第 1 讲　数据科学概述
第 2 讲　数据预处理
- 1.数据预处理
- 2.数据清洗
  - (1)缺失值处理
  - (2)噪声平滑
- 3.数据集成
- 4.数据归约
- 5.数据变换
  - (1)规范化
    - ①Min-Max 标准化(0-1 标准化)
    - ②小数定标标准化
    - ③Z-Score 标准化
    - ④Logistic 标准化
  - (2)数字编码
    - ①One-hot 编码
    - ②TF-IDF
  - (3)离散化
    - ①无监督离散化

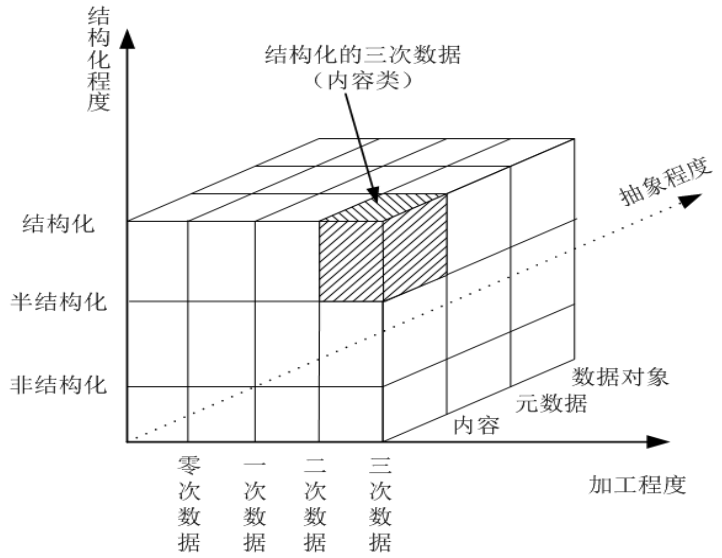
第 3 讲　回归与分类模型
第 4 讲　聚类模型
第 5 讲　模型的评价
第 6 讲　集成算法
*第 7 讲　自然语言理解与文本分析
*第 8 讲　社交网络分析

数据科学

概述

数据科学概述

1.数据的纬度



2.数据的结构化程度

类型	含义	本质	举例
结构化数据	**直接**可以用传统关系数据库存储和管理[1]	先有结构，后有数据	关系型数据库中的数据
非结构化数据	**无法**用关系数据库存储和管理的数据	没有（或难于发现）统一结构	语音，图像文件
半结构化数据	经过**一定的转换**[2]可以用传统关系数据库存储和管理的数据	先有数据，后有结构	HTML,XML 文件[3]等

[注 1]：传统的关系数据库：如 MySQL、Oracle。

为什么是“直接”：因为结构化数据具有明确定义的模式和规则，可以被分解为表格的形式，以便于使用 SQL 查询语言进行访问和操作。

[注 2]：经过解析和提取，将其中的关键信息转换为结构化数据的形式。

例如，对于 HTML 或 XML 文件，可以使用解析器将其转换为关系型数据库中的表格形式，将不同的标记或元素映射到表的列和行，并根据数据内容创建数据库模式。这种转换过程可以使用各种技术和工具进行，如 XPath、XSLT、正则表达式等。转换后的数据可以进行传统数据库操作，比如查询、插入、更新和删除等。

[注 3]：HTML(Hypertext Markup Language)是用于创建网页的标记语言，描述网页的结构和内容。XML(eXtensible Markup Language)是通用的标记语言，描述文档的结构和语义。

3.数据的加工纬度

零次数据：原始数据（没有经过预处理）　　一次数据：加工数据（预处理过的数据）

二次数据：增值数据（分析处理的结果）　　三次数据：洞见数据（直接可用于决策）

4.大数据

大数据依旧是数据，或数据相关的过程，其次，大数据的规模并非一定要达到某一确切的数值，关键在于，是否超过了实际情况下的数据存储能力和数据计算能力。

5.常见的数据类型

表数据：以**行和列**的形式组织的结构化数据。行代表**记录**（数据对象），列代表**属性**（对象特征）。比如：数据库表格、电子表格或 CSV 文件。是最为经典的数据。

点集：由一组**坐标或位置信息**组成的数据。比如：地理位置、坐标、传感器数据等。很多数据都可以看成是某种空间的点的集合。

时间序列：按**时间顺序**排列的数据集合，每个数据点都与特定时间点/时间段相关联。例如：温度数据、文本、通话、DNA 序列。常用于分析和预测趋势、周期性和季节性等。

图像视频：由像素组成的数据集。可以看成两个变量(空间位置和时间)的函数。

网页和报纸：非结构化或半结构化的数据。每篇文章都可以看成是时间序列，整个网页或报纸又具有空间结构。

网络数据：描述网络结构和连接关系的数据。网络数据本质上是图，由节点和联系节点的边构成，节点表示实体（如电脑、服务器、人物等），边表示节点之间的关系或连接（如通信链路、社交关系等）。

6.数据分析的主要困难

(1)数据量大

(2)维数高：模型复杂度和计算量随着维数的增加而指数增加。

将模型限制在一个极小的特殊类里面，如线性模型。

利用数据可能有的特殊结构，例如稀疏性，低秩，光滑性等，通过各类降维方法。

(3)类型复杂：结构化、半结构化、非结构化

7.算法处理大数据的两条思路

(1)降低算法的复杂度；

(2)分布式计算，分而治之，如著名的 MapReduce 框架

8.常用工具

R、Python、Clojure、Haskell、Scala 等数据科学语言工具；

NoSQL、MongoDB、Couchbase、Cassandra 等 NoSQL 工具

SQL、RDMS、DW、OLAP 等传统数据库和数据仓库工具；

HadoopHDFS&MapReduce、Spark、Storm 等支持大数据计算的工具；

HBase、Pig、Hive、Impala、Cascalog 等支持大数据管理、存储和查询的工具。

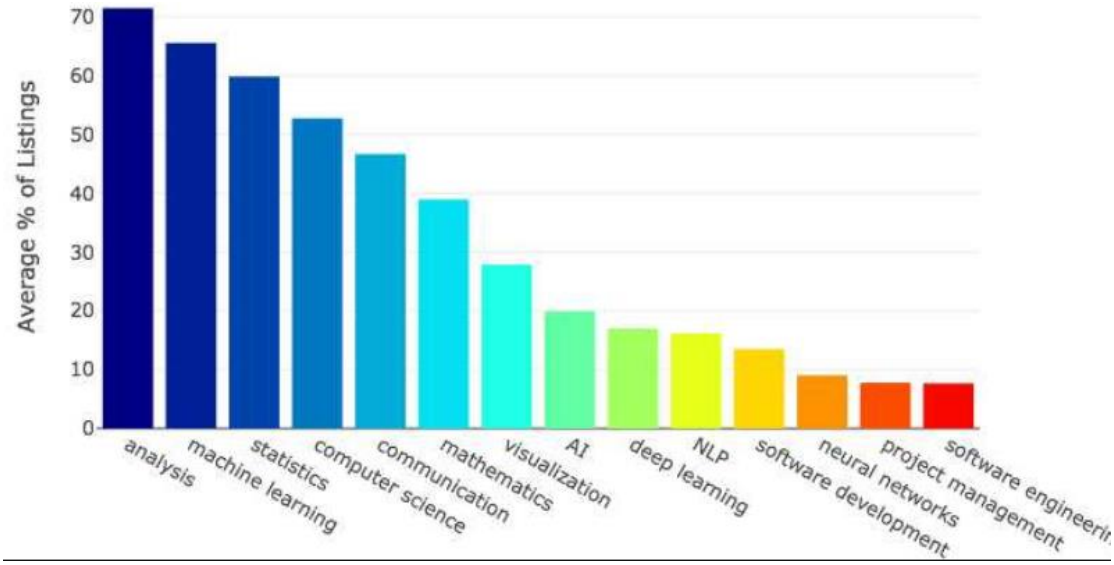
Webscraper、Flume Avro、Sqoop、Hume 等支持数据采集、聚合或传递的工具

Weka、Knime、RapidMiner、SciPy、Pandas 等支持数据挖掘的工具

ggplot2、D3.js、Tableu、Shiny、Flare、Gephi 等支持数据可视化的工具

SAS、SPSS、Matlab 等数据统计分析工具

9.general skill in data scientist job listings



数据

预处理

数据预处理

1.脏数据

不完整：缺少数据值；缺乏某些重要属性；仅包含汇总数据(原因:并未填写)
有噪声：包含错误或者孤立点(原因:故意输错;不愿提交)
不一致：在编码或者命名上存在差异，如：等级：1、2、3； A、B、C；甲、乙、丙
重复记录间的不一致性(函数依赖性)，如：Age="20" Birthday="03/07/1997"
(原因:数据传输错误;命名规范/格式不一致)

2.数据集

数据集由**数据对象**组成。一个数据对象代表一个实体，由**属性**来描述。

数据库的**行对应数据对象，列对应属性**。

属性

表示数据对象的特征。

在文献中，**属性**、**维度**（dimension）、**特征**（feature）、**变量**（variance）可以互换的使用。"维"：一般用在数据仓库中；"特征"：一般用在机器学习中；"变量"：一般用在统计学中。

标称属性：可枚举的(分类的)。用数来这些符号或名称，所以均值与中位数没有意义(但众数有意义)。为避免编码时产生不同的距离，可以这样编码:A(0,0,1) B(0,1,0) C(1,0,0)

二元属性：0/1，也称布尔属性。如性别。

序数属性：值是有意义的序，但相继值之间的差是未定义的。如大小、胜负、优良差。可以用众数中位数表示序数属性的中性趋势，但均值没有意义。

数值属性：不同于上面三个定性的量，数值属性是可度量的量，比如销量。

离散属性：有限个或者可数无限个，如 ID、颜色

连续属性：不是离散的属性，如成绩。与"数值属性"可以互换使用。

3.统计学规律

第一数字定律 Benford's Law：

在 b 进位制中，以数 N 起头的数出现的概率为

$P(N)=\log_b(N+1)-\log_b(N)=\log_b\frac{N+1}{N}$。

数字 N	1	2	3	4	5	6	7	8	9
以 N 为首位的概率%	30.1	17.6	12.5	9.7	7.9	6.7	5.8	5.1	4.6

适用条件：数据不能经过人为修饰；数据不能是规律排序，比如发票编号，身份证号码。

用途：可以用于判断数据是否**可能有问题**。

小概率原理 Law of Small Probabilities：

一个事件如果发生的概率很小的话，那么它在一次试验中是几乎不可能发生的，但在多次重复试验中几乎是必然发生的，数学上称之小概率原理。在统计学中，把小概率事件在一次实验中看成是实际不可能发生的事件,一般认为等于或小于0.05或0.01的概率为小概率。

齐夫定律 Zipf's law：

在自然语言的语料库里，一个单词出现的频率与它在频率表里的排名成反比。

用途：单词的出现频率、网页访问频率、收入前 3%的人的收入、地震震级、固体破碎

时的碎片大小。

4.语言学规律

连接性特征:

包括语言学中的后连接(如字母“q”后总是“u”)、前连接(如字母“x”的前面总是字母“i”,字母“e”很少与“o”和“a”连接)以及间断连接(如在“e”和“e”之间,“r”的出现频率最高)。

重复特征:

两个字符以上的字符串重复出现的现象,叫做语言的重复特征。例如在英文中字符串“th”、“tion”和“tious”的重复率很高。

5.数据预处理的主要任务

(1)描述性数据归总

理解数据的基本情况,发现异常值、检查数据可靠性和一致性

数据清洗:填写空缺的值,平滑噪声数据,识别、删除孤立点,解决不一致性

数据集成:集成多个数据库、数据立方体或文件

(2)数据变换和标准化

规范化和聚集,提高涉及距离度量的挖掘算法的准确性和有效性

数据归约:得到数据集的压缩表示,它小得多,但可以得到相同或相近的结果

数据离散化:数据归约的一部分,通过概念分层和数据的离散化来规约数据,对数字型数据特别重要

(3)特征的选择与降维

选择最相关的特征,以提高模型效率,减少数据的复杂性

①特征选择:从所有特征中选择最具有预测能力的特征子集,以提高模型的准确性、降低过拟合的风险、减少计算成本。

过滤法:通过计算特征和目标变量之间的相关性或统计指标来进行特征选择。

包裹法:使用模型训练和评估的结果来进行特征选择。

嵌入法:将特征选择嵌入到机器学习算法的训练过程中,通过正则化或特征权重更新来选择重要的特征。

②降维:减少数据的维度,以消除冗余和噪声,并简化模型的复杂性。

主成分分析 PCA:通过线性变换将原始特征映射到新的正交特征空间,以捕捉数据中的主要方差。

线性判别分析 LDA:是一种有监督的降维方法,它通过最大化类别之间的方差和最小化类别内部的方差来找到具有最佳分类能力的特征投影。

特征抽取:一种非线性的降维方法,它利用神经网络或自编码器等技术将原始特征映射到低维空间。

描述性数据

1.数据度量的方法

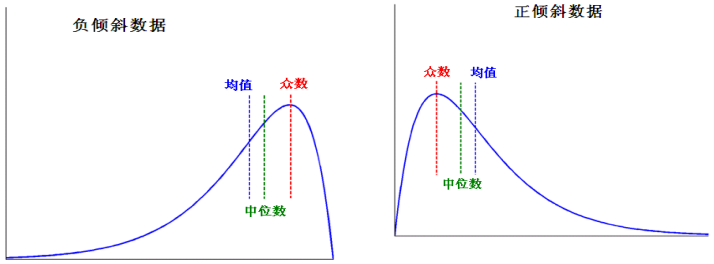
度量数据的**中心趋势**：均值、中位数、众数、中列数
度量数据的**离散程度**：四分位数、四分位数极差、方差

2.截断均值

去掉高、低极端值得到的均值。
如：计算平均工资时，可以截掉上下各 2%的值后计算均值，以抵消少数极端值的影响。

3.中心趋势度量

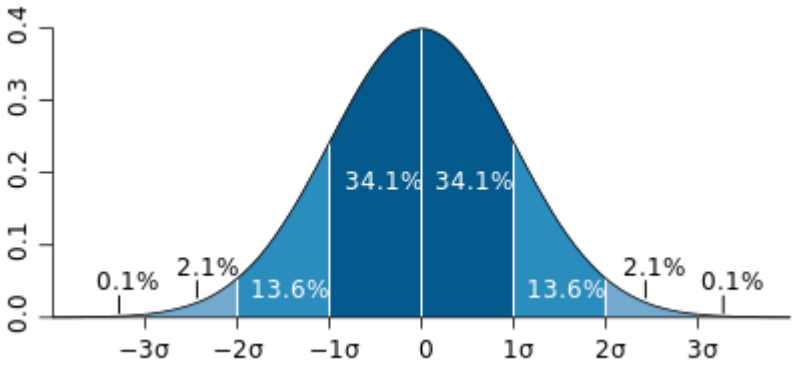
左右对称：均值=中位数=众数　**负倾斜**：均值<中位数<众数　**正倾斜**：众数<中位数<均值



众数与中位数的距离约为中位数与均值距离的 **2** 倍。
某个异常值对**均值**的影响最大(均值敏感)

4.正态分布函数曲线 $N\sim(\mu,\sigma^2)$

正态分布 Normal distribution 也叫高斯分布 Gaussian distribution
$(\mu-\sigma,\mu+\sigma):86.27\%$，$(\mu-2\sigma,\mu+2\sigma):95.45\%$，$(\mu-3\sigma,\mu+3\sigma):99.73\%$



缺失值处理

1.处理方法

(1)删除带有缺失值的样本或特征:

删除样本：删除存在数据缺失的样本。该方法适合某些样本有多个特征存在缺失值，且存在缺失值的样本占整个数据集样本数量的比例不高的情形。

删除特征：当某个特征缺失值较多，且该特征对数据分析的目标影响不大时，可以将该特征删除。

(2)采用某种方法对缺失值进行填补:

均值填补：**连续型特征，平均值填补；离散型特征，众数填补。**但会降低方差、相关性。如果数据是倾斜的，应该使用中位数，或者同一类样本的中位数。

适用于对个体精度不大的数据。

实际可以根据一定的辅助特征，**先将数据集分组**，然后在每一组数据上分别使用均值插补，比如学生信息数据集，"入学年份"特征记录了学生入学的年份，可以首先根据毕业年份对数据进行分组，然后使用每一个分组数据中"年龄"特征的平均值来对缺失值进行填补。

随机填补：随机填补是在均值填补的基础上加上随机项，通过增加缺失值的随机性来改善缺失值分布过于集中的缺陷。

贝叶斯 Bootstrap 方法：设数据集有 n 个样本，某特征 f 存在 $n-k$ 个缺失值。

①先从均匀分布 $U\sim(0,1)$ 中随机抽取 $k-1$ 个随机数并升序排序为 $\{0,a_1,\cdots a_{k-1},1\}$。

②对 $n-k$ 个缺失值，分别从非缺失值 $\{f_1,f_2,\cdots,f_k\}$ 中以概率 $a_1-0,a_2-a_1,\cdots,1-a_{k-1}$ 采样一个值进行填补。

近似贝叶斯方法：

①从非缺失值 $\{f_1,f_2,\cdots,f_k\}$ 中有放回地抽取 k 个值建立一个新集合 F。

②对 $n-k$ 个缺失值，分别从 F 中随机抽取一个值进行填补。

基于模型的填补：由非缺失值构建训练集，训练模型来预测缺失值。比如根据父母身高推测丢失的子女身高。实际使用时需要采用模型评估方法对模型的预测性能进行评估，如果构建的模型预测性能太差，则不适合使用该方法。此外，基于模型的填补方法将增大特征之间的**相关性**。

其他填补法：

全局常数(哑变量方法)：

将缺失的属性值用同一个常数(如 Unknown 或 NaN)替换。但比如将性别特征的缺失值作为一个特殊的取值"unknown"，表示性别未知。此时认为"性别"特征包含"F"、"M"和"unknown"三个不同取值。

EM 算法：

利用不完整的信息实现概率模型的参数化估计的算法。该算法可以用来进行缺失值填补，此时缺失特征被当作隐含变量。

2.文献阅读：

(1)缺失数据，保罗·D.埃里森 林毓玲译出版社:格致出版社出版，2018.06

(2)基于缺失值变量的不完整数据填补与分类研究，硕士学位论文，大连理工大学，吴霞，2020.

