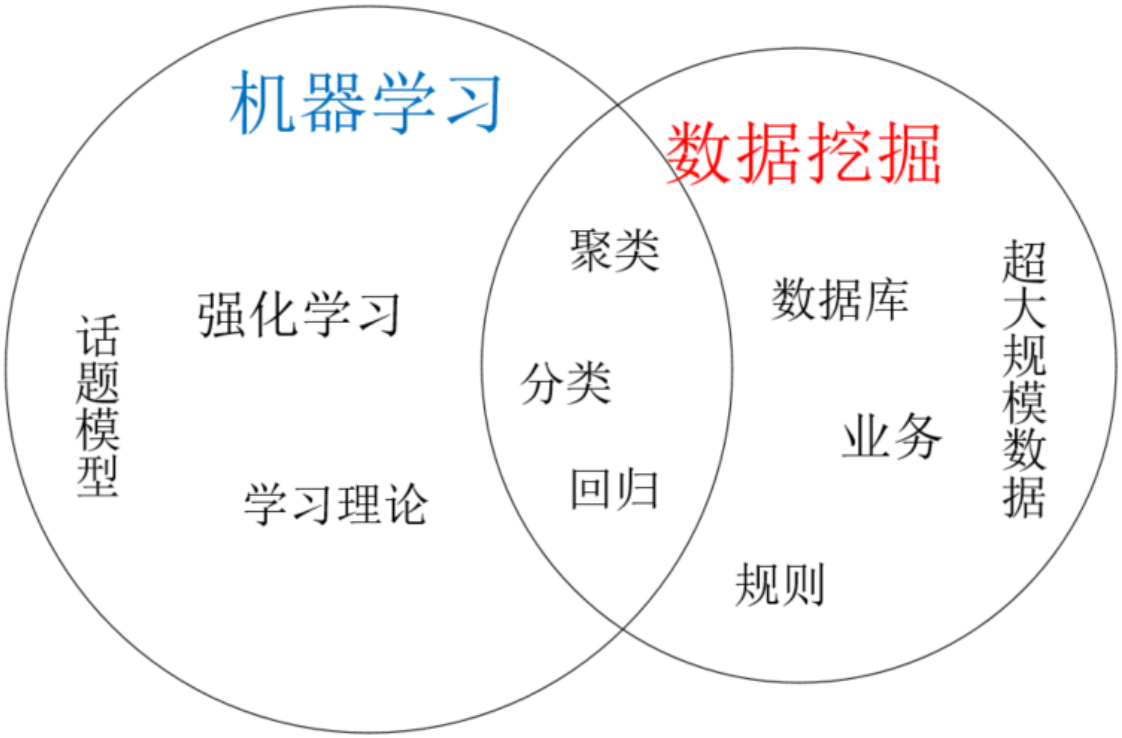


图: 数据挖掘的常用技术

数据挖掘与机器学习

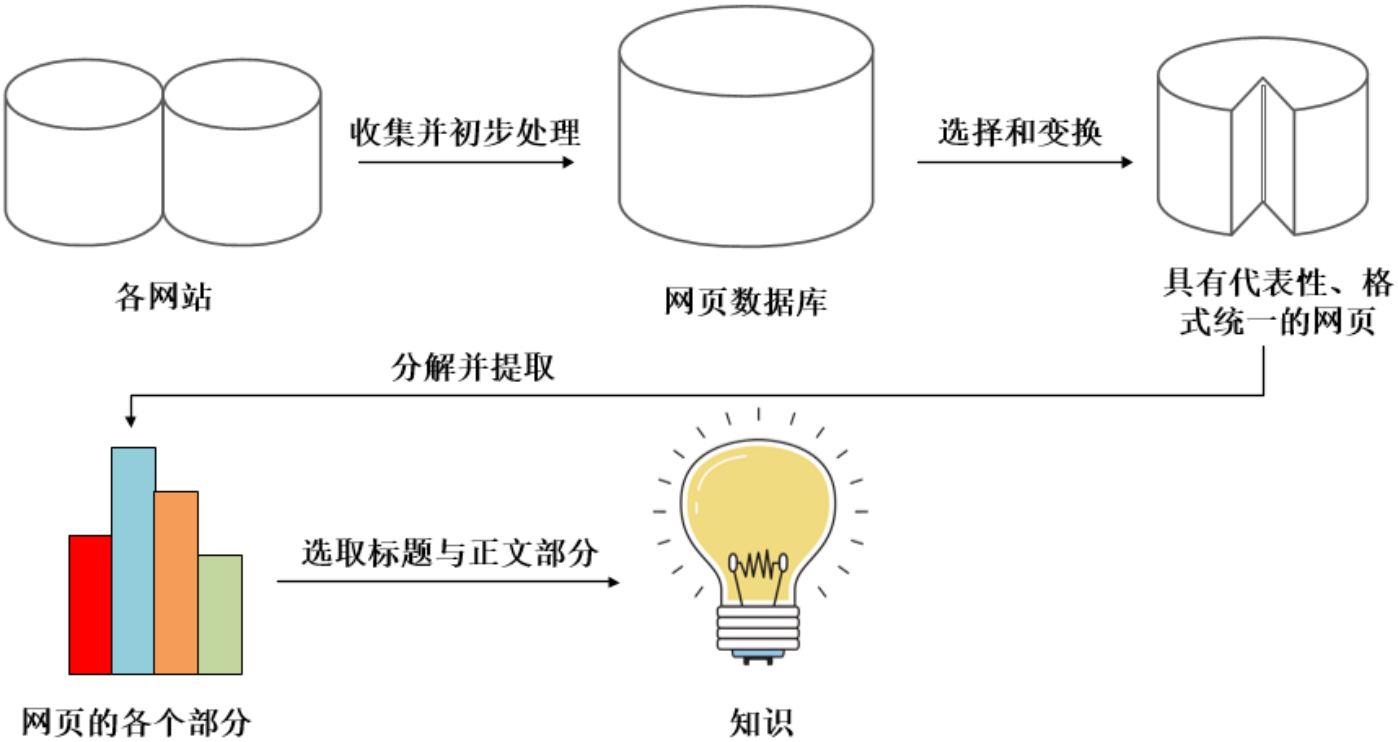
数据挖掘与机器学习有许多相似之处。对于分类和聚类任务，机器学习研究通常关注模型的准确率。除准确率外，数据挖掘研究非常强调挖掘方法在大型数据集上的有效性和可伸缩性，以及处理复杂数据类型的方法，开发新的，非传统的方法。
机器学习是用数据来帮助训练模型，来逼近目标。数据挖掘是从数据中找到有用的知识。



数据挖掘的应用场景

服务于视力障碍者的网页主体数据提取与展示。

以新闻网站为例，一般的读屏软件会将整个页面的数据转换为声音输出。然而，这些数据中存在大量无意义的部分：网页导航栏，广告，侧边栏等等。对于用户而言，他们所需要的仅仅为网页的主体部分：标题和正文——即我们想要挖掘的知识。



