第一章

一，数据挖掘的概念：

数据挖掘是在**大型数据库**中**自动**发现**有用信息**的**过程**。简单地讲就是从大量数据中挖掘或抽取出知识【课本 p2-1.1 什么是数据挖掘】

（比如：根据有一个点集合(x,y)，求这点集合的函数表达式 y=f(x)，给一个 x 可以得到 y）

（比如：根据有一个点集合(x,y)，对这个点集合进行分类，是猫还是狗）

二，数据挖掘技术的演化：

数据挖掘最开始是知识发现（KDD-知识发现 knowledge discovery database）的核心部分。

*（**KDD 框架**在生活中经常**“应用于 web 的可控多兴趣推荐框架”**，经常买书的话会给你推荐书架，买了主板后会给你推送硬盘，看了土味视频后主页推荐全是土味视频。可以认为数据挖掘最开始就是这个**“兴趣框架 KDD——知识发现”的核心部分**）*

总体来说，数据挖掘融合了数据库、人工智能、机器学习、统计学、高性能计算、模式识别、神经网络、数据可视化、信息检索和空间数据分析等多个领域的理论和技术，是 21 世纪初期对人类产生重大影响的十大新兴技术之一。【课本 p2-1.1 什么是数据挖掘 p4-1.3 数据挖掘的起源，防止写论述题】

三，解释“知识发现的过程包含了数据清洗，数据集成，数据选择，数据转换，数据挖掘，模式评估和知识表现”什么意思：

(1)知识发现：从各种信息中，根据不同的需求获得知识的过程。*（比如：从一组点集合中找到对应函数——和数据挖掘概念中举的例子一样）*

(2)数据清洗：就是把错的，重复的，残缺的数据修复或者清理掉*（有些时候因为操作不当有些数据缺字错字什么的，还有时候年龄，身高被填成负数，都需要删除或者修改才能使用）*

(3)数据集成：把不同来源、格式、特点性质的数据整合在一起（逻辑上或物理上有机地集中）。*（可以认为数据都是散开的东一个西一个，我需要计算时就要把散开的数据集中）*

(4)数据转换：有时候数据输入维度太高了需要转化和降维。*（比如：图片那么多像素点需要降低分辨率这样减小计算压力）*

(5)数据准备：了解这些数据干嘛的用户要什么。

(6)数据选择：根据**(3)数据准备**过程用户对数据的要求，从**浩如烟海的(3)数据集成集合**中挑选需要的数据。

【课本上不存在该部分内容，这块老师给了一个句子，理解就好防止要求写论述题】

四，数据挖掘：

(1)分类：目的是构造**一个分类函数或分类模型**（也常常称作**分类器**），该模型能把数据库中的数据项映射到给定类别中的某一个。数据分类为数据挖掘中常见的功能之一，顾名思义即是将分析对象依不同的属性分类加以定义，建立不同的类组。数据挖掘中的分类是指针对未发生的结果进行预测分类，主要包括归纳和推论两步骤，其主要目的在于提高分类的准确度，

建立分类规则，再评估准则的优劣。**常用"判定树"算法。**

（比如：银行部门根据以前的数据将客户分成了不同的类别，现在就可以根据这些来区分新申请贷款的客户，以采取相应的贷款方案。）

(2)聚类：是把整个数据库分成不同的群组。它的目的是使群与群之间的数据**差别很明显**，而同一个群之内的各个数据互相相似。数据聚类主要是利用数据中类似或相同的项目，将同构型较高的数据区隔为不同的聚类，聚类内数据相似度越高越好，聚类间差异度越大越好。在一大群的研究对象中，根据不同的研究目的必定会有异质化的现象，但异质化的现象可能是几个同质化的群组所造成，数据聚类的主要目的便是将不同的同质化的组别差异找出来，**常用"判别分析"与聚类分析"算法。**

（单纯就是相似的东西化为一组。比如：判断两个人长得像不像，遇到双胞胎很像就化为一组，而两个陌生人完全不像就应该在不同的组别。）

(3)关联分析：是寻找数据库中值的相关性。两种常用的技术是关联规则和序列模式。
关联规则是寻找在同一个事件中出现的不同项的相关性；
序列模式与此类似，寻找的是事件之间时间上的相关性；
数据关联分组主要用来发现数据中特征属性间具有高度关联性的一种模式，其所发现的模式通常是用规则来表现。常用"关联规则（又称购物蓝分析）"算法。

（比如：奶粉和啤酒放一起卖案例，就是奶粉和啤酒某种程度上关联度高：爸爸卖奶粉顺便买瓶酒）

(4) 数据估计：根据不同相关属性数据的连续性数值，找出各属性间的关联性，以了解并获得某一特定属性未知的连续性数值，常用"回归分析"及"类神经网络算法"。分类用于预测**一个离散的数据的未来**（数据都是点，是离散的）；回归是预测**一个连续的数据的未来**（数据像股市一样一个连续不断的函数，回归方程就是拟合这个函数）