噪声平滑

1.噪声 Noise

噪声是被测量的变量的随机误差或方差，包括错误的值或偏离期望的离群点。

2.数据平滑

分箱法：

思想：考查数据的近邻来平滑有序数据值。是**局部平滑**。

方法:将数据划分成几个含有相同个数数据的数据段,再将箱中的**每个值用某个值替换**。一般而言，箱的宽度**越大**，平滑效果**越明显**。

某个值的选择：**均值、中位数、边界**（将箱中的最大/小值试作边界，每个值被最近的边界值替换）。

方法与理论知识：

移动平均 Moving Average：计算数据点周围一定窗口大小内的均值来减少噪声的影响。

中值滤波 Median Filtering：非线性噪声平滑，使用数据点周围窗口内的中值来替代当前数据点的值。常用于去除椒盐噪声(图像中的随机黑点和白点)等脉冲噪声。

加权平均 Weighted Averaging：分配不同权重，通常在窗口中心的数据点权重较高，而边缘的权重较低。

指数平滑 Exponential Smoothing：较新的数据点权重较高，较旧的权值较低。常用于时间序列数据(用于捕捉趋势和季节性变化)。

例子：设有数据，升序排列后为 4,8,15,21,21,24,25,28,34，将其划分为三个箱子。
用箱均值平滑：箱 1：9,9,9。箱 2：22,22,22。箱 3：29,29,29。
用箱边界平滑：箱 1：4,4,15。箱 2：21,21,24。箱 3：25,25,34。
用箱中值平滑：箱 1：8,8,8。箱 2：21,21,21。箱 3：28,28,28。

回归法：

用一个函数拟合数据来平滑数据。

离群点分析：

通过聚类来检测离群点，将落在簇集合之外的值视为离群点。

3.异常值的检测(识别离群值)

简单统计：

判断变量的取值是否在合理的范围（如年龄，负数或者 200 岁肯定是异常值）。

3σ 原则：

参见正态分布。

箱型图 Tukey's test：

百分位数 percentile：第 k 个百分位数是具有如下性质的值 x：k%的数据项位于或低于 x，中位数就是第 50 个百分位数

四分位数：

Q1 (下四分位数，也称 QL，25th percentile，第(n+3)/4 个数)

Q3 (上四分位数，也称 QU，75th percentile，第(3n+1)/4 个数)

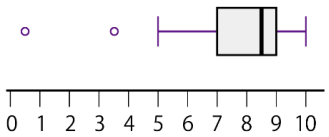
中间四分位数极差/四分位间距(interquartile range，IQR)： IQR = Q3 – Q1

孤立点/离群值(outlier)：落在至少高于第三个四分位数或低于第一个四分位数 1.5×IQR 处的值。位于范围外 1.5×IQR 到 3×IQR 范围的数值，称作适度离群值（mild outlier）。位于范围外 3×IQR 以上的数值，称作极端离群值（extreme outlier）。

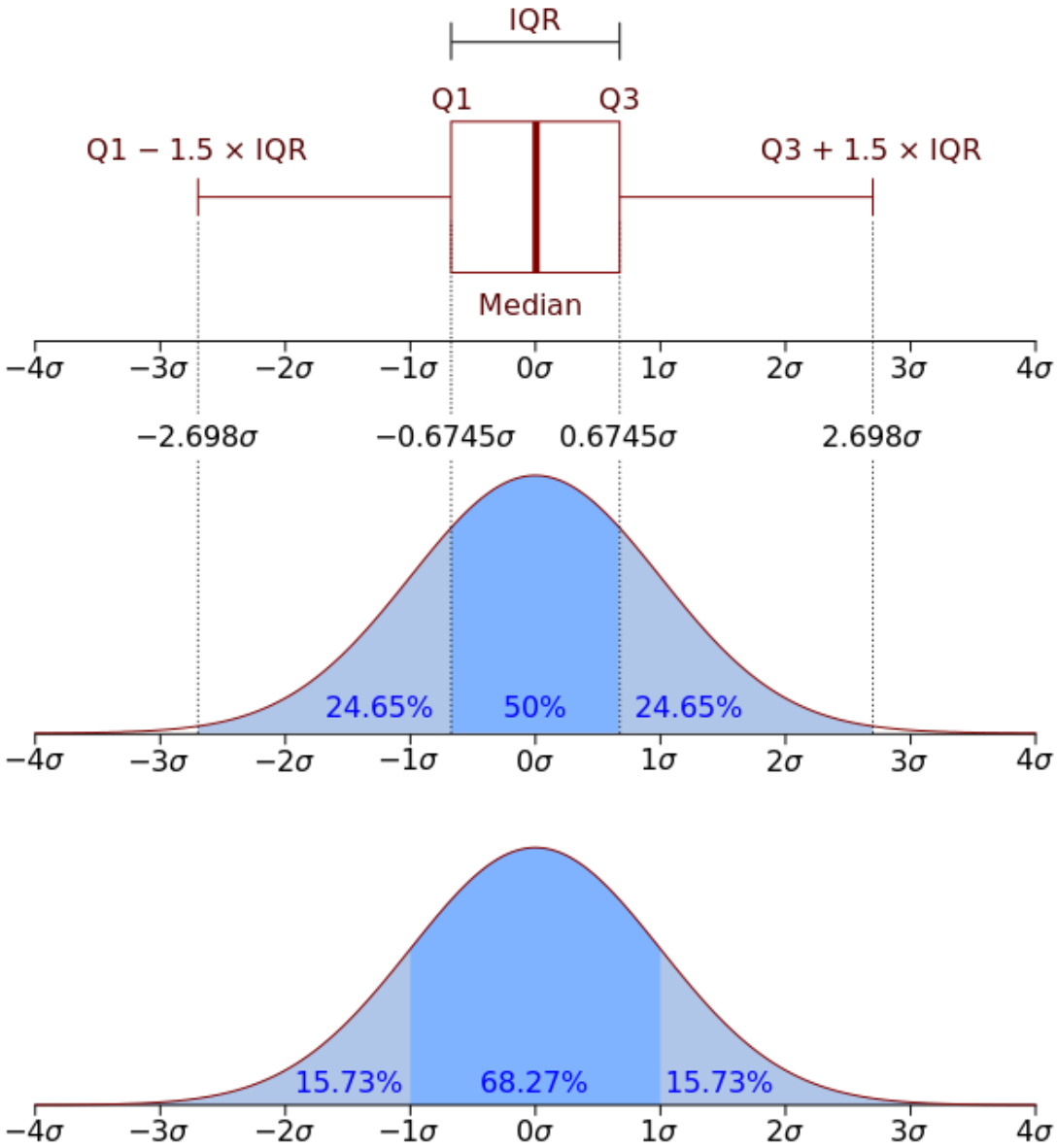
上下限：上限=min{Q3+1.5IQR，MAX}，下限=max{Q1-1.5IQR，MIN}。

换句话说，上下限是忽略掉异常值（包括温和异常值）之后的最大值和最小值。

如右图，下边界=5，Q1=7，Q2=8.5，Q3=9，上边界=10，IQR=Q3-Q1=2(即△Q)。适度离群值=3.5，极端离群值=0.5。



箱型图与正态分布：



基于模型检测：

如果是建立数据模型，异常值是不能完美拟合模型的对象。

如果模型是簇的集合，异常值是不显著属于任何簇的对象。

如果是使用回归模型，异常值是相对远离预测值的对象。

优势：数据充分、检验知识完备的情况下。

劣势：对于多元数据、高维数据的性能不好。

基于距离：

定义邻近性度量，异常对象就是那些远离其他对象的对象。

优点：简单。

缺点：时间复杂度 $O(n^2)$ 且 对参数的选择敏感 且 不能处理不同密度区域的数据集(因为它使用的是全局阈值，不能考虑密度的变化)

基于密度：

当一个点的局部密度显著低于它的大部分近邻时才将其分类为离群点。

优点：适合非均匀分布的数据。给出了离群点的定量度量。即使数据具有不同的区域也能很好处理。

缺点：时间复杂度 $O(n^2)$ 且 参数选择困难。

基于聚类：

当对象不强属于任何簇时，它是基于聚类的离群点。

优点：线性复杂度。可同时发现离群点和簇。

缺点：非常依赖所用的簇的个数和数据中离群点的存在性。簇的质量对离群点的质量影响非常大。

4.异常值的处理

1.删除异常值
2.视为缺失值处理
3.用均值修正
4.不处理

数据集成

1.数据冲突

背景：

信息系统独立。数据源分布异构等原因形成了“信息孤岛”。

数据集成就是合并来自多个数据源的数据。

产生原因：

(1)同名异义：比如数据源 A 中的属性 ID 与 B 中的 ID 不是一个 ID(描述不同的实体)。

(2)异名同义：比如数据源 A 中的 sales_dt 与 B 中的 sales_data 是一样的。

解决方法：

利用**元数据** metadata（data about data）来解决。

每个属性的元数据包括名字、含义、数据类型、取值范围、处理空值的规则、其他描述信息等。以学生信息数据集为例：

属性可以是学生的姓名、年龄和性别。

元数据则可以包括属性的定义、数据类型(字符串、整数等)、取值范围(如年龄 1~100)、单位(如年龄用岁表示)以及其他描述信息(如年龄是指学生的当前年龄，而非出生年月日)。

也就是说，属性是数据实际具有的特征或特性，而元数据是描述这些属性的数据。

2.冗余数据检测——相关性分析

冗余数据：属性 B 可以由属性 A 导出，则属性 B 可能是冗余的。(比如出生导出年龄)

对于**标称数据**，可以用 χ^2 卡方检验。

对于**数值属性**，可以用相关系数 Correlation Cofficient 和协方差 Covariance。

χ^2 相关性检测(卡方检验)：

设 A 有 c 个不同值 $a_1,a_2,\cdots,a_c$ 构成列，B 有 r 个不同值 $b_1,b_2,\cdots,b_r$ 构成行。记 A 的第 i 个值与 B 的第 j 个值构成的联合事件为 (A_i,B_j)。

$\chi^2=\sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(A_{ij}-T_{ij})^2}{T_{ij}}$，式中 A_{ij} 是联合事件 (A_i,B_j) 的观测频度，即实际计数 actual，T_{ij} 是 (A_i,B_j) 的期望频度 Theoretical，即(A 中 a_i 的个数×B 中 b_j 的个数)÷n。然后依据自由度与显著性水平查表确定拒绝域的临界值（自由度=(r-1)×(c-1)，显著性水平常取 α=0.05）

例如右图，a,b,c,d 为观测频度，括号内为期望频度。

性别\视觉	正常	色盲	总和
男	$a\left(\frac{(a+c)(a+b)}{n}\right)$	$b\left(\frac{(b+d)(a+b)}{n}\right)$	$a+b$
女	$c\left(\frac{(a+c)(c+d)}{n}\right)$	$d\left(\frac{(b+d)(c+d)}{n}\right)$	$c+d$
总和	$a+c$	$b+d$	n

$$\chi^2=\sum_{i=1}^{2}\sum_{j=1}^{2}\frac{(A_{ij}-T_{ij})^2}{T_{ij}}\sim\chi^2(2)$$

协方差(Covariance)与相关系数(Pearson 积矩系数)：

协方差 $\mathrm{Cov}(A,B)=E((A-\overline{A})(B-\overline{B}))=\frac{\sum_{i=1}^{n}(a_i-\overline{A})(b_i-\overline{B})}{n}$，其中 $\overline{A},\overline{B}$ 相当于 EA,EB，公式简化后为 $\mathrm{Cov}(A,B)=E(AB)-E(A)E(B)$。

相关系数 $\rho_{A,B}=\frac{\mathrm{Cov}(A,B)}{\sigma_A\sigma_B}$。$\rho_{A,B}=0$ 表示 A,B 不相关(但可能存在线性关系以外的其他函数关系)，$\rho_{A,B}>0$ 表示 A,B 正相关。

相关性，描述两个随机变量是否存在线性关系。**独立性**，则考察的是两个随机变量是否存在某种关系。因此，**相互独立事件必然不相关，反过来却不一定成立**。此外，若随机变量 X,Y 服从二元高斯分布，则二者不相关与独立互为充要条件。相关性不蕴含因果性。

三种相关系数的公式计算可参考 https://zhuanlan.zhihu.com/p/431865174。

协方差的理解可参考 https://www.zhihu.com/question/20852004 中 GRAYLAMB 的回答。

元组重复：

对于给定的唯一数据实体，存在两个或多个相同的元组。

去规范化表(denormalized table)的使用（这样做通常是通过避免连接来改善性能，创建的是宽表）是数据冗余的另一个来源。不一致通常出现在各种不同的副本之间，由于不正确的输入，或者由于更新了部分相关数据，但是未更新所有的相关数据。

小结：

数据冗余的可能情况：

1.同一属性多次出现或仅是命名前后不一致。

2.A 属性能由 B 导出。

数据归约

1.三种策略

数据归约： 用数据集的规约表示，通过比原数据集小很多的数据产生几乎相同的分析结果。有三种方法：

维归约 Dimensionality Reduction：减少数据集的**特征维度**。将原始数据中的冗余或不重要的特征删除，同时保留对数据中重要信息的捕获。

数量归约 Numerosity Reduction：减少数据集中的行数或**样本数量**。在不损失太多信息的情况下，减少数据集的规模，以加快分析和处理的速度。

数据压缩 Data Compression：使用**压缩算法**来减小数据集的大小，同时尽量保留数据的信息。压缩方法的思想是用更紧凑的表示方式来存储数据，从而减小存储空间。

2.维归约

主成分分析 Principal Component Analysis(PCA)：又称 K-L 方法，搜索 k 个最能代表数据的 n 维正交向量($k \le n$)，使得数据投影到一个小得多的空间上(新的坐标系)，实现维规约。

过程：

①对输入数据 $X=\{x_1,x_2,\cdots,x_n\}$ 规范化 $x_i'=x_i-\frac{\sum_{j=1}^{n}x_j}{n}$，使得每个属性都落入相同的区间(避免较大定义域的属性不会支配较小定义域的属性)

②计算协方差矩阵 $\frac{XX^T}{n}$，求特征值与特征向量，选择最大的 k 个特征值(作为规范化输入数据的基)。然后将这 k 个特征值对应的 k 个特征向量 $w_{1\sim k}$ 标准化后分别作为列向量组成特征向量矩阵 W 。

③将数据映射到 k 个特征向量构建的新空间中 $z_i=W^Tx_i$，输出数据 $X'=(z_1,z_2,\cdots,z_n)$ 。

主成分到底怎么理解：

主成分其实就是数据的**新坐标系**，提供关于方差的信息。第一个轴显示的数据方差最大，第二个次之，如此下去。

适用情景：

有序、无序的属性，稀疏、倾斜数据等。将多维数据归约为二维数据。作为多元分析和聚类分析的输入。

特征选择 Feature Selection：选择原始数据集中的部分特征而不创建新的特征空间。常用的方法包括基于统计测试的方法、递归特征消除 RFE。

选择方法：

暴力选择需要 $O(2^n)$，这里有四种方法：

逐步向前选择：从空属性集开始，每次选择当前最好的属性(使用统计显著性测试)。

逐步向后删除：从全集开始，每次删除当前最坏的属性。

逐步向前选择和逐步向后删除**结合**：每一步选择一个最好的属性并删除一个最坏的属性。

决策树归纳：使用决策树生成的内部节点表示属性(树叶对应类预测)。

线性判别降维法 Linear Discriminant Analysis(LDA)：找到在不同类别之间具有最大可分性的投影方向。不同于 PCA 方差最大化理论，LDA 算法希望将数据投影到低维空间之

后，**同一类数据尽可能的紧凑**，不同类的数据尽可能分散。

3.数量归约

随机抽样：从数据集中随机选择样本，以减少数据的大小。还可以分层抽样。
聚类采样：将相似的数据点分组在一起，然后通过单个代表性数据点表示每个组
过滤异常值：移除数据中的异常值，以减小数据集的噪音和离群点。

4.数据压缩

有损压缩：有损压缩方法会删除一些数据，以减小文件大小，但可能损失一部分信息。这种方法常用于图像、音频和视频压缩，例如 JPEG 图像压缩。

无损压缩：无损压缩方法可以压缩数据，但在解压后不会损失任何信息。这种方法用于需要完整保存数据的场合，例如 ZIP 压缩。

数据转换编码

1.Min-Max 标准化(0-1 标准化)

通过线性变换 $x_i^* = \frac{x_i - \min}{\max - \min}$ 使处理过后数据均落在 $[0,1]$ 区间内。

通过线性变换 $x_i^* = \frac{x_i - \frac{\max + \min}{2}}{\frac{\max - \min}{2}}$ 使处理过后数据均落在 $[-1,1]$ 区间内。

适用范围：

0-1 标准化适用于需要将数据简单地变换映射到某一区间中，但其不足之处在于当有新数据加入时，可能会导致数据系列中的最大值或最小值发生变化，此时便需要重新定义最大值、最小值(也就是需要重新标准化)，否则就会**越界**。

如何把某数据集映射到指定的区间 $[a,b]$：$x_i^* = (b-a)\frac{x_i - \min}{\max - \min} + a$

2.小数定标规范化

通过移动数据的小数点位置来进行标准化，小数点移动多少位取决于数据系列中的最大绝对值大小：$x_i^* = \frac{x_i}{10^j}$（j 是使得 $\max(|x_i^*|) < 1$ 的最小整数）

适用于数据系列分布比较**离散**，尤其是数据系列**遍布多个数量级**的情况，若数据系列分布集中在某几个量级上，则小数定标标准化的数据也会集中在某个值附近，不利于建模时的数据区分。优点在于简单实用，在确定了小数点移动位数后，易于还原标准化后的数据系列。

3.Z-Score 标准化

最常用的标准化方法，使得处理后的数据具有固定均值 0 和标准差 1(但不一定是正态)。

基于平均值与标准差规范化：$x_i^* = \frac{x_i - \mu}{\sigma}$。其中 $\mu = \frac{\sum_{i=1}^{n} x_i}{n}, \sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}}$

当数据中存在离群点时，为了降低离群值的影响，将 σ 替换成平均绝对差 $s = \frac{\sum_{i=1}^{n}|f_i - \mu|}{n}$

$|x_{Z-Score}| > 3$ 的样本认为是离群值。

经过 Z-Score 标准化后的数据，能直观反映每个数据点距离平均值点的标准差距离，从而理解整体数据的分布情况。每个数据点离零点的距离可解释为其远离均值的标准差距离。

适用范围：数据系列中最大值或最小值未知、数据分布非常离散(离群点左右了 Min-Max 标准化)的情况。

应用场景：通过将数据标准化到同一比例，可以更好地比较不同变量之间的关系，或者比较来自不同分布的数据。设 A 的满分是 100 分，B 的满分是 700 分，显然 A 考 70 分与 B 考 70 分代表的意义完全不同。因此需要 Z-Score(一个同等的标准)来比较 A 与 B 的成绩。