数据挖掘的应用场景

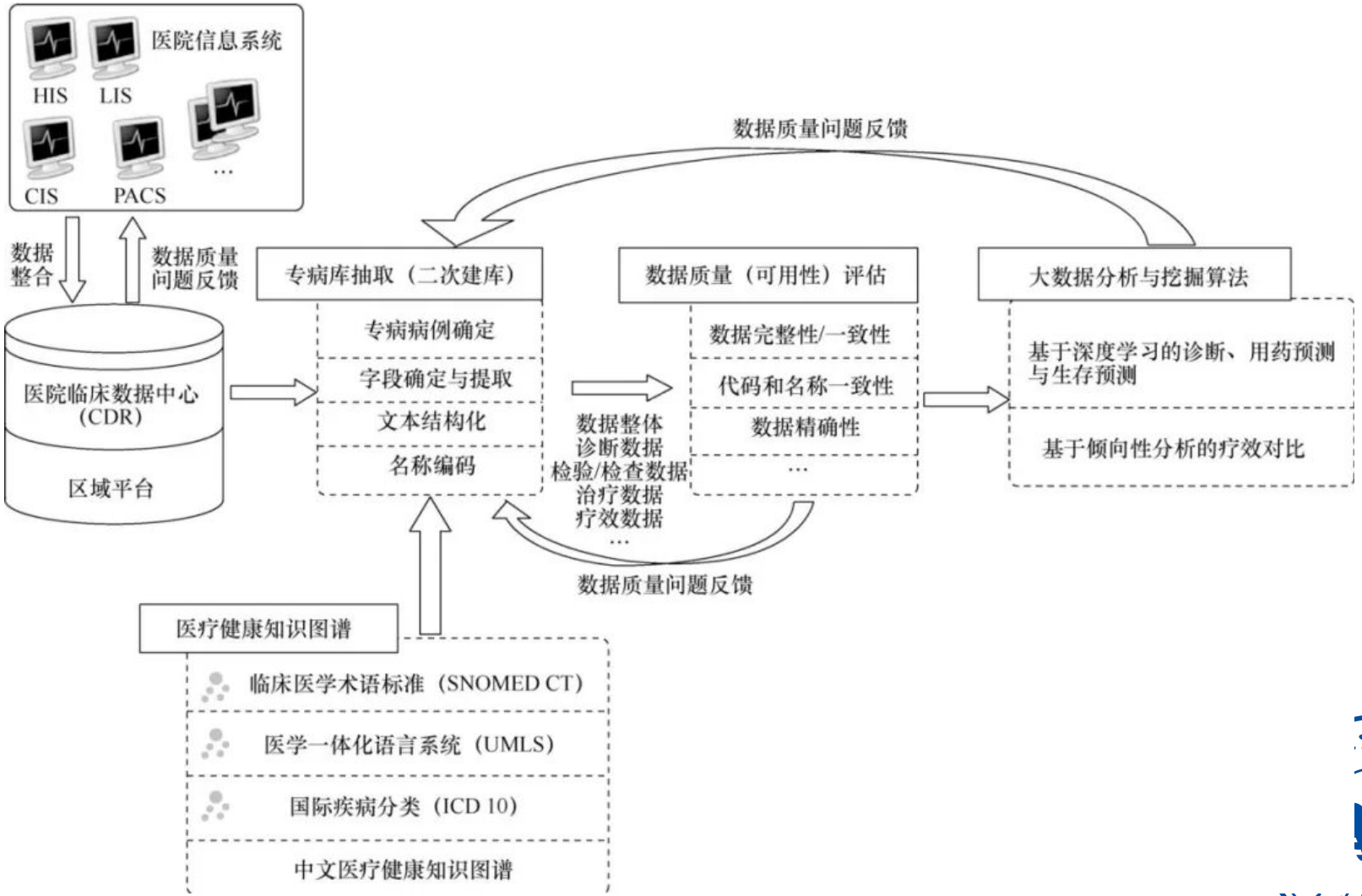


图: 基于电子病历的临床数据挖掘

数据挖掘的应用场景

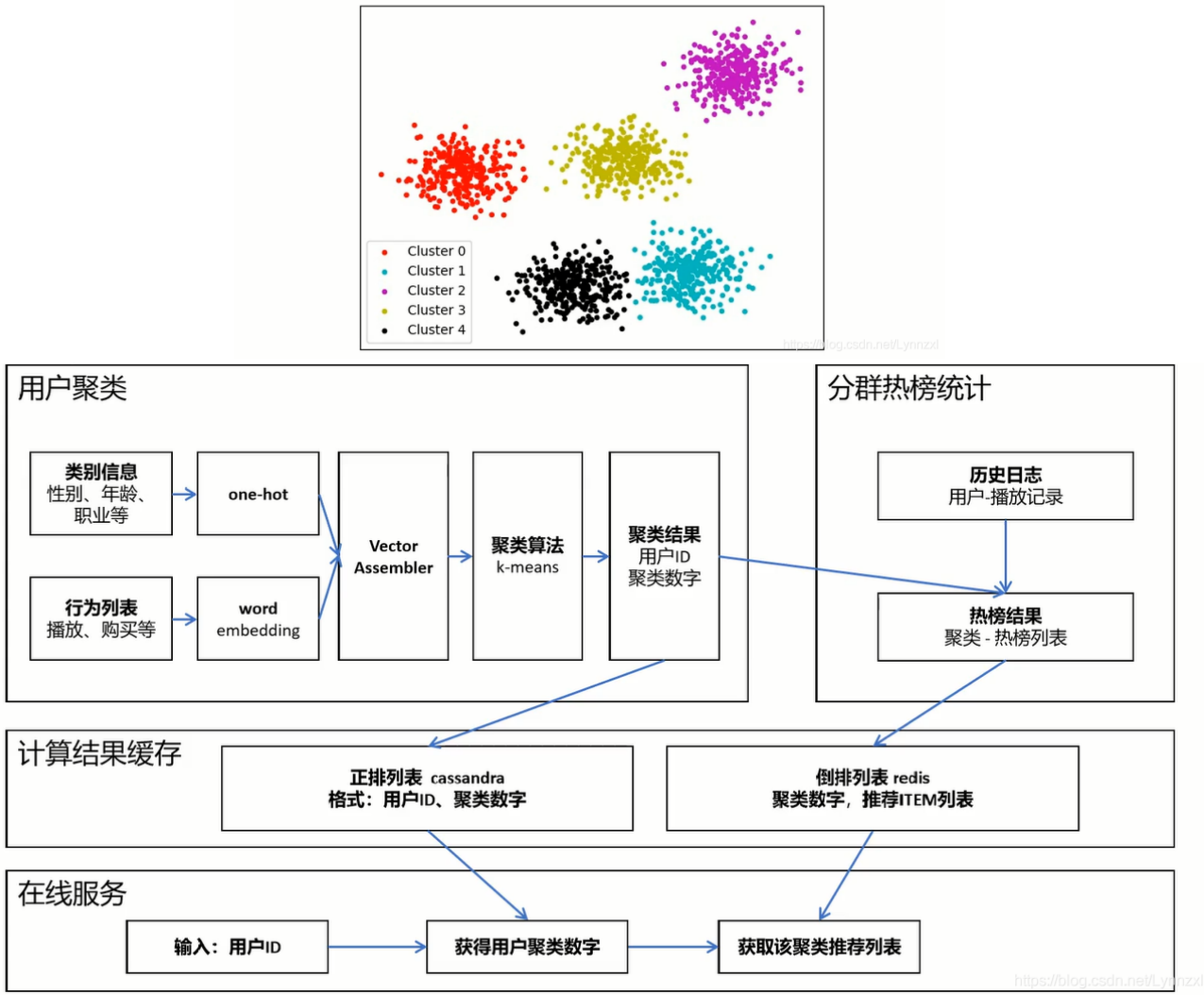


图: 基于用户聚类的推荐系统

数据挖掘的应用场景

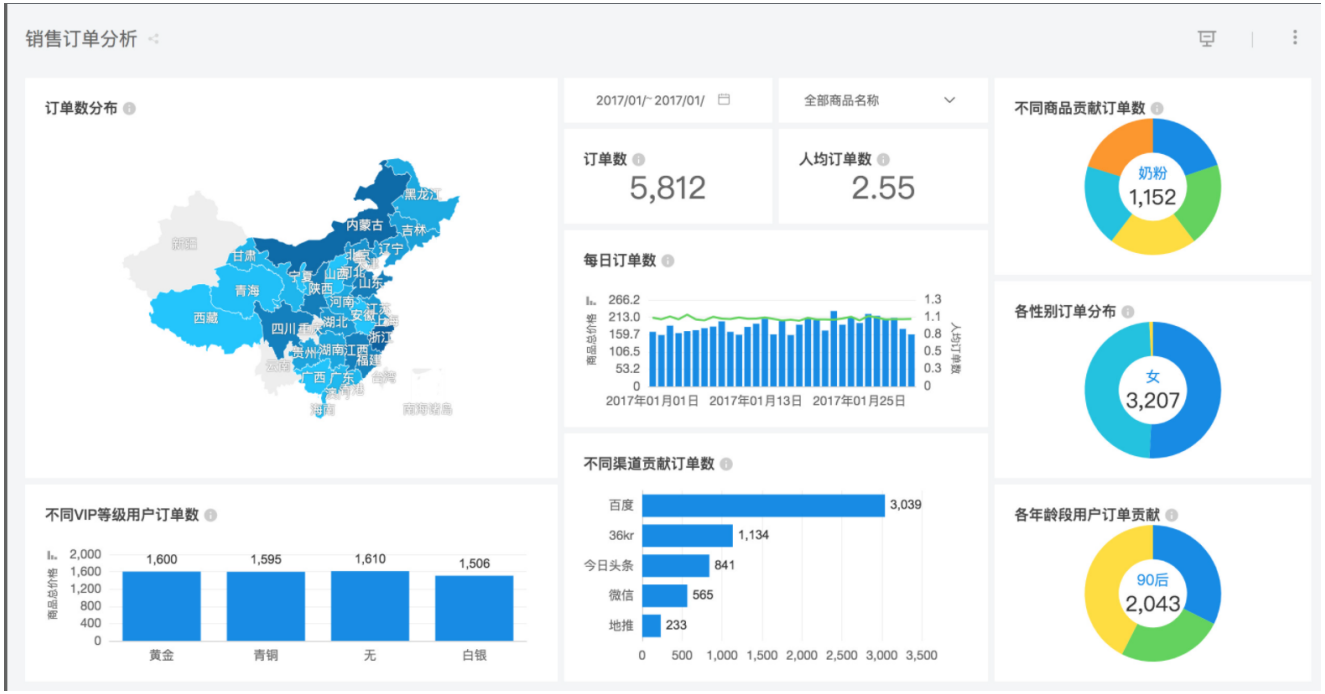


图: 基于数据挖掘的商务智能

数据挖掘的应用场景

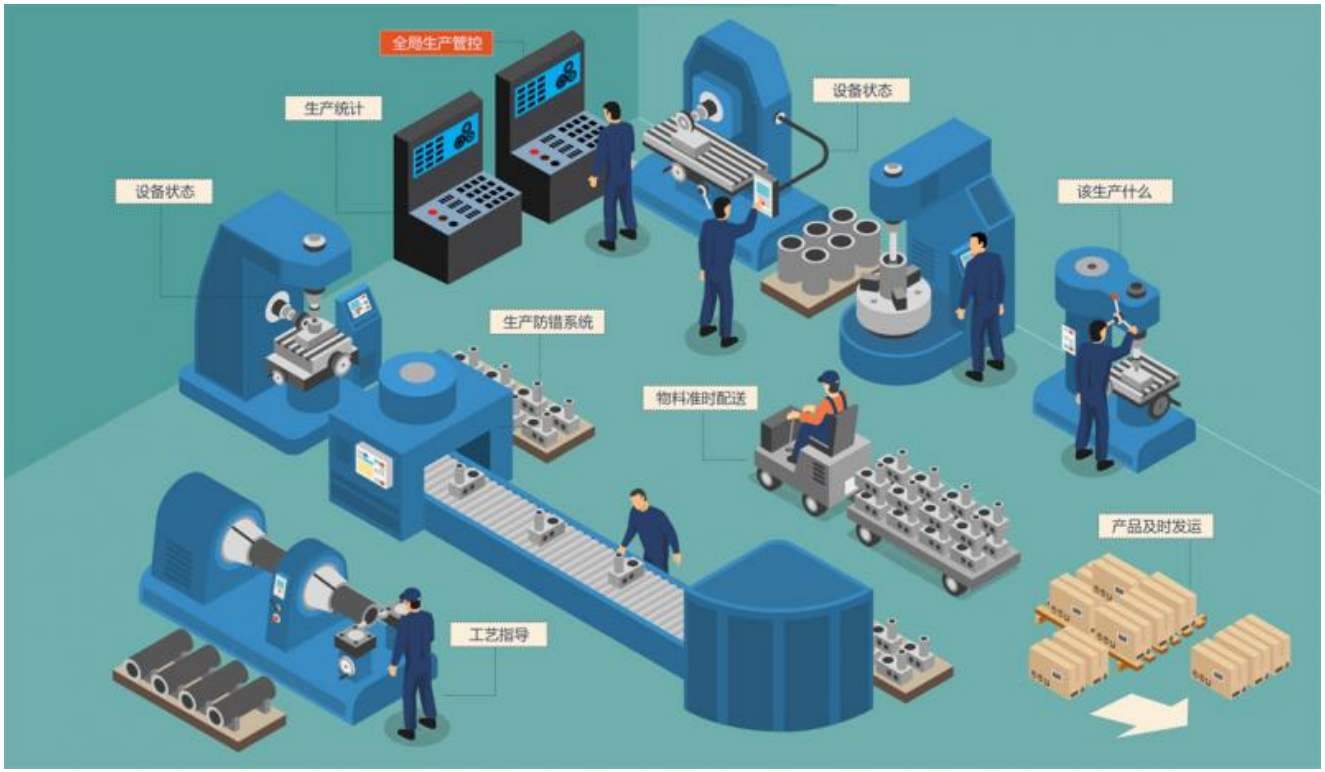


图: 基于数据挖掘的工业智能

数据挖掘的主要挑战

虽然数据挖掘得到了广泛的研究，但是仍然面临如下的挑战：

1. 新的知识类型
2. 数据特征高维
3. 跨学科交叉
4. 实时性要求高
5. 数据不确定性强，噪声大
6. 数据隐私保护

数据挖掘领域的发展

- 1989 IJCAI Workshop on Knowledge Discovery in Databases
  - Knowledge Discovery in Databases (G. Piatetsky-Shapiro and W. Frawley, 1991)
- 1991-1994 Workshops on Knowledge Discovery in Databases
  - Advances in Knowledge Discovery and Data Mining (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 1996)
- 1995-1998 International Conferences on Knowledge Discovery in Databases and Data Mining (KDD' 95-98)
  - Journal of Data Mining and Knowledge Discovery (1997)
- ACM SIGKDD conferences since 1998 and SIGKDD Explorations
- More conferences on data mining
  - PAKDD (1997), PKDD (1997), SIAM-Data Mining (2001), (IEEE) ICDM (2001), etc.
- ACM Transactions on KDD starting in 2007

数据挖掘的会议与期刊

- KDD 会议
  - ACM SIGKDD Int. Conf. on Knowledge Discovery in Databases and Data Mining (KDD)
  - SIAM Data Mining Conf. (SDM)
  - (IEEE) Int. Conf. on Data Mining (ICDM)
  - European Conf. on Machine Learning and Principles and practices of Knowledge Discovery and Data Mining (ECML-PKDD)
  - Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)
  - Int. Conf. on Web Search and Data Mining (WSDM)
- 其他相关会议
  - DB 会议: ACM SIGMOD, VLDB, ICDE, EDBT, ICDT, ···
  - Web and IR 会议: WWW, SIGIR, WSDM