(5)异常检测：在聚类过程中找出离群点*（大家都在一起就它自己一个人孤零零的）*

【课本 p3-1.2 数据挖掘要解决的问题　课本 p5-1.4 数据挖掘任务 课本 p5-图 1.3 四种主要的数据挖掘任务】

五，数据挖掘系统：

数据挖掘系统（data mining system）是指从存放在数据库、数据仓库或其他信息库中的大量数据中挖掘出有趣知识的系统。*（就是一个由前端网页，后端服务器，服务器里面的数据挖掘算法模型等等一系列组成的一个服务体系。——称为数据挖掘系统。）*

一个结构合理的数据挖掘系统应该具有以下几个特点：

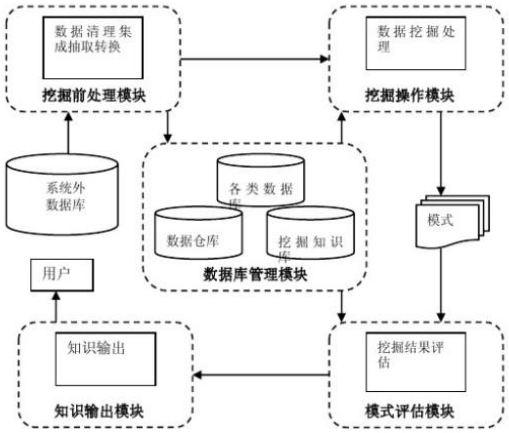
(1)系统功能和辅助工具的完备性；
(2)系统的可扩展性；
(3)支持多种数据源；
(4)对大数据量的处理能力；
(5)良好的用户界面和结果展示能力；

当前出现的数据挖掘系统主要包括**集中式**的和**分布式**的数据挖掘系统，而每种系统的具体结构及其各个组成部分却有多种不同的实现技术和实现方式。

【课本上没有，了解即可，防止论述题】

(6)※上述系统模块进行组合就是数据挖掘的体系架构：

数据挖掘系统由各类数据库、挖掘前处理模块、挖掘操作模块、模式评估模块、知识输出模块组成，这些模块的有机组成就构成了数据挖掘系统的体系结构。



数据挖掘系统的体系结构图

第二章

一、数据预处理对数据仓库和数据挖掘来说都是大问题：

数据预处理：指在主要的处理以前对数据进行的一些处理。

数据预处理的主要思想和方法有：

(1)聚集：将多个对象合并成一个对象*（比如一个商店一个月之中每一天的销售额(定量属性)数据，此处就可以用求和、求平均来实现聚集，而如商品名称(定性属性)可以忽略或汇总来处理）*

(2)抽样：统计学中的抽样目的是实现数据的调查与分析；数据挖掘中抽样的目的是压缩数据量，减少数据挖掘算法的资源开销

※关于抽样的方法有随机抽样、分层抽样、渐进抽样

※随机抽样：有两种变种，即无放回抽样和有放回抽样；

※分层抽样：如果数据总体由不同类型的对象组成，且每种类型的对象数据差别较大则使用分层抽样。*（如班级里面男生 20 人，女生 10 人，对这种类型的目标抽样就可以采用分层抽样，其中分层抽样又分为两种方法，即等个数抽样（示例中男生抽取 5 人，女生抽取 5 人）和等比例抽样（示例中男生抽取 10 人，女生抽取 5 人）;）*

※渐进抽样：样本容量不确定，需要自适应，如预测一个线性回归模型，其中抽取的样本容量逐渐增大，直到模型准确率稳定之后就停止抽样。

(3)维归约：减少所考虑的随机变量或属性的个数，即删除不相干的属性和维数减少数据量*（如一个大量数据的文档，维归约方法用来删除一些用不到数据）*。

方法有小波变换、主成分分析（PCA）*（此处实验有做过）*，把原始数据变换或投影到较小的空间。属性自己选择是一种维归约方法，其中不相关、弱相关或冗余的属性或维被检测或删除。*（其中维归约是数据规约之一，数据规约分为维归约、数据压缩、数值归约）*

(4)特征子集选择：简单来说就是将原始数据转化为有用的特征，更好的表示预测模型处理的实际问题，提升对于未知数据的预测准确性。*（如鸢尾花数据集中从花萼长度、花萼宽度、花瓣长度、花瓣宽度四个特征之中选择花萼长度、花萼宽度分析）*

(5)特征创建：由原来的属性创建新的属性集。其中创建新属性的相关方法有

※特征提取和映射数据到新的空间。

※特征提取：原始数据创建新的特征集称作特征提取*（如照片集合，按照照片是否为人脸进行分类，原始数据是像素集合，而其中一些提供的特征，如眼睛、鼻子、嘴巴这些特征为特征集）*