4. Logistic 标准化

利用 Sigmoid 函数(也称 Logistic 函数) $x_i^* = \frac{1}{1+e^{-x_i}}$，将原始数据系列转化为[0,1]之间。

适用范围： Logistic 标准化方法适用于数据系列分布相对比较集中地分布于零点两侧，否则容易被压缩到两端。

5.四种方法的对比与一些其他方法

对比表格：

方法	优点	缺点	适用范围
0-1	对数据作线性变换，保留数据的原始关系	若原数据最大最小值发生变化需重新定义	需**保留原始数据间的关系**，且最大/小值已经确定
小数定标	简单实用、易于还原标准化后的数据	若原数据最大绝对值发生变化需重新定义	数据系列分布比较离散，数据**遍布多个数量级**
Z-Score	转化为标准正态分布无序数据的最大、最小值	需要记录原数据均值方差	数据中最大最小值未知且数据系列**分布离散**
Logistic	简单，通过单一映射函数对数据进行标准化	对分布零散且远离零点的数据处理效果不佳	数据系列分布比较集中，且均分布于**零点两侧**

其他方法：

1.均值归一法 Mean normalization：

① $x_i^* = \frac{x_i - \mu}{\max}$　　② $x_i^* = \frac{x_i - \mu}{\max - \min}$　　式中 $\mu = \frac{\sum_{i=1}^{n} x_i}{n}$

2.向量归一化：

$$x_i^* = \frac{x}{\sum_{i=1}^{n} x_i}$$

3.指数转换：

除了 Logistic 外还有：

①lg 函数：$x_i^* = \frac{\lg x_i}{\lg \max}$，要求数据都大于 1，结果就会落入[0,1]区间内。

②softmax 函数：$x_i^* = \frac{e^x}{\sum_{i=1}^{n} e^{x_i}}$

6.数字编码

数据源：UCI 的 Wine Data Set。http://archive.ics.uci.edu/ml/datasets/Wine

与 https://archive.ics.uci.edu/ml/datasets/car+evaluation

如：“汽车品牌”={路虎，吉利，奥迪，大众，奔驰}，数字编码后转换成“汽车品牌”={0，1，2，3，4}。“收入水平”={贫困，低收入，小康，中等收入，富有} 数字编码后转换成“收入水平”={0，1，2，3，4}。

存在的缺陷： 会引入错误的序；会引入错误距离

7.One-hot 编码

又称一位有效编码/独热编码，其方法是使用 N 位状态寄存器来对 N 个状态进行编码，每个状态都有他独立的寄存器位，并且在任意时候，其中只有一位有效(为 1)，其余全为 0。可以这样理解，对于每一个特征，如果它有 m 个可能值，那么经过独热编码后，就变成了 m 个二元特征，并且这些特征互斥，每次只有一个激活，因此，数据会变成稀疏的。

如：美国(0,0,1)英国(0,1,0)法国(1,0,0)

优点：

不会给名义型特征的取值人为地引入次序关系。

经过 One-Hot 编码之后，不同的原始特征取值之间拥有相同的距离。

在线性回归模型中，对名义型特征进行 One-Hot 编码的效果通常比数字编码的效果要好。

One-Hot 编码对包含离散型特征的分类模型的效果有很好的提升。

缺点：

特征维度会显著增多：假设存在 10 个包含 100 个取值的离散型特征，经过 One-Hot 编码后的特征数量将变成 1000 个。

增加特征之间的相关性：如美英法编码后，三个特征存在线性关系：$f_1+f_2+f_3=1$。(所以应该做出改变：对于一个包含 K 个取值的离散型特征，将其转换成 K-1 个二元特征，即哑变量编码(dummy encoding)，如

One-hot：

原始特征取值	f_1	f_2	f_3	f_4	f_5
路虎	1	0	0	0	0
吉利	0	1	0	0	0
奥迪	0	0	1	0	0
大众	0	0	0	1	0
奔驰	0	0	0	0	1

哑变量：

原始特征取值	f_1	f_2	f_3	f_4
路虎	1	0	0	0
吉利	0	1	0	0
奥迪	0	0	1	0
大众	0	0	0	1
奔驰	0	0	0	0

)

一个简单的例子如下，编码前后如图：

	Feature_1	Feature_2	Feature_3
Sample_1	1	4	3
Sample_2	2	3	2
Sample_3	1	2	2
Sample_4	2	1	1

	Feature_1	Feature_2	Feature_3
Sample_1	01	1000	100
Sample_2	10	0100	010
Sample_3	01	0010	010
Sample_4	10	0001	001

特征向量就为：

Sample_1 = [0,1,1,0,0,0,1,0,0]，Sample_2 = [1,0,0,1,0,0,0,1,0]，Sample 3,4 同理。

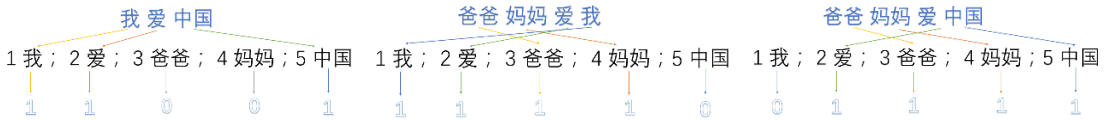
这有什么用？一是解决了分类器不好处理离散数据的问题，二是在一定程度上也起到了扩充特征的作用（上面样本特征数从 3[3 个 Feature]扩展到了 9[9 个向量特征])。编码是为了分类，虽然这些 Feather 是数字，看似有大小关系，其实只是一种类别，因此可以编码为独立的特征向量。

One-hot 在文本特征提取上属于词袋模型 bag of words，以以下语料库为例：

我爱中国 爸爸妈妈爱我 爸爸妈妈爱中国

编码：1 我 2 爱 3 爸爸 4 妈妈 5 中国

提取特征向量：



因此我们得到了最终的特征向量为

我爱中国 ＝1,1,0,0,1　　爸爸妈妈爱我 ＝1,1,1,1,0　　爸爸妈妈爱中国 ＝0,1,1,1,1

One-hot 的用处：**特征提取**。

另一种特征提取方法为 **TF-IDF**：

IF-IDF 是信息检索(IR)中最常用的一种文本表示法。统计每个词出现的词频 TF (Term Frequency)，然后乘上权值参数 IDF (逆文档频率 Inverse Document Frequency)。举个例子：

比如统计一篇文档中的前 10 个关键词。首先想到词频越高，这个词就越重要。但关键词最终基本都是“的”、“是”这样没有实际意义的词(停用词)，因此应该加权值：

TF＝ 某个词出现次数/文章总词数，IDF＝log[语料库的文档总数/(包含该词的文档数+1)]

此算法无法处理一词多义与一义多词的情况。

8.离散化

算法对数据特征类型通常是有要求的：

关联规则算法只能处理布尔类型的数据，决策树算法只能处理特征为离散型的数据。很多算法能处理连续性特征，但是离散化处理能使得部分模型产生更好的预测效果。

将连续性特征转换成为离散型特征的过程称为**特征离散化** data discretization。

将连续性特征的取值范围划分为若干区间段 bin,用区间段代替落在该区间段的特征取值，区间段之间的分割点称为切分点 cut point，分割出来的区间段的个数称为元数 arity。

如何在数据信息损失尽量少的前提下，尽可能减少元数，是数据离散化追求的目标，所以如何选择切分点，产生合理的子区间段成为决定特征离散化成败的**关键**。

离散化步骤：

①特征**排序**：对连续型特征的取值进行升序或者降序排列，减少离散化的运算开销。

②**切分点选择**：根据给定评价准则(比如基于信息增益或者基于统计量)选择切分点。

③区间段**分割或合并**：基于选择好的切分点，对现有区间段进行分割或合并，得到新的区间段。

④在生成的新的区间段上重复以上步骤，直到满足终止条件。

离散化方法：

自顶向下的离散化方法：等距离散化、等频离散化、基于信息增益的离散化方法。

自底向上的离散化方法：基于卡方的离散化方法。

无监督的离散化方法：等距离散化、等频离散化、基于聚类分析的离散化方法。

有监督的离散化方法：基于信息增益的离散化方法、基于卡方的离散化方法。

等距离散化：

将属性的值域$[Min, Max]$划分成 K 个区间，每个区间长度相等(均为 $\omega = \frac{Max - Min}{K}$)。

特点：适用于数据分布均匀的情况，尤其是在没有先验知识的情况下。对数据质量要求高，对离群值敏感。

等频离散化：

将属性的值域$[Min, Max]$划分成 K 个区间，每个区间所含的数据个数相等(均为 $\frac{N}{K}$，其中 N 为取值总数)。

基于聚类：

对需要离散化的连续性特征，采用聚类算法把样本依据该特征划分成相应的簇或者类。在聚类的结果上，基于特定的策略，决定是否对簇进行进一步的分裂或合并，利用自顶向下的策略针对每一个簇继续运行聚类算法，细分成为更小的子簇，或者利用自底向上的策略对相邻的簇进行合并。在最终确定划分簇后，确定切分点及区间个数。

特点：相似的样本能落到相同的区间段内，可以更好地代表原始数据的信息。

基于信息增益：

该方法源自于决策树模型，在建立决策树时，遍历每一个特征，选择熵最小(信息增益最大)的特征作为正式分裂节点。

①对连续型特征进行排序。

②把特征的每一个取值作为候选分裂节点(切分点)，计算出相应的熵，选择熵最小的取值作为正式的切分点，将原来的区间一分为二。

③递归处理第二步中得到的两个新区间段，直到每个区间段内特征的类别一样为止。

④合并相邻的，类的熵值为 0 且特征类别相同的区段，重新计算新区间段类的熵值。

⑤重复第四步到满足终止条件(决策树的深度或叶子数)。

基于卡方：

卡方检验属于非参数检验范畴，是一种比较两个总体之间是否存在显著性差异的方法：$\chi^2=\sum_{i}^{k}\frac{(A_i-E_i)^2}{E_i}$，其中 A_i 为落入区间段的样本个数(观察频数)，E_i 为对应的期望频数，在 n 较大的情况下，χ^2 统计量近似服从自由度为 $k-1$ 的 χ^2 分布。

自下而上的卡方离散化——**ChiMerge** 方法：

①将连续型特征的每一个取值看作是一个单独的区间段，并按照数值大小进行排序。

②对每对相邻的区间段计算卡方统计量。卡方值最小或低于设定阈值的相邻区间段合并。

③对于新的区间段，递归进行第 1,2 步，直到满足终止条件。

特点：Chimerge 每次迭代的过程只能合并两个区间，如果数据量大，则算法的开销会比较大。需要设定显著性水平，然后计算卡方统计量，通过阈值的设定来控制区间数的多少。

自上而下的卡方离散化：

①将连续型特征的所有数据点看作是一个大区间段，并按照数值大小进行排序。

②对每对相邻的数据点计算卡方统计量。卡方值最大或高于设定阈值的相邻区间段合并。

③对于新的区间段，递归进行第 1,2 步，直到满足终止条件。

卡方值 Chi-square value = Σ [(观察频数-期望频数)^2 /期望频数]：

$\chi^2=\sum_{i=1}^{2}\sum_{j=1}^{C}\frac{(A_{ij}-E_{ij})^2}{E_{ij}}$，其中 A_{ij} 为第 i 区间段内类别为 j 的样本个数，k 为比较的区间个数，C 为类别个数，$E_{ij}=\sum_{j=1}^{C}A_{ij}\bullet\frac{\sum_{i=1}^{k}A_{ij}}{n}$。

类别属性依赖最大化：

类别属性依赖最大化 CAIM(Class-attribute interdependence Maximization)是一种基于熵的特征离散化方法。

与 ChiMerge 不同，它采取自顶向下的策略。通过选择切分点 p，把特征的取值空间划分为 $f\le p$ 和 $f>p$ 两个子区间段，用来衡量切分点选择优劣的度量方法是类别属性的相互依赖程度。

假设某个连续型特征有 n 个取值，C 个类别. 假设我们把特征划分为 k 个子区间段，子区间段集合记为：$D=\{[d_0,d_1],(d_1,d_2],\cdots,(d_{k-1},d_k]\}$，其中 d_0 和 dlk 分别为特征的最小值和最大值，$n_{i\cdot}$ 表示属于类别 i 的样本个数，$n_{\cdot j}$ 表示落在区间段 $(d_{j-1},d_j]$ 的样本个数，n_{ij} 表示在区间 $(d_{j-1},d_j]$ 内的且属于类别 i 的样本个数。我们可以得到一个由类别特征和离散化特征取值所构成的二维表：

类别	$[d_0,d_1]$	$(d_1,d_2]$	$\cdots$	$(d_{k-1},d_k]$	类别样本数
1	n_{11}	n_{12}	$\cdots$	n_{1k}	$n_{1\cdot}$
2	n_{21}	n_{22}	$\cdots$	n_{2k}	$n_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C	n_{C1}	n_{C2}	$\cdots$	n_{Ck}	$n_{C\cdot}$
区间样本数	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot K}$	n

评价当前离散化的好坏的评价准则 $CAIM=\dfrac{\sum_{j=1}^{K}\frac{M_j^2}{n_{\cdot j}}}{N}$。其中 $M_j=\max\{n_{1j},n_{2j},\cdots,n_{Cj}\}$。

$CAIM$ 的取值为区间(0, 1]。$CAIM$ 的值越大，说明类和离散区间的相互依赖程度越大，也就说明了离散化效果越好。

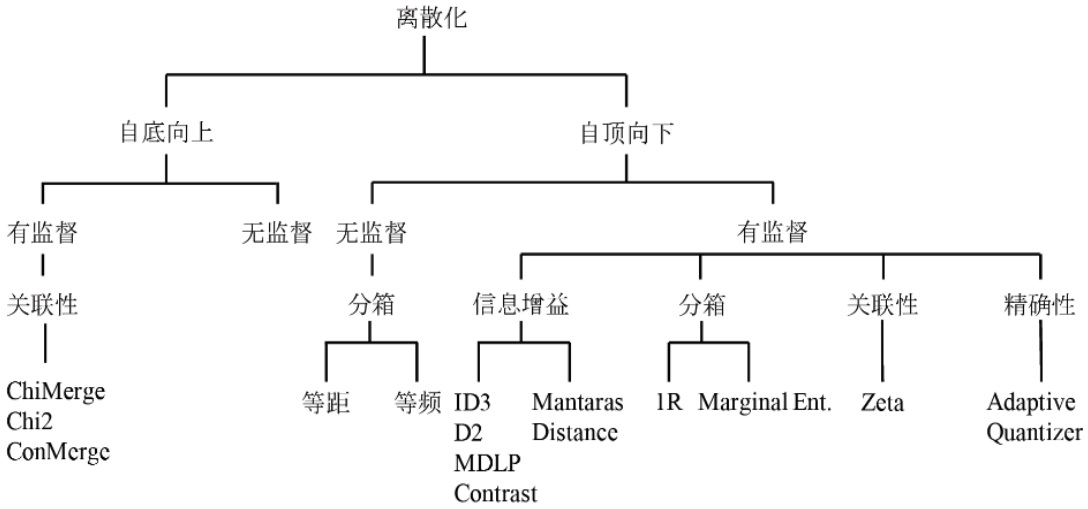
①对进行离散化的特征进行升序排列，确定取值区间的最小值 d_0 和最大值 d_k，初始化划分策略 $D=[d_0,d_k]$。

②把区间内的每个值当作候选切分点，计算把区间二分后的 $CAIM$ 值，并选取 $CAIM$ 值最高的点作为切分点，并更新 D。

③对于 D 中的每个区间段，重复第二步的计算，直到满足终止条件。

特点：$CAIM$ 只关注区间内拥有最多样本的类别与特征之间的关系，忽略同一区间内其他类别的信息， $CAIM$ 最终生成的离散化区间个数往往与样本的类别个数接近。

小结：



文献阅读：

【1】Salvador Garcia, Julian Luengo, etc. “A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning”. In: IEEE Transaction on knowledge and data engineering. 25(4), 2013, pp. 734-750.

【2】Lukasz A. Kurgan and Krzysztof J. Cios. “CAIM discretization algorithm”. In: IEEE transactions on Knowledge and Data Engineering 16.2 (2004), pp. 145–153.

【3】Francis E.H. Tay, “A modified Chi2 Algorithm for Discretization”, IEEE Transactions on knowledge and data engineering, 14(3), 2002.

【4】桑雨. 连续数据离散化方法研究. [博士学位论文]，大连理工大学

9.标称数据的概念分层

便于将标称数据变换到多个粒度层。

以地址这个标称属性集为例，地址这个集合包含{区,市,省,国家}4 个标称属性，但没有相关数据语义的知识，未说明其偏序，如何找出任意的标称属性的分层序呢?