  - ML 会议: ICML, NIPS　PR 会议: CVPR
- 期刊
  - Data Mining and Knowledge Discovery (DAMI or DMKD)
  - IEEE Trans. On Knowledge and Data Eng. (TKDE)
  - KDD Explorations
  - ACM Trans. on KDD

课程结构

数据挖掘与应用课程结构

理论	应用
认识数据，数据预处理	
支持向量机 SVM	
	决策树与 Boosting 算法
经典聚类算法	
	深度聚类算法
离散数据异常检测方法	
	时序数据异常检测方法
工业数据挖掘	

以数据挖掘主要任务为基础，以实用为主要目标。

Q&A

数据挖掘中常用的数据类型

虽然自然界存在海量的数据，但是仅能够被计算机存储和识别的数据才能作为数据挖掘的对象。而在当前数据挖掘系统中常用的数据类型主要包括：

1. 记录型数据 (数据之间相互独立，共享特征空间)
2. 序列化数据（数据之间通过之间维度进行排列）
3. 网络数据（数据之间通过关系显式链接）
4. 结构化数据（数据的结构固定，只在对应的特征不同）

记录型数据是基本类型，不同类型的数据之间可以相互转化

数据的表示

数据由样本构成，每个样本由特征（attribute）进行描述，数据通常以 N * K 的矩阵/张量形式进行表示

pandas.DataFrame: N=4, K=5

	姓名	年龄	性别	平均绩点	是否选课
0	张三	22	男	3.83	True
1	李四	21	女	3.87	False
2	李雷	20	男	2.80	True
3	韩梅梅	22	女	4.00	True

数据的特征

用来描述一个给定对象的一组属性称作属性向量（或特征向量）
常见的属性类型包括：

1. 标称属性（nominal/categorical attribute）：每个值代表某种类别，编码或状态。不一定要用文本表示，也可以用数字表示，除非想要理解具体的语义。如果只是用作分类，只需要数字即可。常见的包括用户名，昵称，身份证
2. 离散属性（binary/bool attribute）：一共只有两种状态
3. 序数属性值之间具有意义的序（ranking）（优秀，良好，中等，差）
4. 连续特征 numeric/quantitative
   1. 区间标度 Interval-scaled
   2. 比率标度 Ratio-scaled

姓名(string)　年龄(int)　性别(string)　平均绩点(float)　是否选课(bool)

样本的相关性描述

前面提到了数据可以用不同形式的属性表示，那么如何基于属性对数据之间的关系进行描述呢？

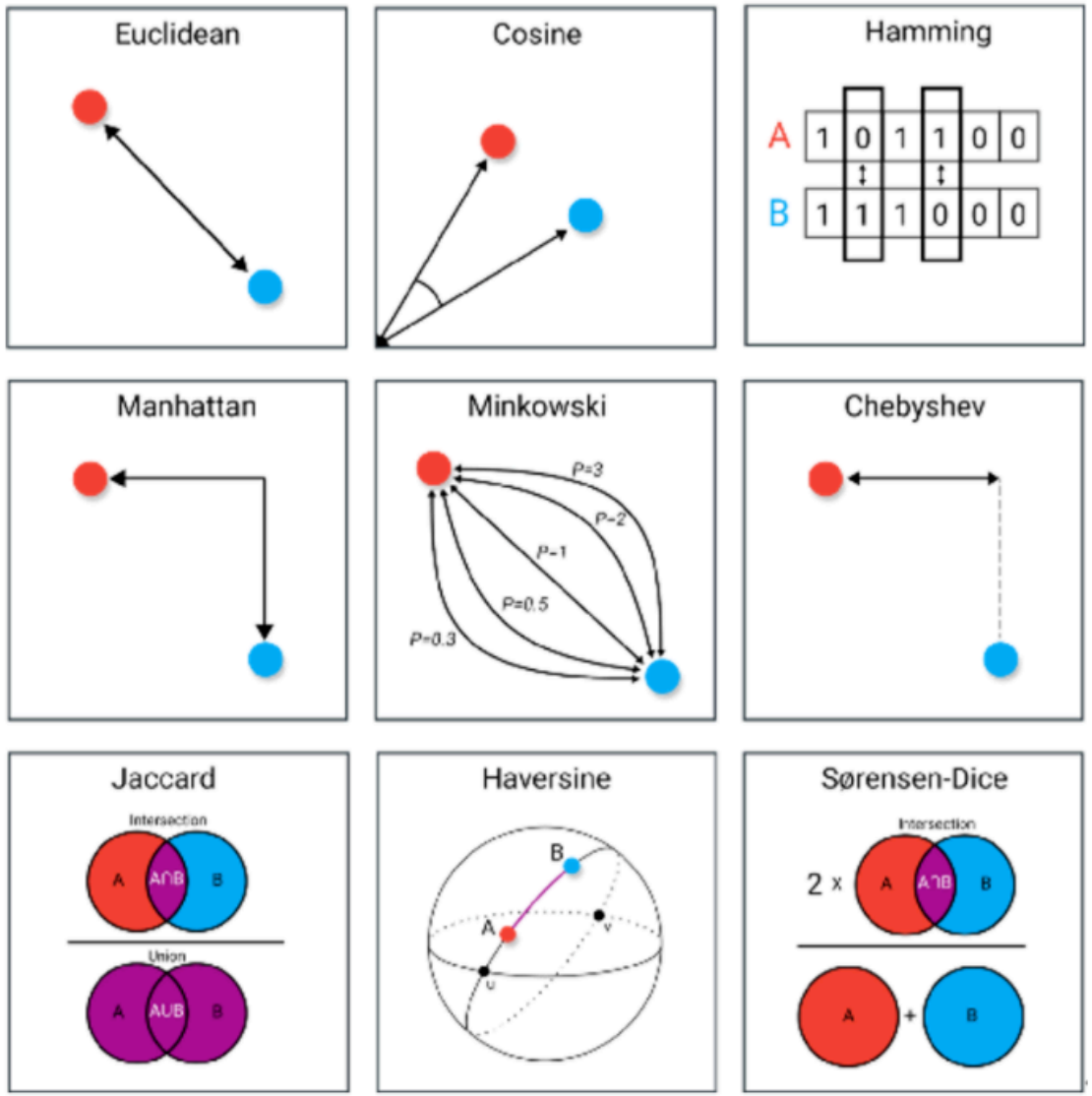
1. 标称属性的距离度量
2. 二元属性的距离度量（Jaccard 以及 TF/FN 等等）
3. 数值属性的距离度量
4. 序数属性的距离度量

数值属性的距离度量

在许多数据挖掘的任务中，数据（样本）之间的的相似性或距离是最基础的信息单元。带有数值属性的数据之间的距离度量得到了广泛的研究。距离度量（distance measure）满足的基本性质

1. 非负性：$dist(x_i, x_j) \geq 0$
2. 同一性：$dist(x_i, x_j) = 0 \quad \text{iff} \quad x_i = x_j$
3. 对称性：$dist(x_i, x_j) = dist(x_j, x_i)$
4. 传递性：$dist(x_i, x_j) \leq dist(x_i, x_k) + dist(x_k, x_j)$