※映射数据到新的空间：此处涉及一个算法，即傅里叶分析

(6)离散化和二元化：

※离散化：将连续属性变换成分类属性*（如年龄按照 0-10，10-20，20-30……分类）；*

※二元化：连续和离散属性可能都需要变换成一个或多个二元属性*（二元化是针对类别数据进行划分的，比如颜色、性别，其中离散化后的数据相当于是类别数据（每个区间 1 个类别），可再进行二元化）*

(7)变量变换：变量的所有值的变换

※简单变换：使用一个简单数学函数分别作用于每一个值*（如 log 转换，求绝对值，求倒数）*

※标准化或规范化：目标是使整个值的集合具有特定的性质。**其目的有：**

※提升模型精度：标准化/归一化后，不同维度之间的特征在数值上有一定比较性，可以大大提高分类器的准确性；

※加速模型收敛：标准化/归一化后，最优解的寻优过程明显会变得平缓，更容易正确的收敛到最优解；

二、描述性数据汇总是预处理对数据质量的要求：

(1)描述性数据汇总：对于许多数据预处理任务，用户希望知道关于数据的中心趋势和离中趋势特征。中心趋势度量包括均值、中位数、众数、中列数，而数据离中趋势度量包括四分位数、四分位数极差和方差。

(2)数据质量：课本上关于数据质量的内容介绍在 24 页，此关键词重点应该在于数据质量分析

※数据质量分析的目的：保证数据的正确性、保证数据的有效性

※数据质量分析的内容

※在数据的正确性分析方面：

※缺失值：缺失数据包括空值或编码为无意义的指（null）；

※数据错误：通常是在输入数据时，造成的排字错误；

※度量标准错误：正确输入但因为不正确的度量标准而导致的错误数据；

※编码不一致：通常包括非标准度量单位或不一致的值。*（例如，同时使用 M 和 male 表示性别。）*

※在数据的有效性方面：

关注数据统计方面的信息*（如占比、方差、均值、分位数等，以此来了解这些数据包含的信息度程度）*

三，数据预处理：数据清洗、数据集成、数据规约、特征提取、离散化处理：

(1)数据清洗：按照一定的规则把“脏数据”“洗掉”，这就是数据清洗。

※需要清洗数据的主要类型：残缺数据、错误数据、重复数据

※残缺数据：这一类数据主要是一些应该有的信息缺失，如供应商的名称、分公司的名称、客户的区域信息缺失、业务系统中主表与明细表不能匹配等。

※错误数据：这一类错误产生的原因是业务系统不够健全，在接收输入后没有进行判断直接写入后台数据库造成的，比如数值数据输成全角数字字符、字符串数据后面有一个回车操作、日期格式不正确、日期越界等。

※重复数据：对于这一类数据——特别是维表中会出现这种情况——将重复数据记录的所有字段导出来，让客户确认并整理。数据清洗是一个反复的过程，不可能在几天内完成，

只有不断的发现问题，解决问题。

※数据清洗的内容：一致性检查、无效值和缺失值处理

※一致性检查：一致性检查是根据每个变量的合理取值范围和相互关系，检查数据是否合乎要求，发现超出正常范围、逻辑上不合理或者相互矛盾的数据。*(例如，用 1-7 级量表测量的变量出现了 0 值，体重出现了负数，都应视为超出正常值域范围。)*

※无效值和缺失值处理：由于调查、编码和录入误差，数据中可能存在一些无效值和缺失值，需要给予适当的处理。**常用的处理方法有：估算，整例删除，变量删除和成对删除。**

※估算：最简单的办法就是用某个变量的样本均值、中位数或众数代替无效值和缺失值；

※整例删除：剔除含有缺失值的样本；

※变量删除：如果某一变量的无效值和缺失值很多，而且该变量对于所研究的问题不是特别重要，则可以考虑将该变量删除；

※成对删除：用一个特殊码*(通常是 9、99、999 等)*代表无效值和缺失值，同时保留数据集中的全部变量和样本。

(2)数据集成：指把数据从多个数据源整合在一起，提供一个观察这些数据的统一视图的过程。建立数据仓库的过程实际上就是数据集成。

数据集成中的两个主要问题是数据结构和数据冗余：

※数据结构：如何对多个数据集进行匹配，当一个数据库的属性与另一个数据库的属性匹配时，必须注意数据的结构

※数据冗余：两个数据集有两个命名不同但实际数据相同的属性，那么其中一个属性就是冗余的

(3)数据规约：指在尽可能保持数据原貌的前提下，最大限度地精简数据量。

※数据挖掘时往往数据量非常大，在少量数据上进行挖掘分析需要很长的时间，数据归约技术可以用来得到数据集的归约表示，它小得多，但仍然接近于保持原数据的完整性，并结果与归约前结果相同或几乎相同。通常有维归约、数值归约。