启发规则:**较高概念分层的属性,通常包含较少的不同值**。国家这一属性包含的不同值的个数和省或者州包含的不同取值数比较，前者个数显然较少。

回归与分类

模型

先谈机器学习

一、机器学习

1.要素

任务 T(Task)：对数据集 M 分类，确定给定的某个数据 M_i 的类别。

性能指标 P(Performance)：分类的准确率。

经验来源 E(Experience)：大量的数据以及其对应的类别。

2.应用

数据挖掘：统计分析、分析评估。

计算机视觉：图像处理。

自然语言处理 NLP：文本分类与聚类、信息检索与过滤、机器翻译。

生物特征识别：人脸识别。

3.术语

数据集 Data Set 里的每条数据称为一个**样本** Sample 或者**示例** Instance。样本信息的结果(所属类别)被称为**标记** Label。

数据对象的**特征** Feature 和**属性** Attribute 是一个含义。属性上的值称为**属性值**。

因为一个样本按照其 N 个属性建立 N 维空间，那么每一个样本都能在该空间中找到自己的坐标向量，所以样本也被称为**特征向量** Feature Vector。

从数据中学得模型的过程称为**学习**或**训练**，每个样本被称为**训练样本**，训练样本组成的数据集称为**训练集**。

分类 Classification：预测的目标属性是离散的。**回归/预测** Regression：预测的目标属性是连续的。**聚类** Clustering：将数据集分成若干个组(簇 Cluster)。

分为**监督学习** Supervised Learning(分类回归)和**无监督学习**(归纳性学习)Unsupervised Learning(聚类降维)。所谓“监督”，就是预设的规则/数据的标签。

4.模型评估

模型的目的是为了**泛化** Generalization，也就是模型适应新样本的能力。由于使用训练集评估模型会严重高估模型的准确率，因此需要将数据集 D 划分为训练集 S 和测试集 T。

(1)留出法 Hold-Out：

正样本 Positive Sample：在二元分类问题中，正样本通常代表我们感兴趣的类别或事件。例如，在一个医疗诊断时正样本可以代表患有某种疾病的患者，而在垃圾邮件过滤器时正样本可以代表垃圾邮件的样本。

负样本 Negative Sample：在二元分类问题中，负样本代表与正样本相反的类别或事件。例如，负样本可以代表没有患有疾病的患者，非垃圾邮件。另外，比如检测人脸时，人脸是正样本，非人脸则是负样本，比如旁边的树、花之类的其他东西。

在多类别分类问题中，可以将其中一个类别定义为正样本，而其他类别被认为是负样本。

留出法将 D 划分为两个互斥的集合 S,T(如 70%S,30%T)，且对正负样本采用分层抽样。

(2)交叉验证法 Cross Validation(k 折交叉验证法)：

将包含 m 个样本的 D 划分为 k 个大小相似的互斥子集，每个子集 $D_i(i=1\sim k)$均通过分层抽样得到，使用 k-1 个子集的并集作为 S，余下的一个子集作为 T。依次方法进行 k 次训练和测试，最终返回值是这 k 次测试结果的均值。

当 k=1，交叉验证法退化为留出法。当 k=m，每个子集只有一个样本数据，就称为留一法 Leave-One-Out(LOO)，评估准确但训练 m 个模型的开销大。因此一般取 k=10。

(3)自助法 Bootstrap：

每次随机从 D 中挑选一个样本复制到 D'中，重复 m 次。得到的 D'包含 m 个样本，但约有 36.8%(1/e)的样本未在 D'中出现，这部分作为 T，D'中的数据作为 S。

方法	采样方法	和原始数据分布	相比原始数据容量	适用数据集	估计偏差
留出	分层抽样	不相同	变小	大	存在
交叉	分层抽样	不相同	变小	大	存在
自助	放回抽样	不相同	不变	小	存在

5.性能度量

(1)在分类中：

精度 Accuracy：正确分类数/全部分类数。**错误率** Error Rate：错误分类数/全部分类数。

混淆矩阵 Confusion Matrix：

预测结果 / 真实数据	正样本	负样本
正样本	TP	FN
负样本	FP	TN

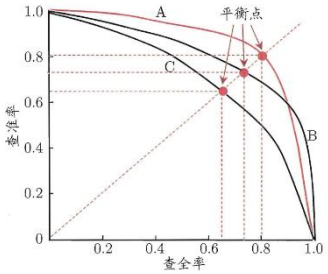
真正 True Positive：被预测为正的正样本。真负 True Negative：被预测成负的负样本。
假正 False Positive：被预测为正的负样本。假负 False Negative：被预测成负的正样本。

$$\text{精度 A}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FN}+\text{FP}+\text{TN}}$$

查准率 Precision：$\text{P}=\frac{\text{TP}}{\text{TP}+\text{FP}}$，分类成正样本的确实是正样本的正确率。

查全率 Recall：$\text{R}=\frac{\text{TP}}{\text{TP}+\text{FN}}$，将所有的正样本正确找出来的比例。

显然 P 和 R 不可兼得，如商品推荐算法更多考虑准，逃犯信息检索更多考虑全。P-R 曲线如右图。还有平衡点、ROC 曲线、AUC 等指标。



以信息检索为例，刚开始在页面上显示的信息是用户可能最感兴趣的信息，此时查准率高，但只显示了部分数据，所以查全率低。随着用户不断地下拉，信息符合用户兴趣的匹配程度逐渐降低，查准率不断下降，查全率逐渐上升。当下拉到信息底部时，此时的信息是最不符合用户兴趣，因此查准率最低，但所有的信息都已经展示，查全率最高。

F1 度量：$\text{F1}=\frac{2\text{PR}}{\text{P}+\text{R}}$，即加权调和平均 $\frac{2}{\frac{1}{\text{P}}+\frac{1}{\text{R}}}$，这使得 F1 接近于 P,R 中更小的那个，所以当 P,R 接近时 F1 值最大。加权调和平均的例子如去程速度 30 返程速度 10 求平均速度。

(2)在回归中：

设数据集 $D=\{x_1,\cdots,x_n\}$ 的预测标签 $\hat{Y}=\{\hat{y}_1,\cdots,\hat{y}_n\}$ 对应的数据标签是 $Y=\{y_1,\cdots,y_n\}$。

平均绝对误差 Mean Absolute Error(MAE)/L1 范数损失：$\text{MAE}=\frac{\sum_{i=1}^{n}|y_i-\hat{y}_i|}{n}$。

均方误差 Mean Squared Error(MSE)/L2 范数损失：$\text{MSE}=\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{n}$。

均方根误差 $\text{RMSE}=\sqrt{\text{MSE}}$

决定系数 $R^2=1-\frac{\sum_{i=1}^{n}(\hat{y}_i-y_i)^2}{\sum_{i=1}^{n}(\bar{y}_i-y_i)^2}$，取值范围[0,1]，越大说明拟合效果越好。但随着样本数

量增加，R^2 亦会增加。于是引入**校正决定系数** $\text{Adjusted } R^2 = 1-\frac{(1-R^2)(n-1)}{n-p-1}$，其中 n 为样本数量，p 为特征数量。

(3)在聚类中：

外部指标(参考)：

将聚类结果与某个“**参考模型**”(如领域专家的划分结果，比如对数据进行标记)进行比较。默认参考模型的性能指标是对样本的最优划分，度量的目的就是使聚类结果与参考模型尽可能相近，也就是聚类结果中被划分在同一簇样本与参考模型样本也被**同样划分**到一个簇的概率越高越好。

给定数据集 $D=\{x_1,\cdots,x_n\}$ 经过聚类划分的簇为 $C=\{C_1,\cdots,C_k\}$，参考模型给出的簇划分为 $C^*=\{C_1^*,\cdots,C_s^*\}$。设 $\boldsymbol{k},\boldsymbol{k}^*$ 为数据在 C,C^* 中的簇标记向量，定义：

$a=|\text{SS}|,\text{SS}=\{(x_i,x_j)|\boldsymbol{k}_i=\boldsymbol{k}_j,\boldsymbol{k}_i^*=\boldsymbol{k}_j^*,i<j\}$，　$b=|\text{SD}|,\text{SD}=\{(x_i,x_j)|\boldsymbol{k}_i=\boldsymbol{k}_j,\boldsymbol{k}_i^*\neq\boldsymbol{k}_j^*,i<j\}$，

$c=|\text{DS}|,\text{DS}=\{(x_i,x_j)|\boldsymbol{k}_i\neq\boldsymbol{k}_j,\boldsymbol{k}_i^*=\boldsymbol{k}_j^*,i<j\}$，　$d=|\text{DD}|,\text{DD}=\{(x_i,x_j)|\boldsymbol{k}_i\neq\boldsymbol{k}_j,\boldsymbol{k}_i^*\neq\boldsymbol{k}_j^*,i<j\}$。

其中 S,D 分别表示数据隶属相同/不同簇。集合 SS 表示在 C 中隶属相同簇并且在 C^* 中仍隶属相同簇的样本对。a,b,c,d 是对应集合中样本对的个数。由于每对样本只能出现在 4 个集合中的一个，因此 $a+b+c+d=\frac{n(n-1)}{2}$。

Jaccard 系数： $\text{JC}=\frac{a}{a+b+c}$，适合二分类问题。

FM 指数： $\text{FMI}=\sqrt{\frac{a}{a+b}\cdot\frac{a}{a+c}}$，结合了 JC 与 RI 的特点。

Rand 指数： $\text{RI}=\frac{2(a+d)}{n(n-1)}$，适合多分类问题。

以上指标取值范围均为$[0,1]$，取值越大代表聚类效果越好。

内部指标(距离)：

直接考察聚类结果，通过计算簇内样本间距与簇间样本距离来评估模型的性能。也就是用簇内样本间距模拟簇内相似度，用簇间样本距离模拟簇间相似度。定义：

$\text{avg}(C)=\frac{2}{|C|(|C|-1)}\sum_{1\le i<j\le|C|}\text{dist}(x_i,x_j)$，　$\text{diam}(C)=\max_{1\le i<j\le|C|}\text{dist}(x_i,x_j)$，

$d_{\min}(C_i,C_j)=\min_{x_i\in C_i,x_j\in C_j}\text{dist}(x_i,x_j)$，　$d_{\text{cen}}(C_i,C_j)=\text{dist}(u_i,u_j)$。

其中 dist 用于计算两个样本之间的距离。$u_i=\frac{\sum_{1\le i\le|C_i|}x_i}{|C_i|}$ 代表簇 C_i 的中心点。avg 用于计算对应簇内样本间的平均距离。diam 用于计算对应簇内样本间的最远距离。$d_{\min}$ 用于计算两簇间样本的最近距离。d_{cen} 用于计算两簇间中心点的距离。

DB 指数： $\text{DBI}=\frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{avg(C_i)+avg(C_j)}{d_{\text{cen}}(C_i,C_j)}\right)$

Dunn 指数： $\text{DI}=\min_{1\le i\le k}\left\{\min_{j\neq i}\left(\frac{d_{\min}(C_i,C_j)}{\max_{1\le x\le k}\text{diam}(C_x)}\right)\right\}$

前者越小越好，后者越大越好。

分类与回归

说在前头：

分类是一种有监督学习问题，主要目标是**根据已知的类别信息（标签）来训练模型，以预测新数据点的类别**。比如信用风险评估:根据用户历史还款信息预测未来是否违约；电子邮件:根据邮件内容将邮件归类(正常邮件/垃圾邮件)。

决策树 Decision Tree，*K* 近邻算法 *K*-Nearest Neighbor(KNN)，朴素贝叶斯 Naive Bayes，支持向量机 Support Vector Machine(SVM)。

回归目标属性 y 是连续的，**预测数据的趋势**，回归任务同样需要带有标签的训练数据，但标签是连续数值。比如依据父母身高预测孩子的。

线性回归 Linear Regression，非线性回归 Nonlinear Regression，Logistic 回归，Lasso 回归，岭回归 Ridge Regression。

聚类是一种无监督学习任务，其主要目标是**将数据集中的数据点划分为不同的簇，试图发现数据中的内在结构**。通常只需原始数据，不需要标签信息。比如依据兴趣、行为等属性对社交媒体用户分群。(注意这里“分类”的依据是已知的“属性”，而不是已知的“标签”)。

K-means 聚类 K-means Clustering，层次聚类 Hierarchical Clustering，密度聚类 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)。

1.分类-决策树

决策树是一种分类方法，从一组无次序无规则的元组中推理出决策树表示形式。采用自顶而下的递归方式，在决策树内部节点进行属性值的比较，进行向下分支，叶子是要学习划分的类。

常用算法：CLS，ID3，C4.5，CART(Classification And Regression Tree)，SLIQ(Super-vised Learning In Quest)，SPRINT(Scalable Parallelizable Induction of Decision Trees)。

https://zhuanlan.zhihu.com/p/517743953

1966 CLS(学习单个概念)→1979 ID3(选择**信息增益**作为特征选择方法)→1986 ID4(在每个可能的决策树节点创建缓冲区，使决策树可以递增式生成，即在不断增加数据的情况下持续构建决策树)→1988 ID5(提高效率)→1993 C4.5(选择**信息增益率**作为特征选择方法，同时处理连续和离散特征，还能处理缺失值)

另一类决策树算法为 CART，与 C4.5 不同的是，CART 的决策树由二元逻辑问题生成，每个树节点只有两个分枝，分别包括学习实例的正例与反例。CART 这种树构建算法既可用于分类也可用于回归。

递推过程：划分数据集时寻找起决定性作用的特征。

决策树的特点：1.推理过程容易理解，决策推理过程可以表示成 If Then 形式。2.推理过程完全依赖于属性变量的取值特点。3.可能忽略目标变量没有贡献的属性变量，这也为判断属性变量的重要性，减少变量的数目提供参考。

CLS 算法　Concept Learning System

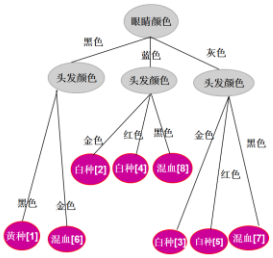
CLS 是早期的决策树学习算法，它是决策树学习算法的基础。

算法思想：

从一棵空决策树开始，选择某一属性(分类属性)作为测试属性，该测试属性对应决策树中的决策结点，根据该属性的值的不同将训练样本分成相应的子集：

如果该子集为空，或该子集中的样本属于同一个类，则该子集为叶结点；否则该子集对应于决策树的内部结点，需要选择一个**新**的分类属性对该子集进行划分，**直到所有的子集都为空或者属于同一类**。如右图。

人员	眼睛颜色	头发颜色	所属人种
1	黑色	黑色	黄种人
2	蓝色	金色	白种人
3	灰色	金色	白种人
4	蓝色	红色	白种人
5	灰色	红色	白种人
6	黑色	金色	混血
7	灰色	黑色	混血
8	蓝色	黑色	混血



算法问题：

测试属性集的组成以及测试属性的先后对决策树的学习具有举足轻重的影响。

ID3 算法　Iterative Dichotomiser 3

算法思想：

从根节点开始，选择“**某**”属性特征(如收入)。选择属性特征的策略：**不纯度** impurity。即落在当前节点的样本类别分布的均衡程度，节点分裂后，节点不纯度应该更低。选择特征及对应分割点，使得分裂前后的不纯度下降最大。

节点不纯度的度量

(1)信息熵 Entropy：描述信息不确定度：$H(x)=-\sum p_i \log_2(p_i)$，其中 p_i 为 $P(X=x_i)$。当样本均匀分布在每一个类中时，熵为 $\log_2 C$，说明不纯度大；当所有的样本属于同一个类时，熵为 0，说明不纯度小。

由 $H(x,y)=H(x)+H(y), P(x,y)=P(x)P(y)$，那么 $H(x)$ 一定与 $P(x)$ 的对数有关。

为什么有一个负号：负号是为了确保信息一定是正数或者是 0，总不能为负数吧！

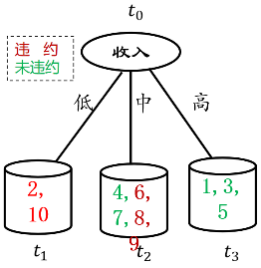
为什么底数为 2：只需要信息量满足低概率事件对应于高的信息量。那么对数的选择是任意的。我们只是遵循信息论的普遍传统，使用 2 作为对数的底！

例子如右图，

$$H(t_1)=-\frac{2}{2}\log_2\frac{2}{2}-\frac{0}{2}\log_2\frac{0}{2}=0$$

$$H(t_2)=-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}=0.971$$

$$H(t_3)=-\frac{0}{3}\log_2\frac{0}{3}-\frac{3}{3}\log_2\frac{3}{3}=0$$

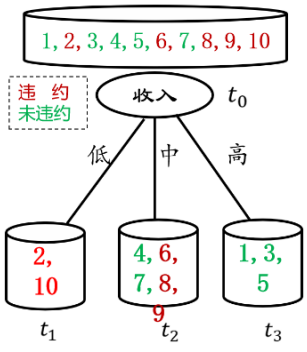


条件熵 Condition Entropy：$H(Y|X)=\sum p_i H(Y|X=x_i)$

(2)信息增益 Information Gain：节点分裂前后信息熵的下降值(得知某特征 A 而使得分类 X 的信息的不确定性减少的程度)：$\text{Gain}(A)=H(X)-H(X|A)$。其中 $H(X)$ 即 $H(t_0)$，$H(X|A)$ 即 $\sum_{k=1}^{k}\frac{n_k}{n}H(t_k)$，$n_k$ 是子节点 t_k 的样本数量，n 是父节点 t_0 的样本数量。信息增益越大，说明该特征对结果的影响越大。

例子如右图，$H(t_0)=-\frac{5}{10}\log_2\frac{5}{10}-\frac{5}{10}\log_2\frac{5}{10}=1$，$H(t_{1,2,3})$ 如前计算，

$$\text{Gain}(\text{收入})=H(t_0)-[\frac{2}{10}H(t_1)+\frac{5}{10}H(t_2)+\frac{3}{10}H(t_3)]=0.5145$$



(3)信息增益率 Gain Ratio： $\text{Gain}_{\text{ratio}}(A)=\dfrac{\text{Gain}(A)}{H(A)}$

(4)基尼指数 Gini Index： $\text{Gini}_{\text{index}}(A)=\sum_{v=1}^{n}\dfrac{|X_v|}{|X|}\text{Gini}(X_v)$，式中 $\text{Gini}(X)=1-\sum_{k=1}^{m}p_k^2$ 。

$\text{Gini}(X)$ 反应了数据集 X 中随机抽取两个样本，其类别不一致的概率。当样本均匀分布在每一个类中，$\text{Gini}(X)=1-\dfrac{1}{m}$（$m$ 为该类的样本个数），说明不纯度大。当样本都分布在同一个类中，$\text{Gini}(X)=0$，说明不纯度小。

$\text{Gini}_{\text{index}}(A)$ 就是对于特征 A 下集合的基尼指数，其中 A 的取值共有 n 种可能，X_v 表示对数据集 X 按照特征 A 划分的数据子集。比如这里特征就是 t_0(收入)，划分为 $t_{1,2,3}$ 共 $n=3$ 个子集。

例子仍同前，

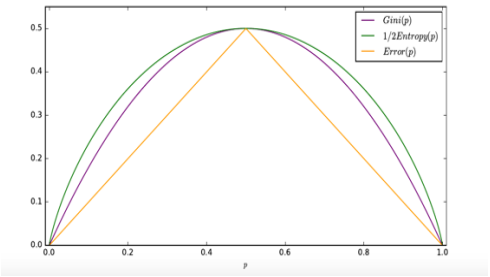
$$\text{Gini}(t_1)=1-[(\frac{2}{2})^2+(\frac{0}{2})^2]=0 \qquad \text{Error}(t_1)=1-\max(\frac{2}{2},\frac{0}{2})=0$$

$$\text{Gini}(t_2)=1-[(\frac{3}{5})^2+(\frac{2}{5})^2]=0.480 \qquad \text{Error}(t_2)=1-\max(\frac{3}{5},\frac{2}{5})=0.4$$

$$\text{Gini}(t_3)=1-[(\frac{0}{3})^2+(\frac{3}{3})^2]=0 \qquad \text{Error}(t_3)=1-\max(\frac{0}{3},\frac{3}{3})=0$$

ID3 采用信息增益作为特征选择的方法(倾向于分裂成很多小节点「节点的样本数较小」，容易造成过拟合)。C4.5 采用信息增益率(减少信息增益准则对取值数目较多的属性有所偏好，避免分裂成过多小节点)。CART 采用基尼指数(选择基尼指数最小的属性进行划分)