数值属性的距离度量


Euclidean
Cosine
Hamming
A 1 0 1 1 0 0
B 1 1 1 0 0 0
Manhattan
Minkowski
P=3
P=2
P=1
P=0.5
P=0.3
Chebyshev
Jaccard
Intersection
A∩B
Union
A∪B
Haversine
Sørensen-Dice
2 x

闵可夫斯基距离

定义

给定样本 $x_i = (x_{i1}, x_{i2}, ..., x_{id}) \in \mathcal{R}^d$ 和 $x_j = (x_{j1}, x_{j2}, ..., x_{jd}) \in \mathcal{R}^d$，闵可夫斯基距离（Minkowski Distance）定义为:

$$\left(\sum_{i=1}^{n} |x_i - x_j|^p \right)^{1/p}$$

上述定义也称为 L_p 范数（norm）。

闵可夫斯基距离是数据挖掘/机器学习中最常用的一组距离度量。

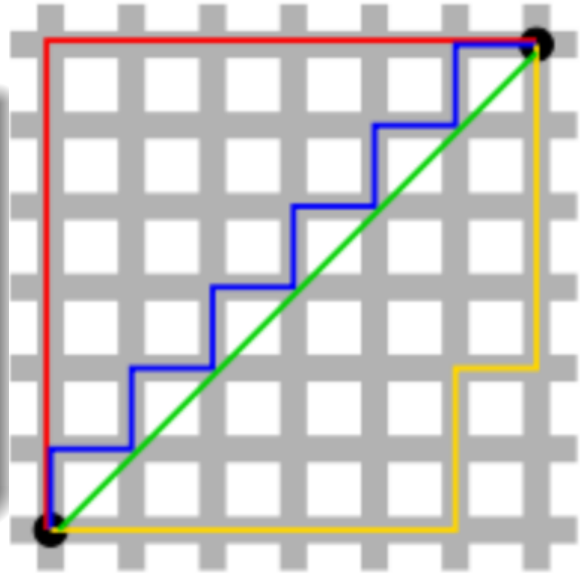
闵可夫斯基距离的一般形式

曼哈顿距离 Manhattan Distance

当 $p = 1$ 时，闵可夫斯基距离为曼哈顿距离：

$$\operatorname{dist}(x_i, x_j) = \|\mathbf{x}_i - \boldsymbol{x}_j\|_1 = \sum_{k=1}^{d} |x_{ik} - x_{jk}|$$

曼哈顿距离也称为街区距离（City Block Distance）



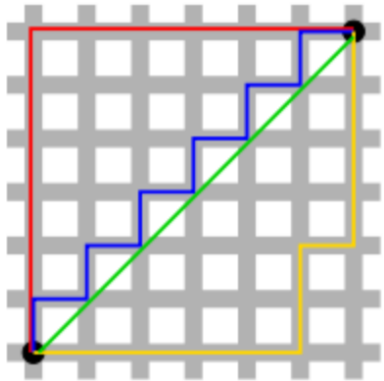
曼哈顿距离与欧式距离

闵可夫斯基距离的一般形式

欧氏距离 Euclidean Distance

当 $p = 2$ 时，闵可夫斯基距离为欧氏距离：

$$dist(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$



曼哈顿距离与欧式距离

随着数据维度的增加，欧几里得距离的作用就越小。这与维度的诅咒有关，它涉及到高维空间的概念，并不像我们直观地期望的那样，从二维或三维空间中发挥作用。

闵可夫斯基距离的一般形式

切比雪夫距离 Chebyshev Distance

当 $p = \infty$ 时，闵可夫斯基距离为欧氏距离：

$$dist(x, y) = \max_d(|x_i - y_i|) = \lim_{k \to \infty} \left(\sum_{i=1}^{d} |p_i - q_i|^k \right)^{1/k}$$

.

闵可夫斯基距离的一般形式

国际象棋中，国王可以直行、横行、斜行。国王走一步，可以移动到相邻的 8 个方格的任意一个。国王从格子（X1,Y1）到格子（X2,Y2）最少需要多少步？这个距离就是切比雪夫距离。

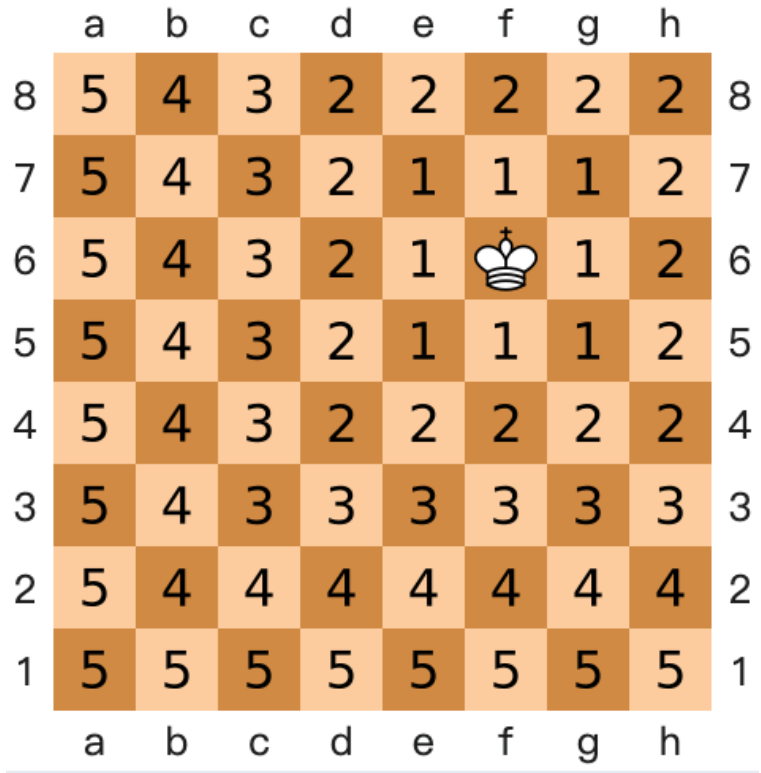
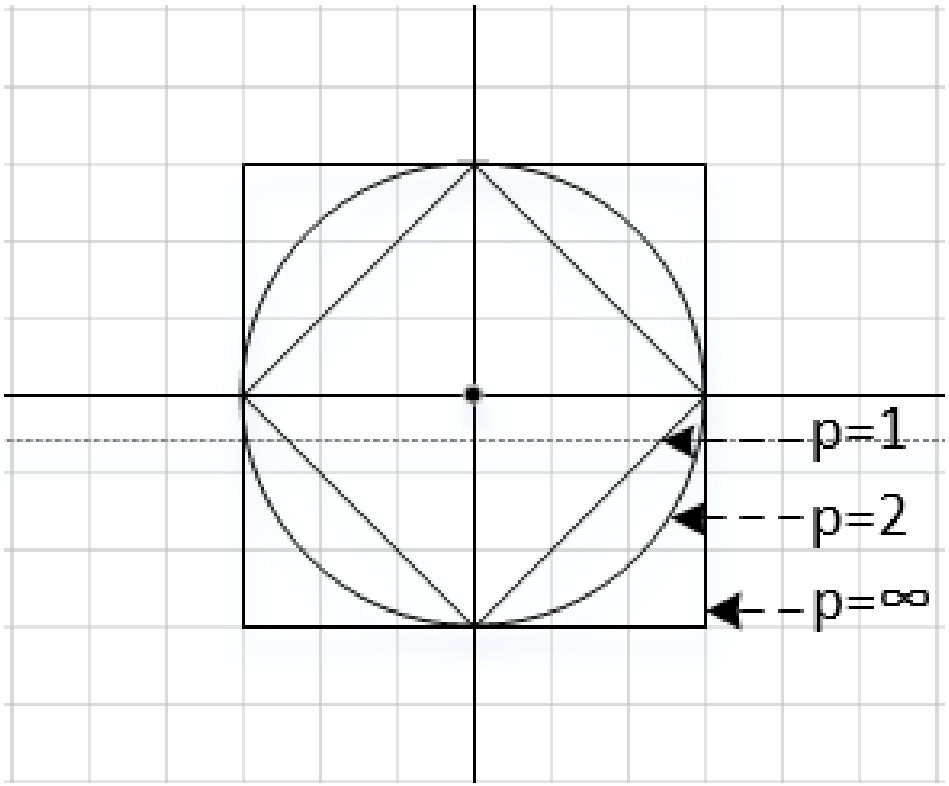


图: 切比雪夫距离

闵可夫斯基距离的可视化

闵可夫斯基距离的可视化如图所示：



常见的闵可夫斯基距离的可视化结果

从向量内积到余弦相似度

向量内积是计算向量距离的基本方式

定义

$$Inner(x,y) =< x,y >= \sum_i x_i y_i$$

向量内积是没有界限的，一种解决方法是除以长度后再求内积，即余弦相似度（Cosine Similarity）

定义

$$\text{Cos Sim}(x,y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}} = \frac{\langle x,y\rangle}{\|x\|\|y\|}$$

余弦相似度与向量的大小无关，只与向量的方向有关。余弦相似度不满足度量测度性质，因此被称为非度量测度（nonmetric measure）

从余弦相似度到皮尔逊相关系数

余弦相似度面临的问题是不满足平移不变性，而皮尔逊相关系数则可以满足平移不变性

定义

皮尔逊相关系数（Pearson Correlation）定义为：

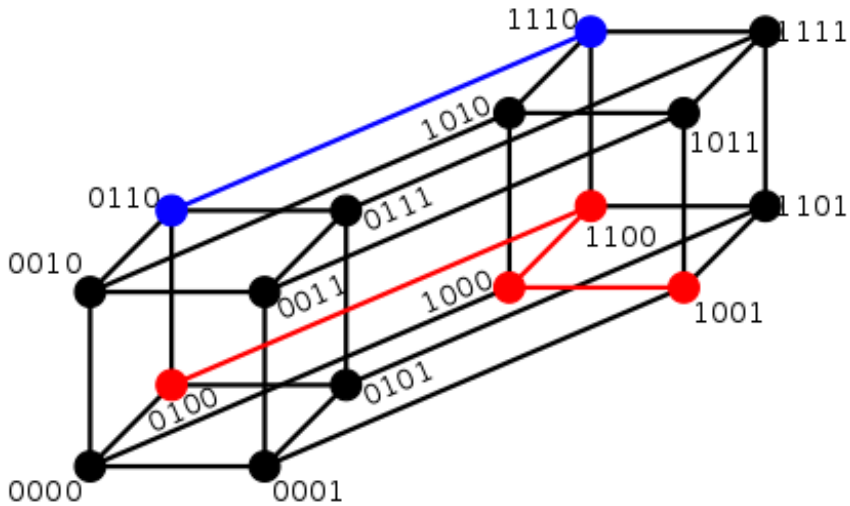
$$\begin{aligned}\mathrm{Corr}(x,y) &= \frac{\sum_i (x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum (x_i-\bar{x})^2}\sqrt{\sum (y_i-\bar{y})^2}} = \frac{\langle x-\bar{x}, y-\bar{y}\rangle}{\|x-\bar{x}\|\|y-\bar{y}\|} \\ &= \mathrm{Cos\,Sim}(x-\bar{x}, y-\bar{y})\end{aligned}$$

皮尔逊相关系数具有平移不变性和尺度不变性，计算出了两个向量（维度）的相关性。

汉明距离 (Hamming Distance)

在信息理论中，Hamming Distance 表示两个等长字符串在对应位置上不同字符的数目。对于两个数字来说，汉明距离就是转成二进制后，对应的位置值不相同的个数。对于二进制串 a 和 b 来说，汉明距离等于 a XOR b 中 1 的数目

汉明距离主要应用在通信编码领域上，用于制定可纠错的编码体系。在机器学习领域中，汉明距离也常常被用于作为一种距离的度量方式。



汉明距离示例

非数值特征数据的关系度量

除了数据特征数据之间的关系度量，数据挖掘任务中还有很多特殊形式的数据：

1. 集合数据 $\Rightarrow$ Jaccard 距离
2. 概率分布 $\Rightarrow$ Divergence（散度）
   1. KL-Divergence
   2. JS-Divergence
3. Wasserstein Distance

杰卡德距离 (Jaccard Distance)

Jaccard 距离是度量集合之间相似度的基本方法：

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

常用于推荐系统和检索系统中。

分布的度量

KL-Divergence

两个离散分布的 KL 散度定义为：

$$D_{\mathrm{KL}}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right)$$

两个连续分布的 KL 散度定义为：

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x)\log\left(\frac{p(x)}{q(x)}\right)dx$$

其他：f-divergence，互信息

常见距离度量的代码实现

目前大部分经典距离度量都已有开源实现，在 Sklearn，Pytorch 等框架中已经原生支持，建议直接调用。

metrics.pairwise.additive_chi2_kernel(X[, Y])	Computes the additive chi-squared kernel between observations in X and Y.
metrics.pairwise.chi2_kernel(X[, Y, gamma])	Computes the exponential chi-squared kernel X and Y.
metrics.pairwise.cosine_similarity(X[, Y, ...])	Compute cosine similarity between samples in X and Y.
metrics.pairwise.cosine_distances(X[, Y])	Compute cosine distance between samples in X and Y.
metrics.pairwise.distance_metrics()	Valid metrics for pairwise_distances.
metrics.pairwise.euclidean_distances(X[, Y, ...])	Considering the rows of X (and Y=X) as vectors, compute the distance matrix between each pair of vectors.
metrics.pairwise.haversine_distances(X[, Y])	Compute the Haversine distance between samples in X and Y.
metrics.pairwise.kernel_metrics()	Valid metrics for pairwise_kernels.
metrics.pairwise.laplacian_kernel(X[, Y, gamma])	Compute the laplacian kernel between X and Y.
metrics.pairwise.linear_kernel(X[, Y, ...])	Compute the linear kernel between X and Y.
metrics.pairwise.manhattan_distances(X[, Y, ...])	Compute the L1 distances between the vectors in X and Y.
metrics.pairwise.nan_euclidean_distances(X)	Calculate the euclidean distances in the presence of missing values.
metrics.pairwise.pairwise_kernels(X[, Y, ...])	Compute the kernel between arrays X and optional array Y.
metrics.pairwise.polynomial_kernel(X[, Y, ...])	Compute the polynomial kernel between X and Y.
metrics.pairwise.rbf_kernel(X[, Y, gamma])	Compute the rbf (gaussian) kernel between X and Y.
metrics.pairwise.sigmoid_kernel(X[, Y, ...])	Compute the sigmoid kernel between X and Y.
metrics.pairwise.paired_euclidean_distances(X, Y)	Computes the paired euclidean distances between X and Y.
metrics.pairwise.paired_manhattan_distances(X, Y)	Compute the L1 distances between the vectors in X and Y.
metrics.pairwise.paired_cosine_distances(X, Y)	Computes the paired cosine distances between X and Y.
metrics.pairwise.paired_distances(X, Y, *[, ...])	Computes the paired distances between X and Y.
metrics.pairwise_distances(X[, Y, metric, ...])	Compute the distance matrix from a vector array X and optional Y.
metrics.pairwise_distances_argmin(X, Y, *[, ...])	Compute minimum distances between one point and a set of points.
metrics.pairwise_distances_argmin_min(X, Y, *)	Compute minimum distances between one point and a set of points.
metrics.pairwise_distances_chunked(X[, Y, ...])	Generate a distance matrix chunk by chunk with optional reduction.

数据的统计描述

用于描述数据整体特征的表征通常可以分为两大类：

1. 中心趋势性指标
2. 数据分散性指标

中心趋势性指标

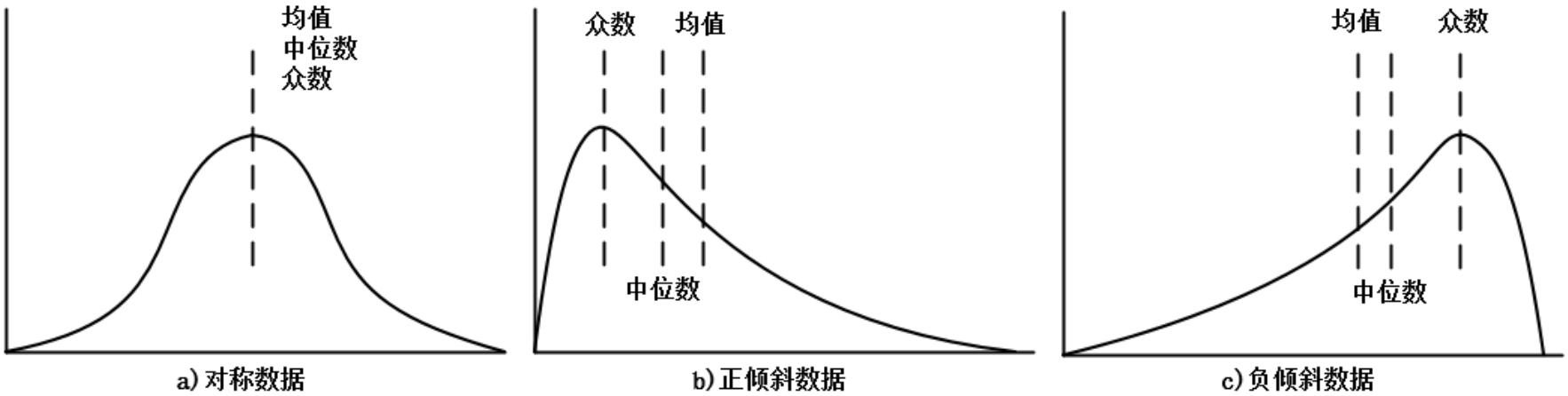
1. 均值（加权均值，截尾均值）
2. 中位数
3. 众数
4. 中列数：数据集的最大值和最小值的平均值