(5)误分率 Misclassification error： $\text{Error}(t)=1-\max\{p_k\}(k=1,2,\cdots,m)$ 。当样本均匀分布在每一个类中，误分率为 $1-\dfrac{1}{m}$，不纯度最大。当样本分布在同一个类中，误分率为 0，不纯度最小。例子见上文 Gini 后面的 Error。



(6)决策树生成算法的小结：

对于二分类，如果正类样本的比例为 p，则有

$H(t)=-p\log_2 p-(1-p)\log_2(1-p)$

$\text{Gini}(t)=2p(1-p)$ ，如右图

$\text{Error}(t)=1-\max\{p,1-p\}$

信息熵最大，说明对不纯度的惩罚最强。

算法	特征类型	不纯度度量	分割的子节点数量	目标特征类型
ID3	离散型	信息熵(信息增益)	$K\geq 2$	离散型
C4.5	离散型、连续型	信息熵(信息增益率)	$K\geq 2$	离散型
C5.0	离散型、连续型	信息熵(信息增益率)	$K\geq 2$	离散型
CART	离散型、连续型	Gini 指数	$K=2$	离散型、连续型

决策树生成后转换成 IF-THEN 规则的集合，使得决策树模型的可解释性较好。

(7)决策树生成的方法小结：

```
def CreateTree():
    if 该子集为空或该子集中的样本属于同一个类：  # 则该子集为叶结点
        return 类标签
    else:  # 该子集是内部结点，需要进行划分。直到所有的子集都为空或属于同一类
        计算信息增益，确定最好特征。划分数据集，创建分支节点。
        for 每个节点：
```

```
            CreateTree()并将返回结果添加到分支节点。
        return 分支节点
```

(8)决策树的剪枝：

预剪枝：在 CreateTree 时就停止信息量较小[即：信息增益(率)较小]的分支。

后剪枝：先 CreateTree，搞完了再剪枝。

(9)决策树的优缺点：

优点：直观。允许存在缺失值。对离群点不敏感。可用于复杂的非线性关系。

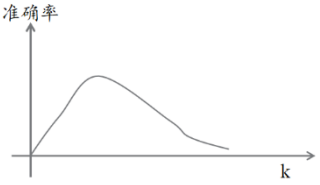
缺点：决策树由于使用贪婪算法的思想，即在每次分裂时选择当前情况下带来信息增益最高的属性进行分裂，容易陷入**局部最优解**。如果缺少合适的剪枝策略或终止分支处理，容易过拟合，比如以 ID 来决策，但无效。

(10)补充：德国信用数据集：

https://archive.ics.uci.edu/ml/datasets/South+German+Credit+%28UPDATE%29

2. 分类-k 邻近算法

当对测试样本进行分类时，找到训练集中与该样本集最相似的 k 个样本，根据 k 个样本的标签(选取其中最多的)确定测试样本的标签。



准确率与 k 的函数一般成单峰。二分类问题中 k 一般设为奇数，防止出现两种类型出现次数相同的情况。k 一般小于 30。

寻找相似样本可以认为距离近的数据相似。

算法流程：

1.确定k的大小和距离计算方法。

2.从训练样本中得到k个与测试样本最近的样本。

3.根据k个最相似的训练样本的类别，通过投票的方式来确定测试样本的类别。

距离度量：

定义：

1.非负性：$f(A,B)\geq 0$，当且仅当 $A=B$ 时等号成立。

2.反身性：$f(A,B)=f(B,A)$

3.三角不等式：$f(A,B)\leq f(A,C)+f(B,C)$

通用公式：

明可夫斯基距离：$d(x_1,x_2)=\sqrt[p]{\sum_{i=1}^{n}|x_{1i}-x_{2i}|^p}$

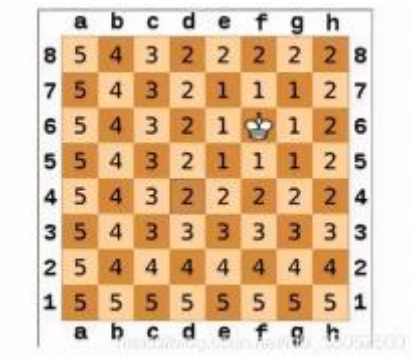
曼哈顿距离：$p=1$。欧氏距离：$p=2$。切比雪夫距离：$p=+\infty$

曼哈顿距离(L1 范数)：也称为城市街区距离，因为开车不能穿墙(只能沿“水平竖直”道路)

欧氏距离：n 维空间有：$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$

切比雪夫距离：

国王走一步能够移动到相邻的 8 个方格中的任意一个。那么国王从格子 (x_1,y_1) 走到格子 (x_2,y_2) 最少需要多少步？



最少步数总是 $\max\{|x_2-x_1|,|y_2-y_1|\}$ 步。有一种类似的一种距离度量方法叫切比雪夫距离(L∞范数)。

余弦距离 Cosine Distance：

也可以叫余弦相似度。相比距离度量，余弦相似度更加注重两个向量在**方向**上的差异，而非距离或长度上。

$$\cos\theta=\frac{\vec{x}\bullet\vec{y}}{\left|\vec{x}\right|\left|\vec{y}\right|}=\frac{\sum_{i=1}^{n}x_iy_i}{\sqrt{\sum_{i=1}^{n}x_i^2}\sqrt{\sum_{i=1}^{n}y_i^2}}$$

杰卡德距离：

$d_J(A,B)$ 用来衡量两个集合差异性，是杰卡德相似系数的补集，定义为 $1-J(A,B)$。而杰卡德相似系数(Jaccard similarity coefficient)，也称杰卡德指数(Jaccard Index)，是用来衡量两个集合相似度的一种指标。

$$J(A,B)=\left|\frac{\#(A\cap B)}{\#(A\cup B)}\right|,\quad d_J(A,B)=1-J(A,B)=\frac{\left|\#(A\cup B)\right|-\left|\#(A\cap B)\right|}{\left|\#(A\cup B)\right|}$$

例如：两个 n 维向量样本 A(0111)和 B(1011)。有 $J(A,B)=\frac{p}{p+q+r}=\frac{2}{2+1+1}=0.5$

p：样本 A 与 B 都是 1 的维度的个数　q：A1，B 0　r：A0，B1　s：A0，B0。

广义 Jaccard 相似系数：$J_g(a,b)=\frac{\sum_i\min(a_i,b_i)}{\sum_i\max(a_i,b_i)}$

k 邻近算法小结：

对异常数据不敏感，具有较好的抗噪性。K 近邻算法的计算效率不高。当训练集较小的时候，K 近邻算法易导致过度拟合。

3.分类-朴素贝叶斯

“朴素”： 假设变量间相互独立(没有这个前提，朴素贝叶斯模型的分类效果会较差) $\Rightarrow P(AB)=P(A)P(B)$，可得：$P(AB|X)=\frac{P(ABX)}{P(X)}=\frac{P(AX)P(BX)}{P(X)P(X)}=P(A|X)P(B|X)$

贝叶斯公式： $P(Y|X)=\frac{P(X|Y)P(Y)}{P(X)}$。如果用 x 代表属性集合(数据集)，y 表示类别标签(预测)，即 $P(y_i|x)=\frac{P(x|y_i)P(y_i)}{P(x)}$。当各个属性相互独立时，有 $P(x|y_i)P(y_i)=P(y_i)\prod_{j=1}^{m}P(a_i|y_i)$，其中 $x=\{a_1,a_2,\cdots,a_n\}$，每个 a_i 代表数据的一个属性，可以理解为 $x=a_i$。

以 $y=True/False$ 为**例子**，有：

$$P(T|X)=\frac{P(x_1|T)P(x_2|T)\cdots P(x_n|T)P(T)}{P(x_1)P(x_2)\cdots P(x_n)},P(F|X)=\frac{P(x_1|F)P(x_2|F)\cdots P(x_n|F)P(F)}{P(x_1)P(x_2)\cdots P(x_n)},$$

不难发现分母相同，于是比较分子大小即可完成分类(分类为 T 或 F)。推广到一般，就是当

分子 $P(y_i)\prod_{j=1}^{m}P(a_i|y_i)$ 最大时，预测 x_i 最可能标签为 y_i。

该思想的来源：

比如要求 P(违约|男,收入中,本科,未婚)的值大多数情况下很难，但 P(男|违约),P(收入中|违约),P(本科|违约), P(未婚|违约)一般是容易的。则有：

P(违约|男,收入中,本科,未婚) = [P(男|违约) • P(收入中|违约) • P(本科|违约) • P(未婚|违约) • P(违约)] / [P(男) • P(收入中) • P(本科) • P(未婚)]

0 概率处理：可能某个特征属性在训练集中没有出现，这样如果按照以上的计算方法，就会导致后验概率为 0 的情况，所以要对条件概率的求解公式修正：$P(B_i)=\frac{0+1}{N}$ (加 1 平滑)

拉普拉斯平滑：$P(X=x_i|Y=k)=\prod_{i=1}^{d}\frac{\sum_{j=1}^{n}I(X_i^{(j)}=x_{ij},y_i=k)+\alpha}{\sum_{j=1}^{n}I(y_i=k)+\lambda\alpha}$。$\lambda$ 为特征的维数，α 是平滑值。x_{ij} 代表第 i 个特征的第 j 个选择。I 指示函数(indicator function)用于描述某个事件是否发生的函数 $I(y_i=k)=\begin{cases}1,if & y_i=k\\0,if & y_i\neq k\end{cases}$

比如特征为有钱/没钱，维数 $\lambda=2$。$\alpha=1$时称为 Laplace 平滑，$0<\alpha<1$时称为 Lidstone 平滑，$\alpha=0$ 说明不做平滑。

当特征是连续变量时，

1.将连续的特征离散化。困难：如何离散化？(tip:等频分箱、等宽分箱、基于聚类的分箱、基于决策树的分箱)

2.采用高斯模型，假定连续型特征服从正态分布，使用正态分布对条件参数进行求解。

$$P(X=x_i|y=c)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

朴素贝叶斯算法**优缺点分析**：

分类效率良好，逻辑简单，易于实现。对小规模数据表现良好，能处理多分类任务，适合增量式训练。特征之间相互独立，只会涉及二维存储，分类算法过程开销小。对缺失数据不敏感，常用于文本分类。

在属性个数多且属性之间的相关性比较大时，分类效果不理想。分类过程中需要知道先验概率和条件参数，而先验概率和条件参数的求解过程取决于我们选择的模型，当选择不合适的模型时，求得的先验概率和条件参数的值也会影响分类的结果。

例子：将邮件划分为正常邮件和垃圾邮件：

不妨记正常邮件为 T，垃圾邮件为 F。训练集邮件总数为 N，训练集正常邮件总数为 N_T，训练集垃圾邮件总数为 N_F，某单词 C 出现在正常或垃圾邮件的数目分别为 C_T 与 C_F，有：

$$P(T)=\frac{N_T+1}{N+2},P(F)=\frac{N_F+1}{N+2},P(C|T)=\frac{C_T+1}{N_T+2},P(C|F)=\frac{C_F+1}{N_F+2}。$$

4.分类- SVM 支持向量机 Support Vector Machine

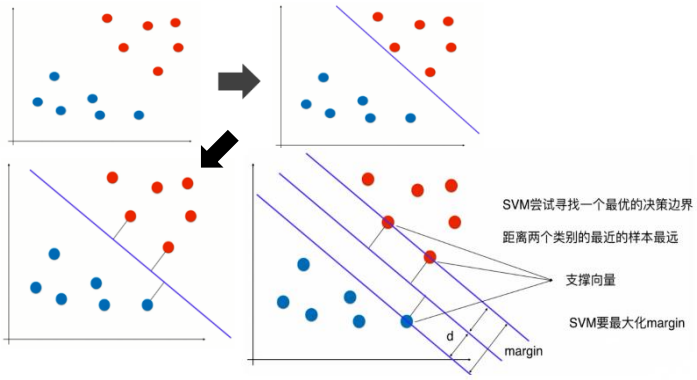
解决**小样本**、**非线性**及**高维**模式识别。根据有限的样本信息在**模型的复杂性**(即对特定训练样本的学习精度)和**学习能力**(即无错误地识别任意样本的能力)之间寻求**最佳折衷(全局最优解)**，以期获得最好的泛化能力(通过使用**最大分类间隙**来设计决策最优分类超平面)。

应用：在模式识别领域中的文本识别与分类、人脸识别等，工程技术、信息过滤。

寻找决策边界：

决策边界明显不唯一，需要找更“好”的。量化标准就是“**分类间隔**”(两虚线的间隔)。

分割线的特点：找到红蓝点中离这根直线最近的点，同时尽可能使得这些点离这根直线**尽可能的远**。总的来说就是这根直线离蓝色的样本尽可能远，离红色样本也尽可能远。即：使两虚线中的**最小距离取得最大值**(不过图四中的两虚线画成实线了，「支持向量所在的两根线」)。



那如何求这个**距离** d：高维(n 维欧式空间)中某个点 $M_0: x_0=(x_0^{(1)},x_0^{(2)},\cdots,x_0^{(n)})$ 到某一个超平面 $L: \omega\bullet x+b=0$ (法向量 $\omega=(\omega^{(1)},\omega^{(2)},\cdots,\omega^{(n)})$)的**距离** $d=\dfrac{|\omega\bullet x_0+b|}{\|\omega\|}$

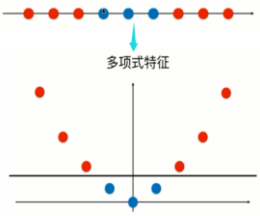
想要**距离最大**，可证就要法向量长度 $\|\omega\|$ 取最小值，也就是 $\min\dfrac{1}{2}\|\omega\|^2$

具体可参考 SMO 算法 https://zhuanlan.zhihu.com/p/29212107

为提高**容错能力**(泛化能力)，允许一些点出现在虚线和支撑向量所在直线(两条直线)之间。因此我们在不等式右边减去一个整数 ζ(zeta)(罚分值 Penalty)，于是要(yào)求的是 $\min\dfrac{1}{2}\|\omega\|^2+\sum_{i=1}^{m}\xi_i$，但是显然 ζ 不能很大，为此我们还要限定 ζ 的大小：$\min\dfrac{1}{2}\|\omega\|^2+C\sum_{i=1}^{m}\xi_i$ (C 越大容错空间越小，C 越小容错空间越大)。此时称为 Soft Margin SVM(之前的称为 Hard Margin SVM)。最终就要不断最小化罚分总和。

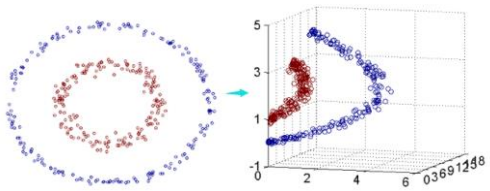
可以化为：约束条件：$\sum_{i=1}^{N}\alpha_i y_i=0, 0\le\alpha_i\le C(i=1,2,\ldots,N)$

目标函数：$\min\dfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(\vec{x_i}\times\vec{x_j})-\sum_{i=1}^{N}\alpha_i$



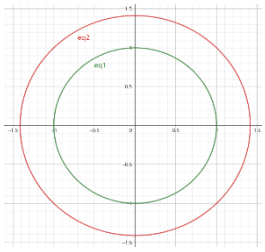
线性不可分：升维

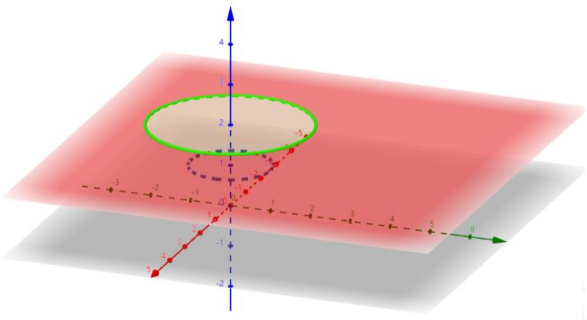
因为升高维度，数据会显得“稀疏”。如右图，没办法在一维空间下找到一个点分割红蓝样本。现在选择升高维度，使得样本能够坐落在 xy 平面，如第二个特征值设置为 x^2。这些点在 x 方向上的位置并没有改变，但是因为加入了 y 这个维度，此时就能找到一根直线分割。



核函数 Kernel 的作用：从低维空间到高维空间的映射，而这个映射可以把低维空间中线性不可分的两类点变成线性可分的(如右图)，其目的与多项式特征中升维是一样的道理。

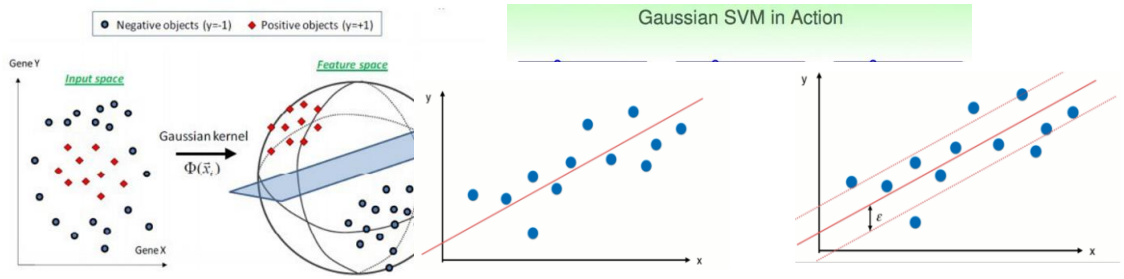
升维的一个例子：平面上无法用直线区分 $x^2+y^2=1$ 与 $x^2+y^2=2$，但 $(x,y)\to(x,y,\sqrt{x^2+y^2})$ 后在三维空间中可由 $z=\dfrac{3}{2}$ 区分这两个平面。





高斯核函数 RBF：$K(x,y)=e^{-\gamma\|x-y\|^2}$ 能将原始空间映射为无穷维空间。不过，如果选得

很大的话，高次特征上的权重实际上衰减得非常快，所以实际上（数值上近似一下）相当于一个低维的子空间;反过来,如果选得很小,则可以将任意的数据映射为线性可分——当然，这并不一定是好事，因为随之而来的可能是非常严重的过拟合问题。不过，总的来说，通过调控参数，高斯核实际上具有相当高的灵活性，也是使用最广泛的核函数之一。如下图：



SVM 只能分类吗？

SVM 解决分类的时候，同样指定一个 Margin 值，只不过现在这个 Margin 的定义和解决分类问题中不同。我们想要 Margin 的两根虚线范围里能够包含的样本数据越多越好。这个思路其实和 SVM 解决分类问题是相反的思路。