数据分散性指标

1. 极差
2. 分位数
   1. 四分位数
   2. 百分位数
   3. 四分位数极差
3. 方差/标准差

数据的统计描述

虽然每个数据集都可以得到对应的统计描述，但是不同的数据分布应由不同的统计指标进行度量。



不同数据分布对统计信息的影响

数据统计的可视化

常见的数据可视化工具：

1. Excel
2. MATLAB
3. Matplotlib (Python)
4. Seaborn (Python)
5. Tikz (Latex)

数据统计的可视化

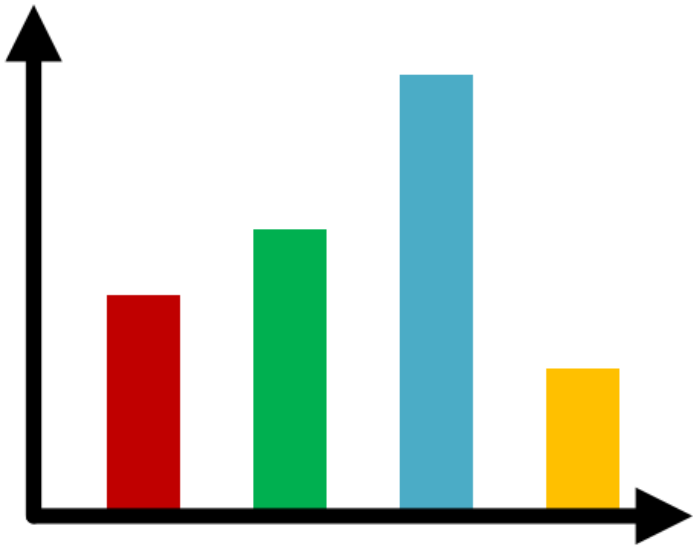
常见的数据可视化工具：

1. Excel
2. MATLAB
3. Matplotlib (Python)
4. Seaborn (Python)
5. Tikz (Latex)

数据可视化工具很多，核心是可视化的目标是否明确，呈现是否清晰简洁。盲目的可视化只会引起误解。

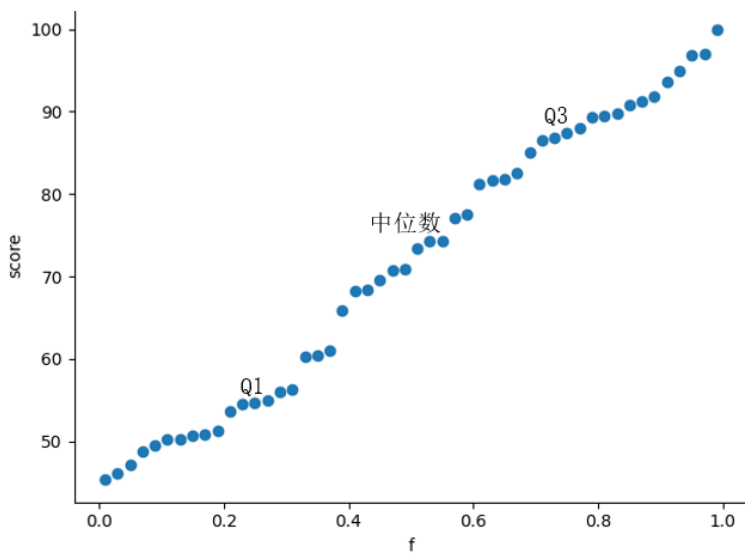
数据统计的可视化

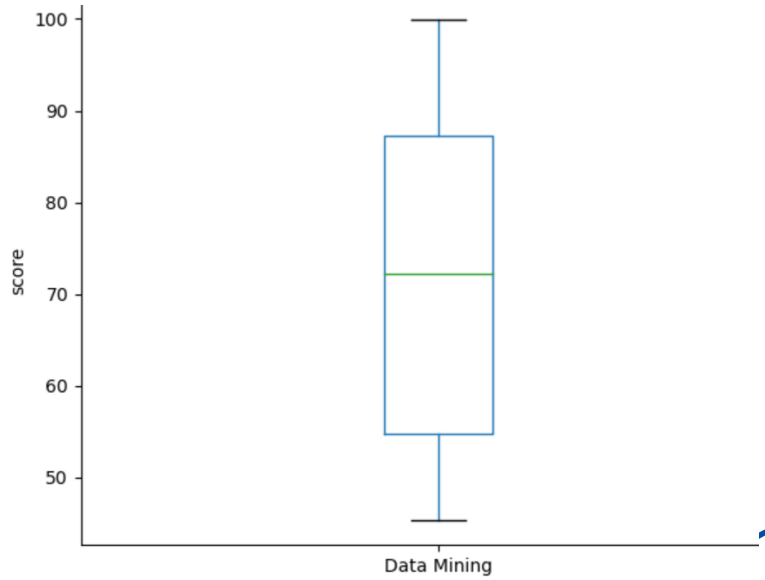
为了展示数据集中数据的频率变化以及主要高频分量，通常使用柱状图和词云图等进行可视化。



数据统计的可视化

为了展示数据的分布以及分布的关键分段指标，通常使用分位数和盒图等进行可视化。



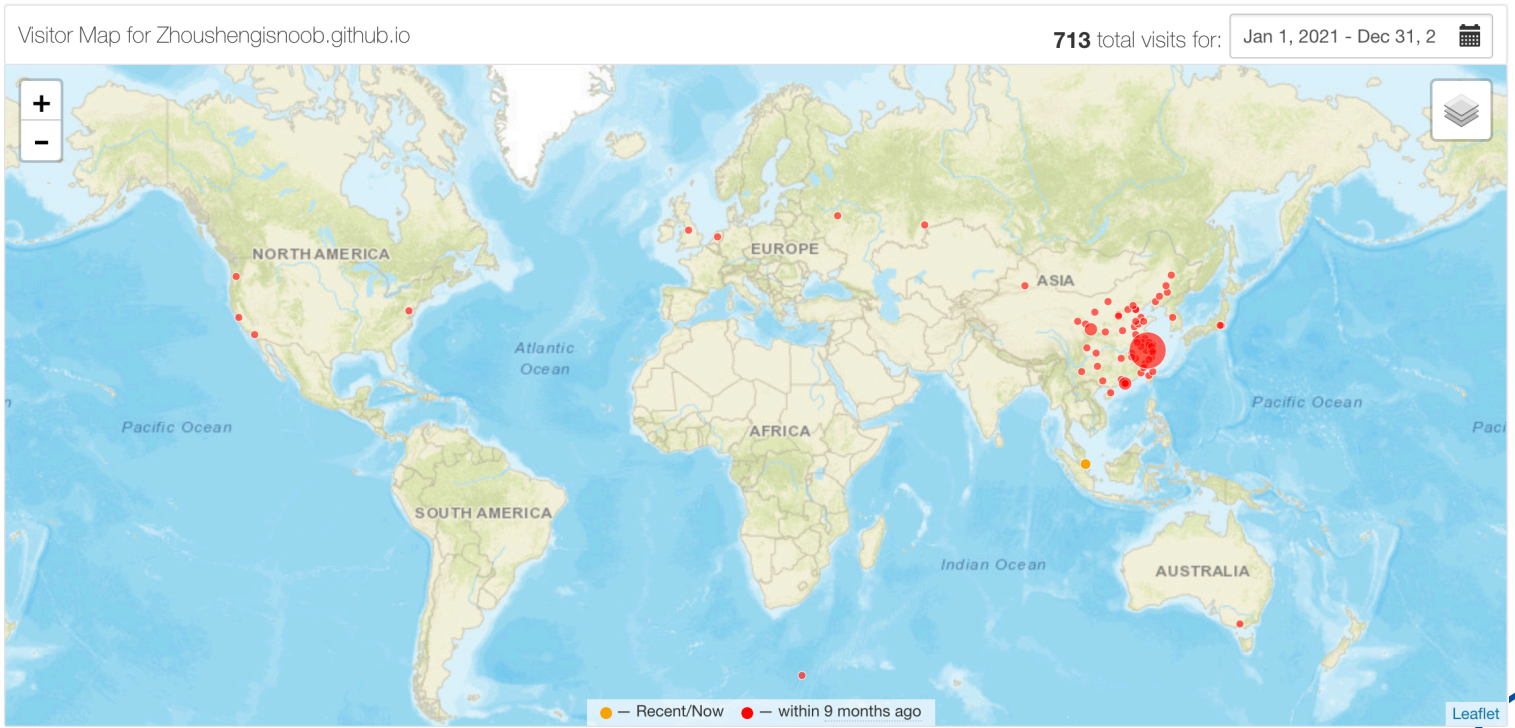


数据统计的可视化

为了展示数据的全局数值分布，通常使用热力图进行可视化。

数据统计的可视化

为了展示数据的地理分布，可以使用地理数据可视化工具。



数据的质量

高质量的数据是高效数据挖掘的前提和保障，而真实场景中数据质量却往往难以满足实用需求。

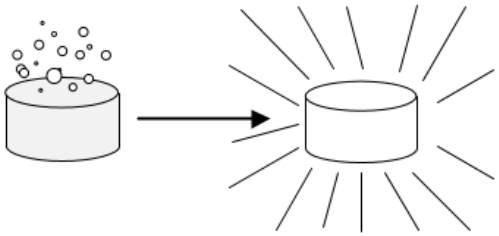
数据质量的三大问题

1. 准确性（数据采集软硬件故障，人为故障，用户主观不愿意被采集数据，传输过程的信息损失）
2. 完整性（数据的特征采集由人工制定）
3. 一致性（不同数据源的数据形式不统一）

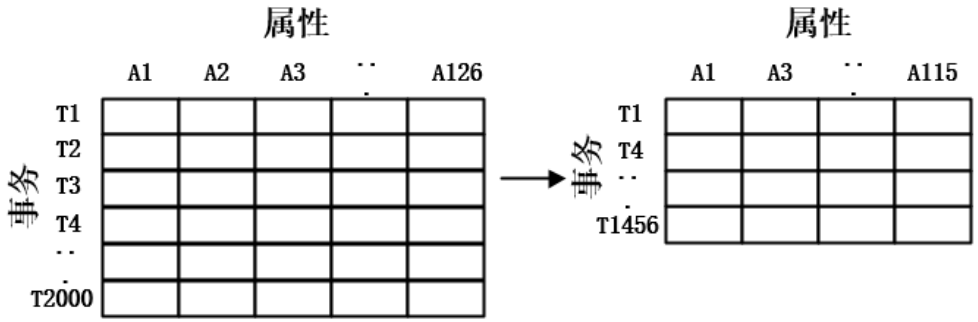
数据预处理的主要步骤

数据预处理主要包含如下的步骤：

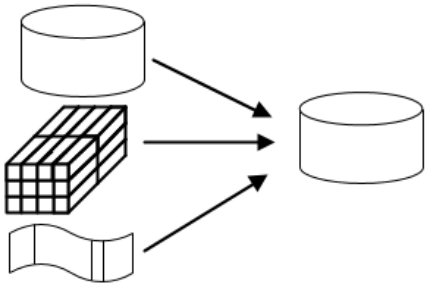
1. 数据清理（Data Cleaning）⇒ 去除缺失，噪声数据
2. 数据集成（Data Integration）⇒ 去除冗余的数据和特征
3. 数据归约（Data Reduction）⇒ 降低数据集的规模
4. 数据变换（Data Transformation）⇒ 提升数据挖掘效率



数据清洗



数据归约



数据集成

$$-2, 32, 100, 59, 48 \longrightarrow -0.02,\ 0.32,\ 1.00,\ 0.59,\ 0.48$$

数据变换

数据清理

数据清理通常包含如下的操作：填补缺失值，识别并去除噪声数据。

填补缺失值的方式：

1. 手工填写
2. 全局常量
3. 统计变量（均值，中位数）
4. 丢弃属性

清理噪声数据的方式：

1. 统计数据识别
2. 规则识别噪声
3. 异常检测算法识别噪声
4. 回归方式识别噪声

数据集成

数据集成的主要目的是将不同数据源的数据整合到统一的数据中，其中核心的问题是避免数据的冗余，判断是否冗余则主要依赖相关分析。

标称数据的相关分析：X^2（卡方）检验

$$\chi^2 = \Sigma \frac{(f_o - f_e)^2}{f_e}$$

用途：

1. 拟合优度检验 chi-squared test goodness of fit
2. 独立性检验 chi-squared test of independence