文献阅读：

《统计学习方法》李航著；

吴恩达视频：https://study.163.com/course/introduction/1210076550.htm

5.回归

线性回归：

直接求解

$f_\theta(X_i)=\theta^T X_i$，式中 $\theta=(\theta_0,\theta_1,\cdots,\theta_d)^T, X_i=(x_{i0},x_{i1},\cdots,x_{id})^T$ (转置的原因是在概率论、机器学习等里面，默认都是**列向量**)，$1\le i\le n$ 且 $i\in\mathbb{N}$。

其中 $x_0\equiv 1$，代表 θ_0 是**偏置项**。i 代表的是对于第 i 行(一个数据对象)，d 代表的是有 d 个属性。因此 f 的作用是先将数据集表示为一个 $n\times(d+1)$ 的矩阵 X (每一行代表一个示例，后 d 个元素对应示例的 d 个特征值)，然后对矩阵的各个属性值赋予参数 θ，求和得到一个目标预测值。

预测肯定存在误差，即：**真实值=预测值+误差**。对每个样本 $y^{(i)}=\theta^T x^{(i)}+\varepsilon^{(i)}$ (标签=预测标签+误差项)。$\varepsilon^{(i)}$ 独立同分布，且服从 $\mu=0,\sigma=\theta^2$ 的高斯分布(正态分布)。由于误差服从高斯分布，因此 $P(\varepsilon^{(i)})=\dfrac{\exp\left(-\dfrac{(\varepsilon^{(i)}-\mu)^2}{2\sigma^2}\right)}{\sqrt{2\pi}\sigma}=\dfrac{\exp\left(-\dfrac{(\varepsilon^{(i)})^2}{2\sigma^2}\right)}{\sqrt{2\pi}\sigma}$ (注意到均值 $\mu=0$)。因此 θ 与 x 组合，成为真实值 y 的概率 $P(y^{(i)}|x^{(i)};\theta)=\dfrac{\exp\left(-\dfrac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right)}{\sqrt{2\pi}\sigma}$。将关注点从对 x 的概率函数转变为对 θ 的似然函数(已知发生事件 x，希望知道参数 θ 应该是多少) $L(\theta)=\prod_{i=1}^{n}P(y^{(i)}|x^{(i)};\theta)$，

$$\ln L(\theta)=\ln\prod_{i=1}^{n}P(y^{(i)}|x^{(i)};\theta)=\ln\prod_{i=1}^{n}\frac{\exp\left(-\dfrac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right)}{\sqrt{2\pi}\sigma}=\sum_{i=1}^{n}\ln\frac{\exp\left(-\dfrac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right)}{\sqrt{2\pi}\sigma}=n\ln\frac{1}{\sqrt{2\pi}\sigma}-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y^{(i)}-\theta^T x^{(i)})^2$$

。似然函数解决的是：什么样的 θ 能使得这样的 θ 跟 x 组合之后，成为

y 的可能性越大。

不妨令目标函数 $\boldsymbol{J(\theta)=\frac{1}{2}\sum_{i=1}^{n}(y^{(i)}-\theta^T x^{(i)})^2}$，因为 $\ln L(\theta)$ 越大越好，因此 $J(\theta)$ 越小越好。因为矩阵的平方等于转置乘自身，因此 $J(\theta)=\frac{1}{2}\sum_{i=1}^{n}(\theta^T x^{(i)}-y^{(i)})^2=\frac{1}{2}(X\theta-y)^T(X\theta-y)$，求偏导 $\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T-y^T)(X\theta-y)\right)=\nabla_\theta\left(\frac{1}{2}(\theta^T X^T X\theta-\theta^T X^T y-y^T X\theta+y^T y)\right)=$ $\frac{1}{2}(2X^T X\theta-X^T y-(y^T X)^T)=X^T X\theta-X^T y$，令偏导等于 0，则 $\boldsymbol{\theta=(X^T X)^{-1}X^T y}$。

注 1： $P(x|\theta)$ 是一个有着两个变量的函数。如果将 θ 设为常量，得到的是一个概率函数(关于 x 的函数)；如果将 x 设为常量，得到的是一个似然函数(关于 θ 的函数)。已知一定量的样本数据 x，在 x 条件下，参数 θ 的最大似然估计就是使得该样本数据 x 出现的概率(可能性)最大，那么参数 θ 的似然函数就是该样本数据出现的概率。似然和概率只是数值相等 $L(\theta|x)=P(x|\theta)=\sum_{i=1}^{n}P(x_i|\theta)$，但定义域不同，似然函数的定义域是 θ，而概率函数的定义域是 x，即我们只能说“参数等于某个值时的似然是多少，事件(发生)的概率是多少”。最大似然估计的含义：**事件已经发生，里面未知参数取何值时，该事件发生的概率最大。**

注 2： $X^T X$ 是对称阵。对于对称阵 A，有 $\nabla_\theta(\theta^T A\theta)=2A\theta$。$\nabla_\theta f(\theta)$ 即 $\frac{\partial f(\theta)}{\partial\theta}$。

注 3： $\mathrm{MSE}=E(\theta^T)=(y-\theta^T X)^T(y-\theta^T X),\frac{\partial \mathrm{MSE}(\theta)}{\partial(\theta^T)}=2X^T(X\theta^T-y)$ 亦得 $\theta^T=(X^T X)^{-1}X^T y$。

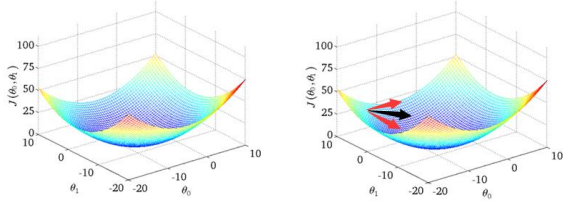
模型诊断 $R^2(y,\hat{y})=1-\frac{SS_{res}}{SS_{tot}}$。残差平方和 $SS_{res}=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$，总平方和 $SS_{tot}=\sum_{i=1}^{n}(y_i-\bar{y})^2$。其中 y_i 为真实值，$\bar{y}$ 为真实值的平均值，$\hat{y}_i$ 为模型估计值。

梯度下降：

上面已经得到 θ 表达式，那机器“学习”在哪里呢？首先，不是所有的算法都能直接计算出参数值。其次，矩阵的逆不一定存在，因此这个公式不能保证求出 θ。

因为梯度的方向始终是向上，而一般要求解的是损失函数的最小值，所以叫梯度下降。

以只有 $\theta_{0,1}$ 参数为例，因为 $x_{0,1}$ 没有关系，那他们对应的 $\theta_{0,1}$ 更不可能有关系。因此分别求偏导 $\frac{\partial J}{\partial\theta_0},\frac{\partial J}{\partial\theta_1}$ (分别对应右图中的上/下面的红色箭头)。向量相加即为综合结果。

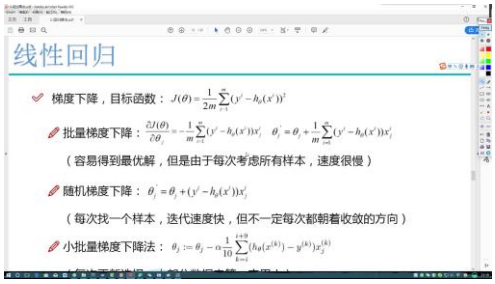


目标函数： $J(\theta)=\frac{1}{2n}\sum_{i=1}^{n}(y^{(i)}-\theta^T x^{(i)})^2=\frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)}-h_\theta(x^{(i)})\right)^2$。下面式中 i 代表第 i 个样本，j 代表第 j 列(属性)。因为是对 θ_j 求导，因此只需要考虑 θ_j 的系数 x_j。

批量梯度下降： $\frac{\partial J(\theta)}{\partial\theta_j}=-\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}-h_\theta(x^{(i)})\right)x_j^{(i)},\theta_j^{'}=\theta_j+\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}-h_\theta(x^{(i)})\right)x_j^{(i)}$。

随机梯度下降： $\theta_j^{'}=\theta_j+\left(y^{(i)}-h_\theta(x^{(i)})\right)x_j^{(i)}$，也就是小批梯度下降中 batch=1。

小批量梯度下降： $\theta_j^{'}=\theta_j-\alpha\frac{1}{n_c}\sum_{k=i}^{i+n_c}\left(h_\theta(x^{(k)})-y^{(k)}\right)x_j^{(k)}$，$n_c$ 为选择的数据个数，常取 256,512。α 为学习率，越小越好。



非线性回归(多项式回归)：

一元 m 次多项式回归方程：$y=\omega_0+\omega_1 x+\omega_2 x^2+\cdots+\omega_m x^m$

二元二次多项式回归方程：$y=\omega_0+\omega_1 x_1+\omega_2 x_2+\omega_3 x_1^2+\omega_4 x_2^2+\omega_5 x_1 x_2$

令 $x_1=x, x_2=x^2, \cdots, x_m=x^m$ 可使一元 m 次方程化为 m 元线性回归方程 $y=\omega_0+\omega_1 x_1+\cdots+\omega_m x_m$。因此用多元线性函数的回归方法就可解决多项式回归问题。

显然由于任意一个函数在一个较小的范围内都可以用多项式任意逼近，很容易出现过拟合问题。处理过拟合问题常用的方法：1.减少特征数量：人工的挑选重要的特征，去除不重要的特征。2.正则化 Regularization：保留所有特征，但是减少参数 ω 的值。

正则化：避免过拟合，能处理特征点比样本点多的问题

L1 Lasso 回归　　降维　　有 $J=J_0+\alpha\|\omega\|_1=J_0+\alpha\sum_{\omega}|\omega|$

L2 Ridge 岭回归　采用惩罚算子 w，计算时用 $|y-X\bullet w|$，　$w_{ridge}=(X^T x+aI)^{-1}X^T y$

有 $J=J_0+\alpha\|\omega\|_2=J_0+\alpha\sum_{\omega}\omega^2$

注：范数是一个用于衡量向量大小的函数。通常用符号 $\|x\|$ 表示，表示向量 x 的范数。常见的范数有 L1 范数，L2 范数等。

Logistic 回归：

以购房为例，假如只考虑能不能买得起房，就是一个突变问题，需要 S 型曲线来拟合。

即 $f(y)=\dfrac{e^y}{1+e^y}$，值域为 (0,1)，可用来代表概率。虽然名为回归，但实际常解决二分类问题。

6.关联规则挖掘

客户 1：尿布，啤酒，薯片…
客户 2：尿布，可乐，薯片…
客户 3：尿布，啤酒… 。
客户 4：尿布，啤酒，薯片…
客户 5：啤酒，可乐，薯片…
……
客户 100：……

例子如右：这里研究关联规则："啤酒→尿布"

置信度： P(啤酒→尿布)=P(尿布|啤酒)=P(啤酒∪尿布)/P(啤酒)$=\dfrac{3}{4}=75\%$

支持度： P(啤酒∪尿布)=N(尿布与啤酒同时出现)/N(全部购物数据)$=\dfrac{3}{100}$

设置**最小支持度阈值**和**最小置信度阈值**，当某个关联规则同时满足这两个预选设好的阈值，便把它称为**强关联规则**。

事务：每一条购物信息　**项：**购物信息中的一项物品　**项集：**包含 0 或多个项的集合

k 项集：含有 k 个项的项集　**关联规则：**蕴含式 $A\rightarrow B$，其中 $A\neq\varnothing, B\neq\varnothing, A\cap B=\varnothing$

频繁项集：支持度大于预定义的最小支持度阈值的项集

于是，从数据集中挖掘强关联规则的过程分为两步：

1.需要找出满足最小支持度阈值的项集，即频繁项集；

2.根据最小置信度阈值，从频繁项集中生成强关联规则。

也就是找出 1 项集，2 项集，…，n 项集。然后从中找出大于支持度阈值(如 50%)的项集，再判断置信度。

当数据集的项过多时，该算法产生的候选项集过多，计算量太大，假设有100个1项集，则候选频繁项集个数：$C_{100}^{1}+C_{100}^{2}+\cdots+C_{100}^{100}=2^{100}-1=1267650600228229401496703205375$

此时要用到 **Apriori 性质**：

如果一个项集 A 是频繁项集，那么它的非空子集 B 也是频繁项集。

或者：如果一个项集不是频繁项集，那它的超集也不是频繁项集。

因此只需检索由频繁 k 项集生成的 $k+1$ 项集，而若某 k 项集已不满足，$k+1$ 项集就更不可能。

缺点：

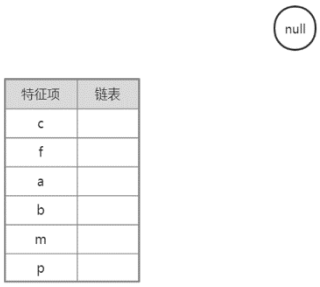
Apriori 算法每一次迭代都需要遍历数据集，当数据集过大无法一次性载入内存时，遍历的效率会变得非常低。Apriori 算法仍然会产生大量无用的候选多项集。

FP-growth 算法：

高效地发现频繁项集，但是不能用于发现关联规则。FP-growth 算法的执行速度快于 Apriori 算法，通常性能要好两个数量级以上。FP-growth 算法只需要对数据集扫描两次，它发现频繁项集的过程如下：构建 FP 树，从 FP 树中挖掘频繁项集。

如下例，原数据为前两列：

编号	含有数据项	排序的频繁项集
1	f,a,c,d,g,i,m,p	c,f,a,m,p
2	a,b,c,f,l,m,o	c,f,a,b,m
3	b,f,h,j,o	f,b
4	b,c,k,s,p	c,b,p
5	a,f,c,e,l,p,m,n	c,f,a,m,p



①设置最小支持度为 0.6，频繁 1 项集应出现至少 0.6 * 5 = 3 次。

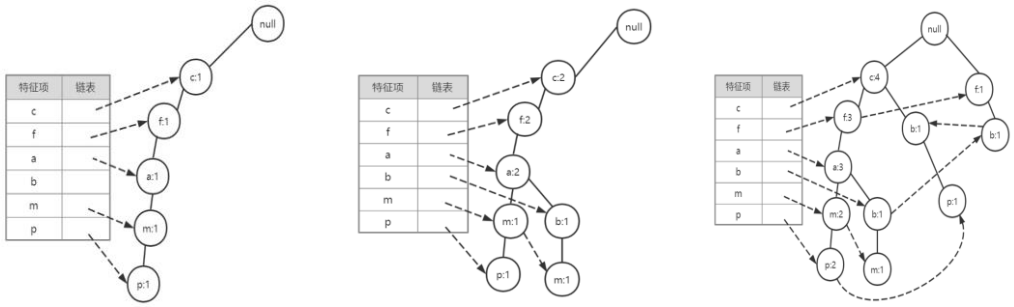
②扫描数据库(上表的前两列)，识别所有的单个项并计算它们的支持度计数，提取出支持度计数≥3 的项，并按照支持度从高到低排序：L={{c:4},{f:4},{a:3},{b:3},{m:3},{p:3}}，得到上表的第三列。

③构建根节点，标记为 null，创建频繁项集，将链接列设为空，如右上图。

④将第 1 条事务插入频繁模式树中，只插入频繁项集部分{c,f,a,m,p}，将每个节点的支持度计数设为 1，同时设置好对应项的链接。如左下图。

⑤将第 2 条事务插入频繁模式树中{c,f,a,b,m}，当节点相同，**共享节点并将节点的支持度加 1**，同时更新对应项的链接。如下中图。

⑥最终频繁模式树如右下。



⑦条件模式基(conditional pattern base)是指从原始数据集的频繁树中投影出的子数据集，它与特定后缀相关联。以所查询元素项为结尾的路径集合。每条路径其实都是一条前缀路径(prefix path)。总之，一条前缀路径是介于所查询元素项与树根节点之间的所有内容。

通过创建的头指针表来得到前缀路径。头指针表包含相同类型元素链表的起始指针。一旦到达了每一个元素项，就可以上溯这棵树直到根节点为止。

条件模式基有什么用？以 p 的条件模式基为例，当 p 元素出现的前提下，c,f,a,b,m 各元素出现的概率。

例如：构造以 p 为后缀的条件模式库的方法如下：

1.找到频繁模式树中所有的 p 节点，如图蓝框。

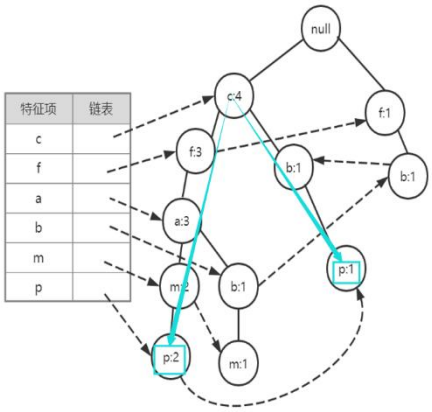
2.回溯所有的路径(如图)得到(cbp, cfamp)，除去 p 节点后，将路径中的剩余节点当作新的项集。

3.项集的支持度设置为对应路径中 p 节点的支持度。

得到以 p 为后缀的条件模式库为：{cfam：2，cb：1}。寻找方法如下图：

并以此可得到其他特征项的条件模式库：

特征项	条件模式库
c	{}
f	{c:3}
a	{cf:3}
b	{cfa:1；c:1；f:1}
m	{cfa:2；cfab:1}
p	{cfam:2；cb:1}



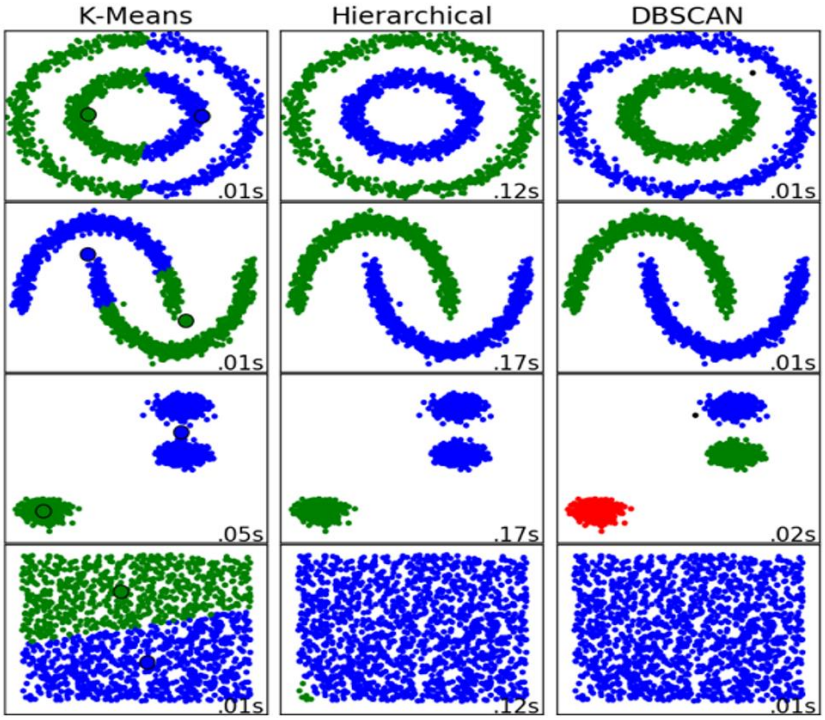
聚类分析

说在前头

聚类模型的**本质**：将数据集中相似的样本(**无标签**)进行分组(**加标签**)的过程。

每个组称为一个簇(cluster)。每个簇的样本对应一个潜在的类别。样本是没有类别标签的，因此聚类是一种典型的无监督学习任务，这些簇满足以下两个条件：相同簇的样本之间距离较近，不同簇的样本之间距离较远。

常见聚类：划分聚类、层次聚类、基于密度。



1.划分聚类(基于原型的聚类 Prototype-Based Clustering)

假设聚类结构能通过一组原型刻画。

K 均值聚类：

①取初始质心：从数据集中随机选取 k 个初始质心。(k 是事先给定的)

②划分数据点：计算每个点与各个质心的距离(如欧氏距离)，将数据点划分到距离最近的质心所在的簇内。

③寻找新质心：原质心不在新簇的质心位置，于是应该计算每一个簇的新质心，也就是求均方误差和 SSE(Sum of Squared Error)最小化问题。$SSE=\sum_{i=1}^{k}\sum_{x_j\in C_i} r_{ik}\left\|x_j-u_i\right\|^2$，使 SSE 最小的中心点是均值 $u_i=\frac{\sum_{x_j\in C_i} X}{j_i}$。式中 C_i 是第 i 个簇，含有 j 个数据点，x_j 为该簇内第 j 个数据点，u_i 是质心，$r_{ik}\in\{0,1\}$(若 x_i 被划分到簇 k 中则 $r_{ik}=1$，且对于 $j\neq k$ 有 $r_{ij}=0,\sum_{j=1}^{k} r_{ij}=1$)。$\sum_{i=1}^{k}$ 表示各个簇，$\sum_{x_j\in C_i}$ 表示簇内。

④重复②~③直至各个数据点不再变更所属的簇。

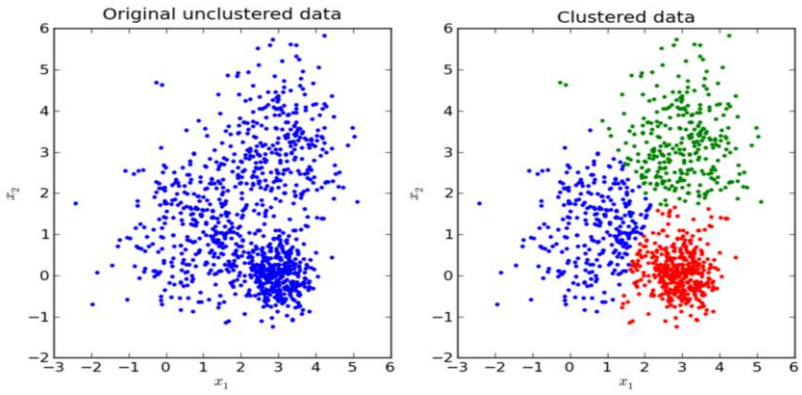
多次运行，每次选取一组不同的随机初始质心可以避免陷入局部最优。

K-means 算法的优点：算法实现简单、直观。

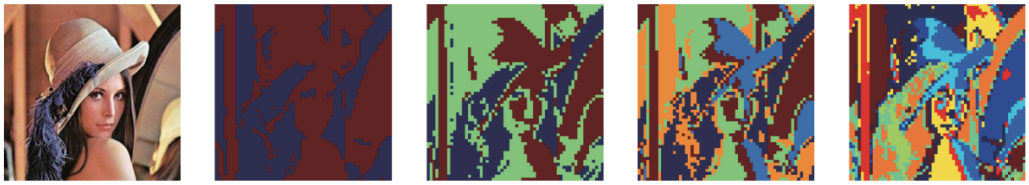
K-means 算法的缺点：事先指定 k 值，聚类结果依赖于 k 个初始质心的选择。易陷入局部最优，不易处理非簇状数据。聚类结果容易受离群值影响。

时间复杂度：$O(mnkt)$：m为特征维度、n为样本数量、k为簇数量、t为迭代次数。每一步迭代中，每个样本需要与k个簇中心进行距离计算。

K 均值聚类分类效果如下：



图像压缩：给出了原始图像以及使用不同 k 值得到的 K-means 分割结果：



原始图片　$K=2$　$K=3$　$K=5$　$K=10$

其他原型聚类

K-means 聚类变种——**K-medoids** 聚类（K 中心点聚类）

K-medoids 聚类使用数据点作为聚类中心，而不是像 K-means 聚类那样使用平均值。这意味着 K-medoids 对于噪声和异常值更具鲁棒性，并且能够产生更好的聚类结果。

鲁棒性是指一个算法或系统对于异常情况、噪声或错误的输入数据的处理能力和稳定性。具有良好鲁棒性的算法或系统能够在面对这些问题时仍然保持稳定可靠的输出结果。

二分 K-means：一种基于 K-means 算法的聚类方法，与传统的 K-means 不同之处在于，它利用了二分思想，将原始数据集不断划分为两个子集，直至达到指定的聚类数目，得到最终的聚类结果。(每次从所有簇中选择具有最大 J 值的簇，然后划分为两个簇，重复该过程，直至簇集合中含有 K 个簇。其中 J 指的是簇内平方误差和 SSE，K 指的是最终聚类的簇数)。

K-medians 聚类：使用样本每个维度的中位数作为簇的质心。

K-means++聚类：在初始化质心的过程中，选择相互距离尽可能远的样本作为初始质心。

基于粗糙集的 K-means 聚类：在基于粗糙集的 K-means 算法中，一个样本可以被划分给多个簇。

学习向量量化 LVQ(Learing Vector Quantization)：是一种有监督的聚类方法，因为它需要训练数据的标签信息(或假设存在数据带有类别标签)来指导向量更新过程。它通常用于具有清晰类别的数据集。可看作通过聚类来形成类别“子类”结构，每个子类对应一个聚类簇。

高斯混合聚类 GMM(Gaussian Mixture Model)：是一种无监督学习方法，使用高斯分布、贝叶斯公式、极大似然法、聚类等原理，用于探索数据中的潜在聚类结构。

2.层次聚类

层次聚类(hierarchical clustering)在不同层级上对样本进行聚类，逐步形成树状的结构.根据层次分解是以自底向上(合并)是自顶向下(分裂)方式，层次聚类方法可以分为：**聚合式(凝聚式)聚类**(agglomerative clustering)、**分拆式(分裂式)聚类**(divisive clustering)，两种方法均是启发式的策略，并没有去优化一个明确的目标函数来实现聚类，很难严格评价聚类的效果。

层次聚类方法的**局限性**：1.计算量大。2.一旦合并或分裂执行，就不能修正。如果某个合并或分裂决策在后来证明是不好的选择，该方法也无法退回并更正。

原型聚类样本所在簇可能变化，但层次聚类样本所在簇迭代后不变。

凝聚层次聚类

采用自底向上策略，首先将每个样本作为单独的一个原子簇，然后合并这些原子簇形成越来越大的原子簇，直到所有的样本都在一个簇中（层次的最上层），或者达到一个终止条件。绝大多数层次聚类方法属于这一类。

在开始时把每个样本都当成一簇，然后在每一次迭代中将最相似的(距离最近)两个簇进行合并，直到把所有簇合并为包含所有样本的一簇。

单连接(最近邻距离)：簇 G 和簇 H 之间的距离定义为两簇之间最近的成员之间的距离。

完整连接(最远邻距离)：簇 G 和簇 H 之间的距离定义为两簇之间最远的成员之间的距离。

平均连接：表示两簇间所有成员对的平均距离 $d_{avg}(G,H)=\dfrac{\sum_{i\in G}\sum_{i'\in H}d_{i,i'}}{n_G n_H}$

单连接法只需要两簇内有成员对距离足够近就将两簇合并，而并没有考虑其他簇内其他成员的距离，因此单连接方法形成的簇很有可能违背**紧致性特征**(簇内成员应该尽可能相似)。(也就是因为存在某些点相似就把两个簇当做相似了)

完整连接法是另外一个极端，只有当两簇的联合的成员间的距离相对较小时才将两簇进行合并，因此完整连接法倾向于生成紧致簇。(与单连接恰好相反)

平均连接法介于单连接和完整连接之间，易于生成相对紧致的簇同时簇间距离较远。

AGNES 层次聚类算法(单连接自底而上凝聚层次聚类方法)：

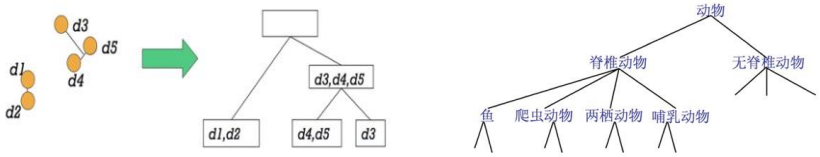
先将每个样本看成一个簇，然后每次对距离最短的两个簇进行合并，不断重复，直到达到预设的聚类簇个数。

分裂层次聚类

采用自顶向下策略，首先将所有样本置于一个簇中，然后逐渐细分为越来越小的簇，直到每个样本自成一个簇，或者达到某个终止条件，例如达到了某个希望的簇的数目，或者两个最近的簇之间的距离超过了某个阈值。

分拆式聚类将所有样本集合看作一簇，以自上而下的方式，递归地将现有的簇分拆为两个子簇。如：**二分 K-means 聚类**(选择半径最大的簇，对该簇进行 K-means 聚类分为两个子簇，重复此过程直到达到想要的簇个数)。**最小生成树法**(将每个样本看作一个图节点，将样本间距离看作节点间边的权重，根据此图建立最小生成树。从权重最大处将该簇分拆为两簇，然后重复此过程直到达到想要的簇个数。实际上，该方法得到的聚类结果和单连接的聚合聚类得到的结果一致。)

聚合式聚类，分拆式聚类如下图(左聚合 右分拆)：



3.密度聚类

密度聚类又称为基于密度聚类(density-based clustering)。此类算法假设聚类结构由样本分布的紧密程度确定，以数据集在空间分布上的稠密程度为依据进行聚类，即只要一个区域中的**样本密度大于某个阈值**，就把它划入与之相近的簇中。

不同于基于质心的原型聚类算法，基于密度的算法不需要事先设置 k 值。常用的密度聚类算法：DBSCAN、MDCA、OPTICS、DENCLUE 等，下面主要讲解 DBSCAN 算法。

①用户设定邻域参数$(\varepsilon, \text{Minpts})$。

ε 为半径，Minpts 代表一个簇中最少的数据点个数，高于该值为高密度区域，否则为低密度区域。

②数据点分类

基于邻域参数，在 DNSCAN 算法中，所有的数据点可以分为以下三个类别：

核心点：如果某个点的邻域内的数据**点数目高于阈值** Minpts，则将这个点视为核心点。

边界点：边界点是**位于核心点邻域之内**的，但是其自身的邻域内数据点数小于 Minpts 的点，起到将高密度区域与低密度区域分割开的作用。

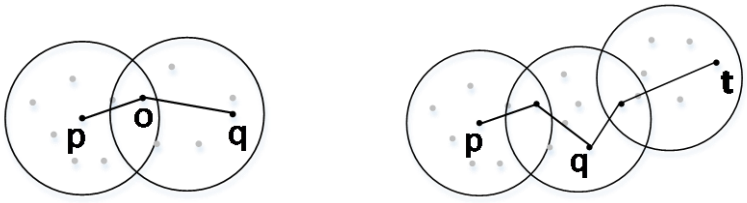
噪声点：既不是核心点也不是边界点的其他数据点，他们组成低密度区域。

密度直达： o 在 p 的邻域内，且 p 为核心点，则称从 p 到 o 是密度直达。

密度可达： o 在 p 的邻域内，从 p 到 o 是密度直达，而 q 对象的邻域内不包括 p，但是包括 o，这样 $p \rightarrow o \rightarrow q$，称 p 到 q 是密度可达的。

密度相连： q 和 p 是密度可达的，q 和 t 也是密度可达的，则 p 和 t 是密度相连的。

下图分别表示 p 到 q 是密度可达， p 和 t 是密度相连。



DBSCAN 的目标为找到密度相连数据点的最大集合，此集合作为最终的一簇。其实就是从某个随机点开始向外扩张直至 r 邻域内的数据个数小于 Minpts。

首先核心点各自成簇，采用密度相连的概念逐步对簇进行合并，打上标记。最终核心点密集的区域会被低密度噪声点包围，噪声点不单独成簇。所以采用 DBSCAN 进行聚类，最终的结果有一些数据点是没有标记的(作为簇-1)，这些就是噪声点。

优点：

1.无需事先设定簇的个数，算法根据数据自身找出各簇。
2.适于稠密的非凸数据集，可以发现任意形状的簇。
3.可以在聚类时发现噪音点、对数据集中的异常点不敏感。
4.对样本输入顺序不敏感

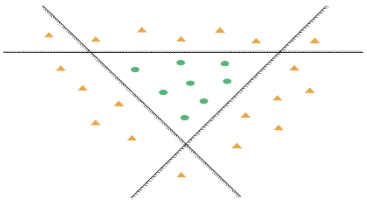
缺点：

1.因为是基于密度，若两个簇没有明显的可分间隔，很有可能被合并为同一个簇。
2.参数调节较为复杂，参数设置对结果影响较大。
3.对高维的数据处理效果不好。

集成模型

1.定义与作用与一些术语

集成学习 Ensemble Learning 通过构建并结合多个学习器来完成学习任务。



作用：①增强模型的表达能力：单个感知机无法正确分类数据，集成三个感知机能正确分类数据，如右图。②降低误差：分类器之间独立。

如果所有的个体学习器都是同类型的，则称这样的集成是同质的，同质集成的个体学习器称为基学习器。如果个体学习器不是同类型的，那么这样的集成是异质的，异质集成的学习器称为组件学习器。

弱学习器：性能略优于随机猜测的学习器。如在二分类问题上预测精度略高于 50%的分类器。

2.如何得到结果

①多数投票方法(majority vote)：选取预测次数最多的类别 C_1(有时拒绝预测没有过半的类别)。也能使用加权投票。

②平均(averaging)：$H(x)=\frac{1}{n}\sum_{i=1}^{n}h_i(x)$

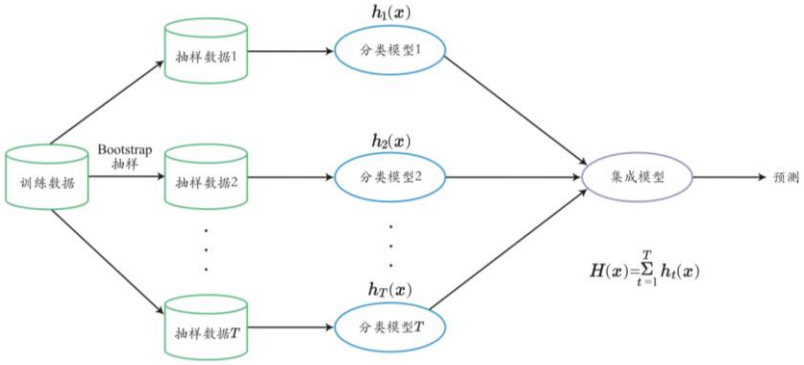
加权平均(weighted averaging)：$H(x)=\sum_{i=1}^{n}w_ih_i(x)$，其中 $\sum_{i=1}^{n}w_i=1$ 且 $w_i\geq 0$。如 AdaBoost。

个体学习器性能差异大的时候选用加权平均，个体学习器差异小的时候选用算术平均。因为权重是训练得出的，不完全可靠。

③学习法：通过另一个学习器来结合预测结果。比如 stacking 将弱学习器的学习结果作为输入，重新训练一个学习器来得到最终结果，即：多个初级学习器(基本模型)被训练，它们的输出作为次级学习器(元模型)的输入，然后次级学习器被训练。

3.典型的集成方法

Bagging： 对样本或特征随机取样(自助采样法 Bootstrap Sampling)，学习产生多个独立的模型，然后平均所有模型的预测值。如下图，基于每个抽样数据集 D_t 训练得到基础模型 $h_t(x)$，最终的分类器 $H(x)=\text{sign}\left(\sum_{t=1}^{T}h_t(x)\right)$。可以减小模型的方差(降低模型对训练数据的具体实例的敏感性，从而提高稳定性和泛化性)。典型代表随机森林。



通过有放回抽样得到与原数据集 D 大小一样的样本集 D_t。给定 n 个样本的数据集 D 用来创建分类器 M_t，基于分类器的投票返回类预测。一个样本有 $1-\frac{1}{n}$ 的概率不会被选到，则一个样本不被抽到的概率为 $\lim_{n\to\infty}(1-\frac{1}{n})^n=\frac{1}{e}=0.368$，说明训练集包含 63.2% 的不同样本。

每个 D_t 学习得到一个分类器 M_t。对于分类问题采用多数投票，对于回归问题采用均值投票。

有效性原因： 通过降低由于不稳定学习器 (unstable learners) 所造成的方差来降低错误率。不稳定学习器指那些在训练数据发生轻微变化的情况下,学习结果会发生很大变化的学习器，例如决策树。

Bagging 的优势：

用来提高那些方差大但偏差小的基模型(决策树，神经网络等)的预测性能。

单个模型不稳定，对训练数据轻微的改变就能够造成分类器性能很明显的变化 。

便于**并行**化。多个抽样数据的获取及基模型的训练互相没有关联，方便进行并行计算。

随机森林 Random Forest：“随机”是其核心，“森林”意在说明它是通过组合多棵决策树来构建模型。

假设使用三棵决策树组合成随机森林，每棵树各不相同且预测结果相互独立，每棵树的预测错误率为 40%。那么两棵树及以上预测错误的概率为：$0.4^3+C_3^2\times0.4^2\times(1-0.4)=0.352$。

随机森林的算法原理：

随机森林在构建每棵树的时候，为了**保证各棵树之间的独立性**，通常会采用两到三层的随机性：随机有放回的抽取样本，随机选取 m 个特征，随机选择特征取值进行分割(不遍历特征所有取值)。即：