数值数据的相关分析

1. 相关系数

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \overline{A})(b_i - \overline{B})}{n\sigma_A\sigma_B}$$

2. 协方差矩阵

数据规约

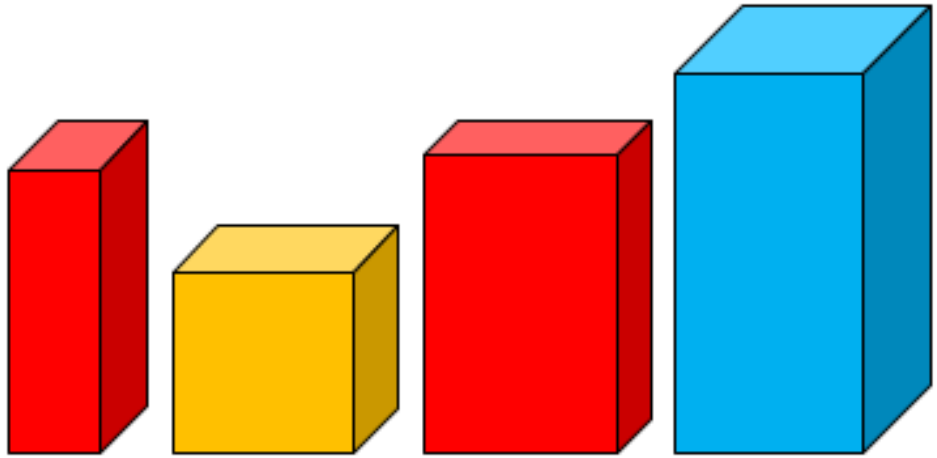
数据规约的主要目的是减少数据量，但任然接近于保持原始数据的完整性。

数据规约的主要方式包括：

1. 维度规约（dimension reduction）
   1. 小波变换
   2. 主成分分析
   3. 表征学习
2. 数量规约（numerosity reduction）
   1. 聚类
   2. 采样
3. 数据压缩（data compression）

为何要做主成分分析？

特征冗余: 在多维特征中，有些维度没有意义——听君一席话，如听一席话



	长	宽	高	底面积	体积	颜色
0	1	2	3	2	6	红
1	2	2	2	4	8	黄
2	2	1	3	2	6	红
3	2	3	4	6	24	蓝

去除冗余信息

	长	宽	高	颜色
0	1	2	3	红
1	2	2	2	黄
2	2	3	4	蓝

主成分分析 PCA

定义

PCA 的主要思想是将 n 维特征映射到 k 维上，这 k 维是全新的正交特征也被称为主成分，是在原有 n 维特征的基础上重新构造出来的 k 维特征。

PCA 的核心目标是只保留包含绝大部分方差的维度特征，而忽略包含方差几乎为 0 的特征维度，实现对数据特征的降维处理。

内积与正交基

向量的内积操作讲两个向量映射为实数：

$$(a_1, a_2, \cdots, a_n)^\top \cdot (b_1, b_2, \cdots, b_n)^\top = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$

以二维空间为例，向量的内积可以理解为两个向量的模长乘积乘以两个向量的夹角余弦：

$$A \cdot B = |A||B|\cos(a) = |A|\cos(a)|B|$$

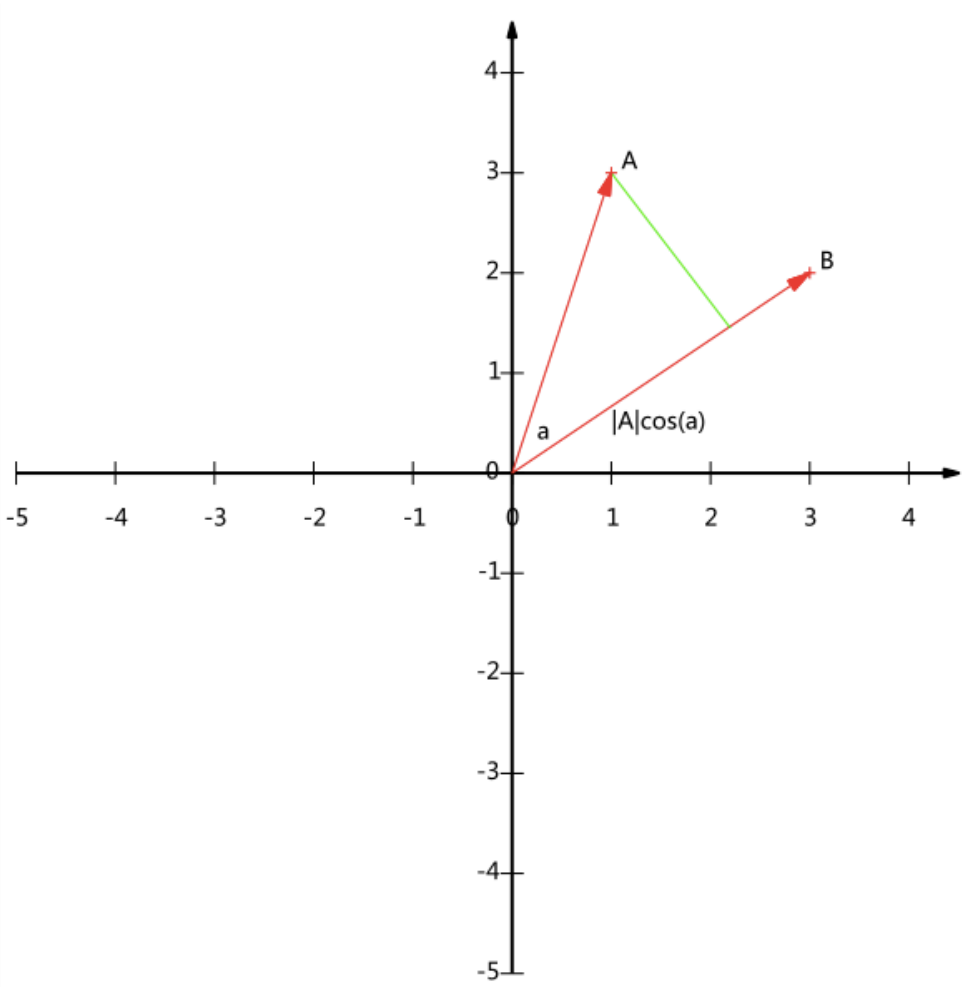
其中 $|A|$ 是向量 A 的模（标量长度），a 是两个向量之间的夹角。$|A|cos(a)$ 称为向量 A 投影的矢量长度。

标量长度与矢量长度

标量长度总是大于等于 0，值就是线段的长度；而矢量长度可能为负，其绝对值是线段长度，而符号取决于其方向与标准方向相同或相反。

内积与正交基

二维空间中的内积操作：



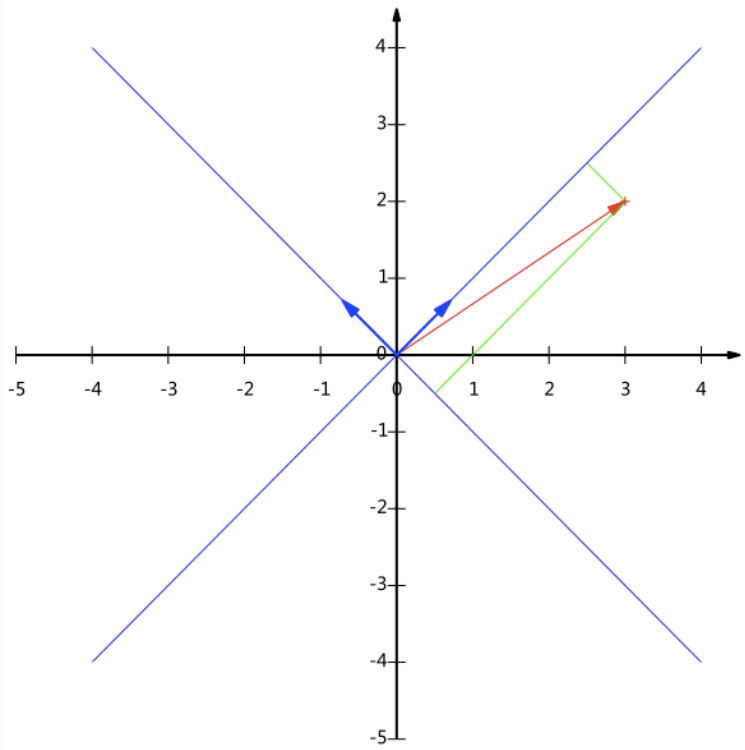
向量的内积操作

基向量

如果向量 B 的模为 1，则向量 A 和向量 B 的内积等于 A 向 B 所在直线投影的矢量长度。此时向量 B 也称为基向量。

内积与正交基

要准确描述向量，首先要确定一组基，然后给出在基所在的各个直线上的投影值。任何两个线性无关（不在同一直线上）的二维向量都可以成为一组基，但是我们通常选取正交的一组向量作为基，也称为正交基。



二维空间中的两组正交基

基变换

对于原空间的一个向量，如果用在使用新基表示的空间中重新描述该向量，只需将基向量对应的矩阵成一原向量，就可以作为基变换之后新的坐标。单向量的基变换：

$$\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 5/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$

多向量的基变换：

$$\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 2/\sqrt{2} & 4/\sqrt{2} & 6/\sqrt{2} \\ 0 & 0 & 0 \end{pmatrix}$$