使用 Bootstrap 抽样，从训练集 D 中获得大小为 n 的抽样训练集 D_t。从 d 个特征中随机选择 m 个特征，基于 D_t 中随机选取的 m 个特征学习得到一个决策树 $h_t(x)$。输出集成模型

$$H(x)=\begin{cases}\frac{1}{T}\left(\sum_{t=1}^{T}h_t(x)\right) & if\ Regression\\ \text{majority\_vote}\left(\{h_t(x)\}_{t=1}^{T}\right) & if\ Classification\end{cases}$$

其中 m 的选取可以交叉验证，也可以使用经验公式 $m=\log_2 d+1$，d 为特征维度。

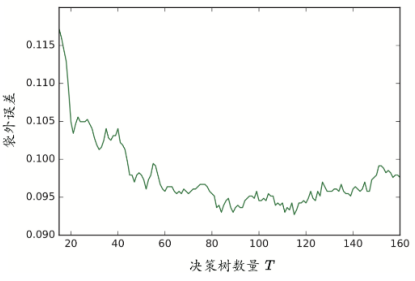
主要特点： 除了对样本进行有放回抽样，还对特征进行随机抽样，基本分类器最常见为决策树。

(**决策树局限性：** 局部最优，分类边界「决策特征只涉及单个特征的逻辑判断，导致决策边界是平行于坐标轴的直线，这就限制了决策树对分类边界的表达能力」)

随机森林的性能评估：

分类间隔 margin：正确分类某样本的决策树的比例减去错误分类样本决策树的比例，此值越大越好。假设对样本 A 有 75%的树分类正确，那么分类间隔就是 75% - 25% = 50%。

袋外误差 OOB (Out-Of-Bag Error)：随机森林对袋外样本(对某一棵树，没有被抽样到训练集中的样本)的预测错误率。计算方式：对每个样本，计算把该样本作为袋外样本的树对该样本的分类情况，以简单多数投票作为该样本的分类结果，以误分样本个数占样本总数的比率作为随机森林的袋外误差。



随机森林与特征选择：

随机森林能够给出特征的重要性度量，帮助进行特征选择。

计算特征的平均信息增益大小：训练决策树时，可以算出每个特征在每棵树中有多大的信息增益，算出每个特征在随机森林中的平均信息增益，作为该特征的重要性

计算每个特征对模型准确率的影响：通过打乱样本中某一特征的特征值顺序，产生新样本。新样本放入建立好的随机森林模型计算准确率，对于重要的特征来说，打乱顺序会极大的降低模型的准确率。

算法分析：

设单棵决策树的方差为 $\sigma^2(x)$，任两课决策树之间的相关性为 $\rho(x)$，随机森林一共由 T 棵决策树构成，则模型方差 $\mathrm{var}(x)=\rho(x)\sigma^2(x)+\frac{1-\rho(x)}{T}\sigma^2(x)$。

随机森林优缺点：

优点：

能够处理很高维度的数据，并且不用做特征选择。

对特征之间存在的多重共线性不敏感，并且能够在一定程度上处理缺失数据和不均衡数据。

在训练完后能够给出哪些特征比较重要。

容易做成并行化方法。

缺点：

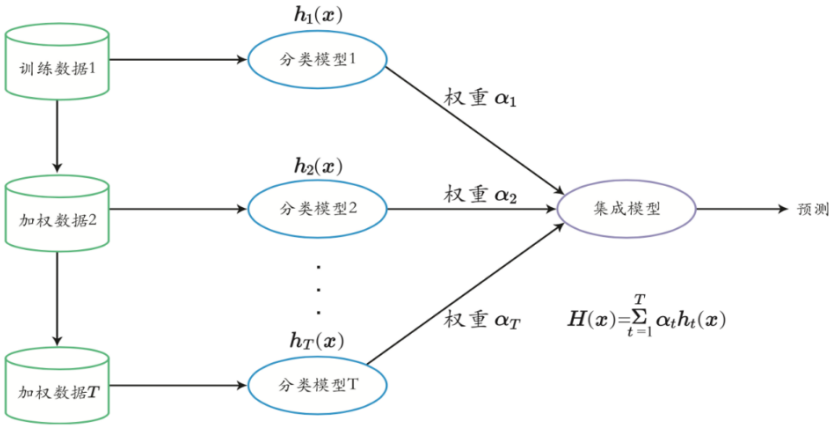
处理噪音较大的小样本和低维数据集的问题上会过度拟合。

相对于决策树，预测速度较慢，模型可解释性较差。

随机森林其他功能——计算样本的相似度

在建立随机森林的时候，记录样本两两之间出现在同一叶子节点的次数，生成相似性矩阵(proximity matrix)。如果越多的出现在同一叶节点，说明这两个样本越相似；如果越少的出现在同一叶节点，说明两个样本差异越大。

Boosting： 起源于 PAC(Probably Approximately Correct)。**串行**训练多个模型，后面的模型是基于前面模型的训练结果(误差)(重点关注前一个分错了的样本)。典型代表 AdaBoost。



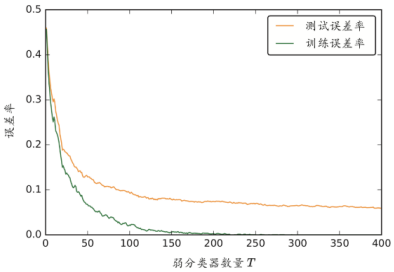
在 AdaBoost 方法中，将权重赋给每一个训练样本，迭代地学习 T 个分类器。初始权重均为 $\frac{1}{n}$，每一步迭代，被分错的样本的权重会降低。第 t 个基础模型的**权重更迭方法**如下：

计算当前弱分类器误差(加权错误率，弱分类器错误分类的样本的权重之和与训练集所有样本总权重之比)：$\varepsilon_t=\frac{\sum_{i=1}^{n}\omega_i^t I(y_i\neq h_t(x_i))}{\sum_{i=1}^{n}\omega_i^t}$，根据该误差计算权重：$\alpha_t=\frac{1}{2}\log(\frac{1-\varepsilon_t}{\varepsilon_t})$，更新样

本权重：$\omega_i^{t+1} \leftarrow \frac{\omega_i^t e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$（$Z_t$ 为归一化因子），最终的分类器为 $H(x)=\text{sign}(\sum_{t=1}^{T}\alpha_t h_t(x))$。

式中，I 为指示函数，当 True 时值为 1，False 时值为 0。sign 函数在 $x \geq 0$ 时输出+1，否则输出-1。当弱分类器对样本正确分类时 $y_i h_t(x_i)=1$，样本权重乘以小于1的因子 $e^{-\alpha_t}$，相当于降低了正确分类的样本的权重。当弱分类器对样本错误分类时 $y_i h_t(x_i)=-1$，样本权重乘以大于1的因子 $e^{-\alpha_t}$，相当于提高了错误样本的权重。

泛化误差对模型的测试集进行误差评估，计算公式为 $\text{Error}_{\text{test}}=E_{x,y}(L(y_i,H(x_i)))$。理论上随着弱分类器数目 T 的增大，泛化误差上界会增大。经验性结果表明，即使 T 很大也不容易过度拟合。



AdaBoost 的优缺点：

优点：防止过拟合，减少方差，减少偏差。

缺点：由于集成了多个弱分类器，模型的可解释性降低。当弱分类器太复杂或者效果太差时，容易导致过拟合。对于异常值比较敏感。

神经网络

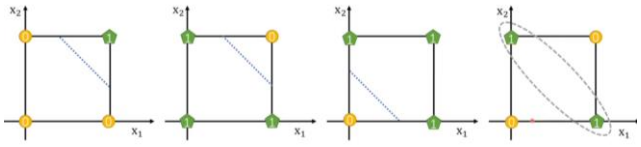
1.神经元模型

McCulloch-Pitts 模型(**M-P 模型**)：对于输入信号 $x_1,\cdots,x_n$ 赋以权重 $w_1,\cdots,w_n$，计算其 $\text{SUM}=w_1x_1+\cdots+w_nx_n=W^TX$，输出值 $Y=f(\text{SUM-T})$，式中 f 为激活函数(如 Sigmoid 函数 $f=\frac{1}{1+e^{-x}}$，Tanh 函数 $f=\tanh(x)$，ReLU 函数 $f=\max(0,x)$)，T 为阈值。

2.感知机

感知机 Perceptron：由两层神经元(输入层、输出层)组成。

能解决与、非问题，不能解决异或问题。why？参考下图，依次为与、与非、或、异或(定义可参考 https://blog.csdn.net/londa/article/details/119855241)：



$x_1 \wedge x_2$　　$\neg(x_1 \wedge x_2)$　　$x_1 \vee x_2$　　$x_1 \oplus x_2$

如何通过感知机优化 M-P 模型中的 w,T？

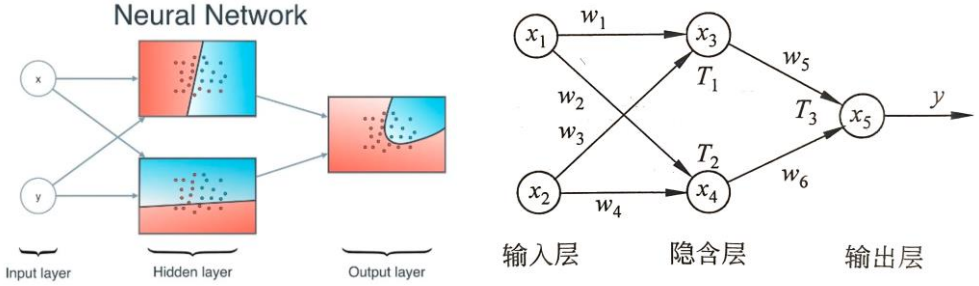
首先，将 T 视为 w_{n+1} 且 $w_{n+1}\equiv -1$，也就是将对阈值的学习转换为对权重的学习。接下来，对于训练样本 (x,y)，设目前感知机输出的是 y，权重调整为：$w_i \leftarrow w_i+\Delta w_i$，其中 $\Delta w_i=\eta(y-y)x_i$，式中学习率(Learing Rate) $\eta\in(0,1)$。

3.(多层)神经网络 Neural Network

神经元模型只能处理线性问题，对于非线性问题的模拟，Neural Network 把多个神经元组合到一起，形成一个网络来运算，这便可以找出更复杂的数据规律。

如左图，input layer 即输入层，hidden layer 即隐含层，output layer 即输出层。

如右图，我们依旧对每个 x_i 赋以权重 w_i，对每个 SUM_i 赋以对应的 T_i，得到对应的输出 $x_{3,4,5}$，其中 $x_{3,4}$ 继续用作输入，x_5 才是最终输出的 y。



比如在 Iris 数据集中，输入层是 sepal length,sepal width,petal length,petal width，输出层是 setosa,versicolor, virginica。

神经网络体验网址：https://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=0&networkShape=5,6&seed=0.70553&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false

手写数据集体验网址：https://www.3blue1brown.com/lessons/neural-networks

4.误差逆传播算法

BP 算法(误差逆传播 Error Back Propagation)分为两个过程：

①正向传播：输入样本从输入层经隐含层至输出层，如果输出层的实际输出与样本标签(期望输出)不符，则转至误差反向传播，反之结束算法。

②反向传播：将输出误差(期望输出值减实际输出值)按原通路反向传递，经隐含层至输入层。反向传递过程中将误差分摊给各层的各个神经元，依据各神经元的误差通过梯度下降法调整神经元的 w,T 以减小误差。

反复执行①②直至达到预定训练次数或误差小于预设值。

不足：训练时间长、局部最优。

5.深度学习

深度学习即很深层的神经网络。

由于层数过多，BP 算法难收敛，因此有如下方法训练：

①无监督逐层训练：先预训练(按网络的层级结构对网络内部参数进行分组，对每组参数训练产生在局部较好的参数值)，然后将这些局部最优的结果联合起来使用 BP 算法对整个网络参数微调。

②权值共享：让一组神经元使用相同的权重。比如 CNN 就使用卷积核滑动。

常见的深度学习方法：

卷积神经网络 CNN(Convolutional Neural Network)：前向反馈神经网络，使用局部感受野，认为图像中局部范围内的像素之间联系较为紧密，距离较远的像素之间相关性较弱。所以每个神经元只对局部进行感知，然后在更高层将局部信息综合起来得到全局信息。

循环神经网络 RNN(Recurrent Neural Network)：让上一层的输入作为下一层的输入。

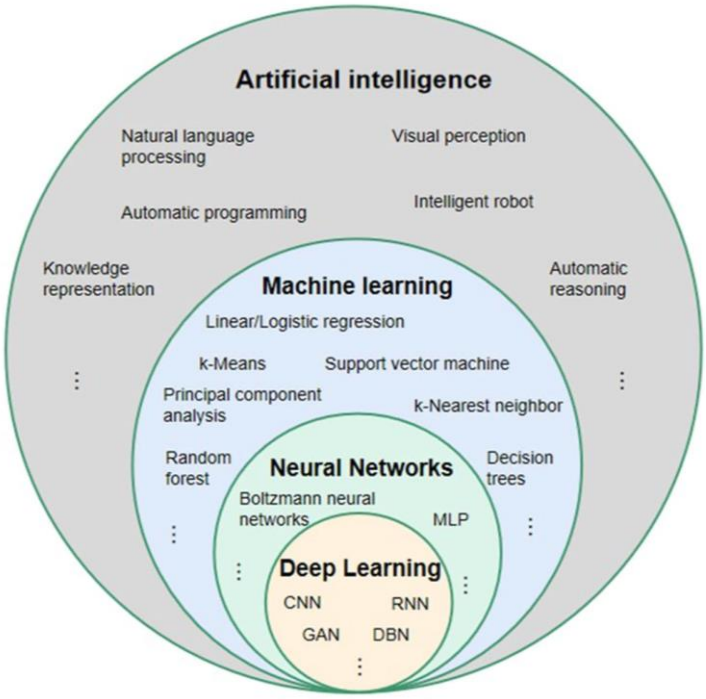
生成式对抗网络 GAN(Generative Adversarial Network)：生成模型 Generative Model 尽量生成真实的图片去欺骗判别模型 Discriminative Model，判别模型则尽量将生成图片和真实图片区分开来，形成一个动态的博弈过程。

自编码器 AE(AutoEncoder)：通过编码器对输入信息降维(映射到低维空间，也称潜在空间)，获取输入信息的表征(即表达方式)。解码器将降维的特征重新恢复到输入信息。编码器有 SAE、DAE、CAE、VAE。

深度置信网络 DBN(Deep Belief Nets)：一种生成模型，通过训练权重 w，让整个神经网络以最大概率生成训练数据。组成元件是受限玻尔兹曼机 RBM。RBM 是一种神经感知器，由一个显层(用于输入训练数据)和一个隐层(用作特征检测器)构成，二者神经元之间为双向全连接，内部无互连神经元。DBN 由 RBM 串联组成，上一个 RBM 的输出是下一个 RBM 的输入。

图卷积网络 GCN(Graph Convolution Network)GCN 是 CNN 在图数据上的推广，同时对节点特征信息和结构信息进行端对端学习。面对拓扑图(关系型数据)每个顶点的相邻顶点数都可能不同，没办法使用相同尺寸的卷积核进行卷积操作，需要使用空间域与谱域。空间域直接将卷积操作定义在每个节点的连接关系上，谱域是对图进行傅里叶变换后再卷积。

6.机器学习&神经网络&深度学习的关系图


Artificial intelligence
Natural language processing
Visual perception
Automatic programming
Intelligent robot
Knowledge representation
Automatic reasoning
Machine learning
Linear/Logistic regression
k-Means
Support vector machine
Principal component analysis
k-Nearest neighbor
Random forest
Decision trees
Neural Networks
Boltzmann neural networks
MLP
Deep Learning
CNN
RNN
GAN
DBN

PageRank 算法

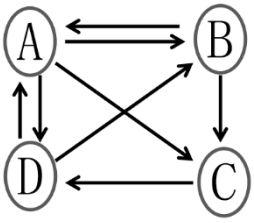
基本思想：

不只考虑词项，还考虑指向该网页的链接情况。如果一个网页被很多其他网页链接到，说明这个网页比较重要，PageRank 值较高。被越多优质(PageRank 高)的网页所指的网页，它是优质的概率就越大。

边上的权值表示网页之间随机跳转的概率(浏览者从当前网页点击到当前网页所指向网页的概率)。

如下例：

转移矩阵 $M=\begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & 1 & 0 \end{pmatrix}$，初始 T0 状态 Rank 值 $V_0=\begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}$。



第一轮：$T_0 \to T_1$

$$V_1=MV=\begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{pmatrix}=\begin{pmatrix} \frac{1}{2}\bullet\frac{1}{4}+\frac{1}{2}\bullet\frac{1}{4} \\ \frac{1}{3}\bullet\frac{1}{4}+\frac{1}{2}\bullet\frac{1}{4} \\ \frac{1}{3}\bullet\frac{1}{4}+\frac{1}{2}\bullet\frac{1}{4} \\ \frac{1}{3}\bullet\frac{1}{4}+1\bullet\frac{1}{4} \end{pmatrix}$$

第二轮：$T_0 \to T_1 \to T_2$

$$V_2=M^2V=\begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & 1 & 0 \end{pmatrix}^2\begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{pmatrix}$$

……

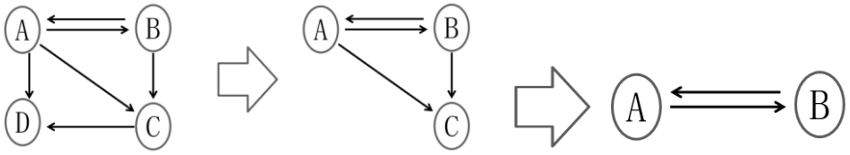
第 k 轮：$T_0 \to T_1 \to T_2 \to \cdots \to T_k$

$V^k=M^kV$

由于 M^k 是收敛的，所以迭代到一定次数后 V^k 趋于稳定。

PageRank 有效避免了词项作弊，但存在问题：

(1)算法本身的问题： Dead End 问题。这里可以把 D 节点去掉，但 C 也成了 dead end，所以 C 也要去掉。



(2)作弊问题：

①链接农场问题：

由互联网中的一部分网页组成，这些网页非常密集地互相连接在一起。通过创建一个堆砌大量链接而没有实质内容的网页，这些链接彼此互链或指向特定网站，以提高某个或者某些特定网页的 pagerank 值为目的。

②交换链接：

交换链接是指网站之间人为地互相增加对方网站的链接，是增加外链成本最低和使用最多的一种方法。由于交换链接可以增加网站的外链，提高网站的 pagerank 值。

③黄金链：

指一些高权重的网站出售首页的链接给作弊网站，提高作弊网站的 pagerank 值。