基变换与数据降维

基变换可以将原空间中的数据，变换到新的空间中，并用新空间的基进行表示。（向量的维度有新空间中基的个数决定）

如果新空间的基的个数少于原空间中基的个数，那么基变换就实现了数据降维

什么是最优的数据降维？

高维空间中可以存在无数组基，对应无数种数据降维方式，但是哪一种数据降维方式是最优的呢？

主成分分析的两种目标

最大可分性

在正交属性空间中，存在这样的一个超平面，使得样本点在这个超平面上的投影尽可能分开

从信息熵的角度，数据足够分开，整个数据集包含的信息量越大。

最近重构性

在正交属性空间中，存在这样的一个超平面，使得样本点到超平面的距离足够接近

数据在属性维度的方差

给定某一属性维度，所有数据在这个维度上的投影分量组成了属性的向量表示。数据投影的离散程度可用这个该属性维度的方差表示：

属性维度的方差

给定一个 n 维的属性向量，其方差可以定义为：

$$\mathrm{Var}(a) = \frac{1}{n}\sum_{i=1}^{n}(a_i - \overline{a})^2 \tag{1}$$

如果该属性维度进行了标准化，则方差可以简化为：

$$\mathrm{Var}(a) = \frac{1}{n}\sum_{i=1}^{n}a_i^2 \tag{2}$$

属性的协方差

两个属性维度之间的相关性，通常用协方差来表示。

属性间的协方差

给定两个 n 维的向量，向量之间的协方差可以定义为

$$Cov(a,b) = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{n}$$

如果这两个向量进行了标准化，则协方差可以简化为：

$$Cov(a,b) = \frac{\sum_{i=1}^{n} a_i b_i}{n}$$

主成分分析优化目标

目标

给定一组 m 维向量降维 k 维（0<k<m），主成分分析目标是选择 k 个单位（模为 1）正交基，使得原始数据变换到这组基上后，各属性两两间协方差为 0，而字段的方差则尽可能大

一句话总结：在正交的约束下，取方差最大的 K 个属性维度

矩阵形式目标

上述优化目标，能否使用统一的形式进行表示？

矩阵形式目标

上述优化目标，能否使用统一的形式进行表示？
给定 n 个二维数据（向量）

$$X = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \end{pmatrix}$$

该矩阵乘以自身的转置为：

$$\frac{1}{n}XX^\top = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^n a_i^2 & \frac{1}{n}\sum_{i=1}^n a_i b_i \\ \frac{1}{n}\sum_{i=1}^n a_i b_i & \frac{1}{n}\sum_{i=1}^n b_i^2 \end{pmatrix}$$

一个矩阵同时包含了单个属性维度的方差（对角线元素）和不同属性维度之间的协方差（非对角线元素）

矩阵形式的主成分分析优化目标 1

矩阵形式优化目标

主成分分析的优化目标是使得降维之后的特征矩阵，其协方差矩阵为对角阵，即除对角线以外的所有元素均为 0。并且在对角线上将元素按照大小从上到下排列后，取前 k 个特征维度。

$$A = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$$

$$A = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n) \tag{3}$$

基变换与协方差矩阵推导

给定特征矩阵 X 以及对应的协方差矩阵 C，P 是一组基向量，Y 是 X 经过 P 对应的基变换得到的特征矩阵，其对应的协方差矩阵为 D，则 D 与 C 的关系如下：

$$
\begin{aligned}
D &= \frac{1}{n} Y Y^\top \\
&= \frac{1}{n} (PX)(PX)^\top \\
&= \frac{1}{n} PXX^\top P^\top \\
&= P\left(\frac{1}{n} XX^\top\right) P^\top \\
&= PCP^\top
\end{aligned}
$$

矩阵形式的主成分分析优化目标 2

矩阵形式优化目标

寻找一个矩阵 P，满足 $PCP^\top$ 是一个对角矩阵，并且对角元素按照从大到小依次排列，那么 P 的前 k 行就是要寻找的基。用 P 的前 k 行组成的矩阵乘以 X 就使得 X 从 m 维降到了 k 维并满足上述优化条件。

如何快速找到这样的矩阵呢？

实对称矩阵的数学性质

一个 n 行 n 列的实对称矩阵 C 一定可以找到 n 个单位正交特征向量，设这 n 个特征向量为 $e_1, e_2, ...e_n$，将其按照列组成矩阵：

$$E = \begin{pmatrix} e_1 & e_2 & \cdots & e_n \end{pmatrix}$$

该矩阵具有如下的性质：

$$E^\top C E = \Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$$

其对角元素为各特征向量对应的特征值。

主成分分析的矩阵形式

根据上述推导，我们可以得到主成分分析实现数据降维过程为：

1. 对所有样本进行中心化：

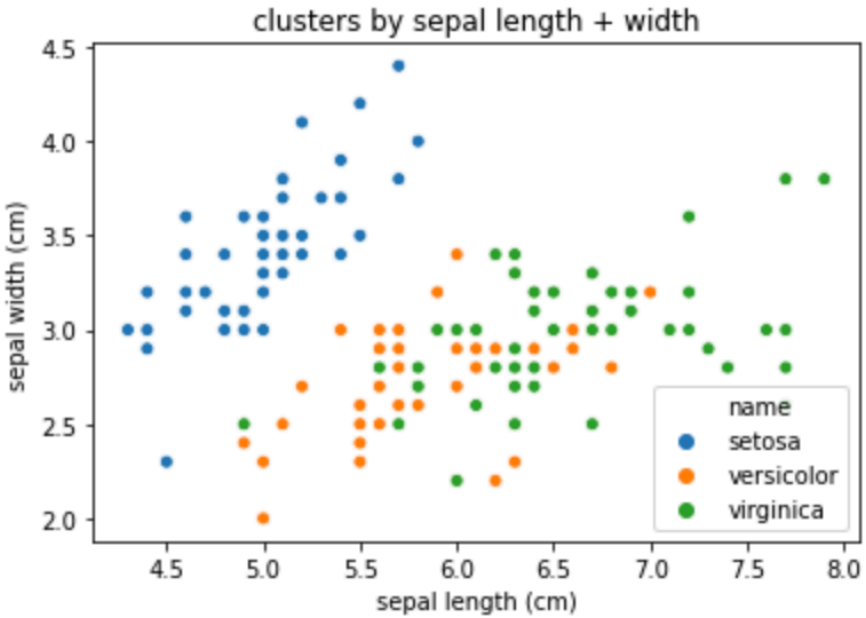
$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$$

2. 计算样本的协方差矩阵 $XX^\top$
3. 对协方差矩阵进行矩阵分解 $XX^\top$
4. 取 k 个最大的特征值对应的特征向量并组成矩阵

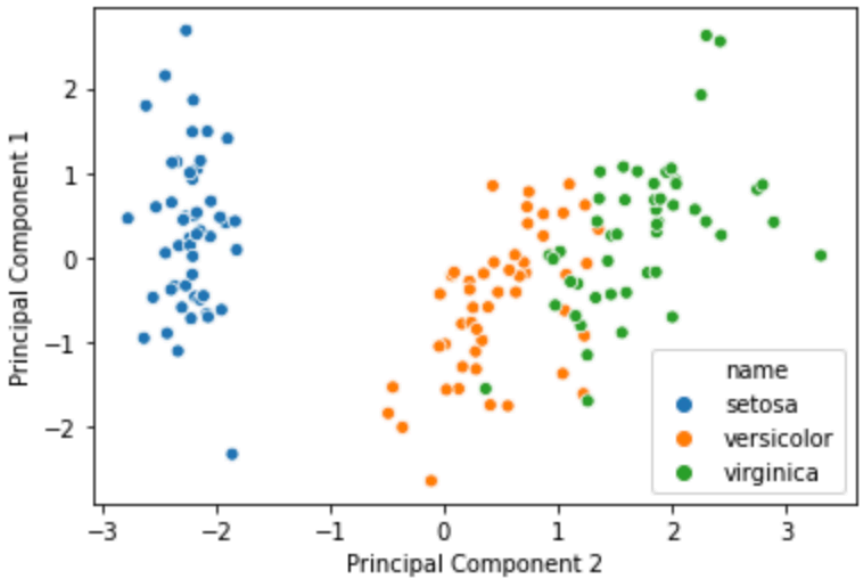
$$E = (e_1 \quad e_2 \cdots e_n)$$

PCA 实现

一行代码：sklearn.decomposition.PCA



原始特征

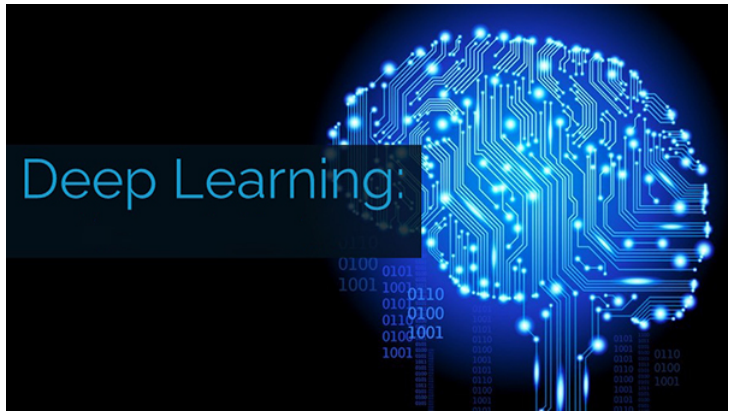


PCA 特征

PCA 的主要缺陷

PCA 是经典的数据降维方法，但是仍然面临如下缺陷：

1. 计算复杂度高
2. 维度诅咒
3. 只能处理特征向量数据



Deep Learning

经典的数据降维技术

经典的数据降维技术包括：

1. 主成分分析（Principle component analysis）
2. 等距特征映射 (Isometric Mapping)
3. 线性判别分析 (Linear Discriminant Analysis)
4. T-SNE (t-distributed stochastic neighbor embedding)

经典的深度学习降维技术包括：

1. Auto-Encoder
2. Deep Generative Model
3. Unsupervised Learning
4. Self-supervised Learning

要点总结

本次课程的核心知识点如下：

1. 数据挖掘的基本概念
2. 常见的数据形式
3. 常见的数据特征关系度量
4. 主成分分析

参考文献

1. 《数据挖掘》Jiawei Han etc.
2. 《机器学习》周志华
3. Blog:https://www.cnblogs.com/wj-1314/p/8032780.html
4. Blog:https://toutiao.io/posts/8r7zk9b/preview