维归约指通过减少属性的方式压缩数据量，通过移除不相关的属性，可以提高模型效率

（维归约最开始的数据预处理有介绍）

※常见的维归约方法有：分类树、随机森林通过对分类效果的影响大小筛选属性；小波变换、主成分分析通过把原数据变换或投影到较小的空间来降低维数。

※数值归约用较小的数据表示形式替换原始数据。代表方法为对数线性回归、聚类、抽样等。

第三章

一，关联规则挖掘相关基本概念和路线图：

关联规则挖掘：目标是发现数据项集之间的关联关系或相关关系。

（关联规则挖掘的一个典型例子是购物篮分析，关联规则挖掘有助于发现交易数据库中不同商品项之间的关系，找出顾客购买行为模式，如购买了某一商品对购物对其他商品的影响。分析结果可以应用于商品货架布局、货存安排以及根据购买模式对用户进行分类。）

关联规则挖掘基本概念：

(1)项与项集：数据库中不可分割的最小单位信息，称为项目，用符号 i 表示。项的集合称为项集。*设集合 I={i1, i2, ..., ik}是项集，I 中项目的个数为 k，则集合 I 称为 k-项集。*

(2)事务：*设 I={i1, i2, ..., ik}是由数据库中所有项目构成的集合，一次处理所含项目的集合用 T 表示，T={t1,*

t2, ..., tn}。每一个包含 ti 子项的项集都是 I 子集。

(3)项集的频数（支持度计数）：包括项集的事务数称为项集的频数（支持度计数）

(4)关联规则：*关联规则是形如 X=>Y 的蕴含式，其中 X、Y 分别是 I 的真子集，并且 X∩Y=∅。X 称为规则的前提，Y 称为规则的结果。关联规则反映 X 中的项目出现时，Y 中的项目也跟着出现的规律。*

(5)关联规则的支持度（Support）：*关联规则的支持度是交易集中同时包含的 X 和 Y 的交易数与所有交易数之比，记为 support(X=>Y)，即 support(X=>Y)=supportX∪Y=P(XY)。支持度反映了 X 和 Y 中所含的项在事务集中同时出现的概率。*

（支持度定义：表示同时包含 A 和 B 的事务占所有事务的比例。如果用 P(A) 表示包含 A 的事务的比例，那么 Support = P（A & B)）

(6)关联规则的置信度（Confidence）：关联规则的置信度是交易集中包含 X 和 Y 的交易数与所有包含 X 的交易数之比，记为 confidence(X=>Y)，即：confidence(X=>Y)==P(Y|X)。置信度反映了包含 X 的事务中，出现 Y 的条件概率。

（置信度定义：表示包含 A 的事务中同时包含 B 的事务的比例，即同时包含 A 和 B 的事务占包含 A 的事务的比例。公式表达：Confidence = P（A & B）/ P(A)）

(7)最小支持度与最小置信度：通常用户为了达到一定的要求，需要指定规则必须满足的支持度和置信度阈限，*当 support(X=>Y)、confidence(X=>Y)分别大于等于各自的阈限值时，认为 X=>Y 是有趣的，此两个值称为最小支持阈值(min_sup)和最小置信阈值(min_conf)。其中，min_sup 描述了关联规则的最低重要程度，min_conf 规定了关联规则必须满足的最低可靠性。*

(8)频繁项集：*设 U={u1, u2, ..., un}为项目的集合，且 UI，U≠Ø，对于给定的最小支持度 min_sup，如果项集 U 的支持度 support(U)≧min_sup，则称 U 为频繁项集，否则，U 为非频繁项集。*

(9)强关联规则：*support(X=>Y)≧min_sup 且 confidence(X=>Y)≧min_conf，称关联规则 X=>Y 为强关联规则，否则为弱关联规则。*

二，有效的和可伸缩的频繁项集挖掘方法：

频繁项集：出现在许多事务中的有意义的联系（一般计算的时候支持度超过阈(yu)值就认为是频繁项集）其中针对频繁项集的计算一般是非常复杂的，因此有**三种方法可以降低产生频繁项集的计算复杂度。**

(1)减少候选项集的数目：*如先验(apriori)原理，是一种不用计算支持度而删除某些候选项集的方法*

(2)减少比较次数：*利用更高级得到数据结构或者存储候选项集或者压缩数据集来减少比较次数*

(3)减少事务数目：*随着候选项集的规模越来越大，能支持项集的事务越来越少*

三，Apriori 算法：

Apriori 算法使用基于支持度的剪枝技术，系统地控制候选集的指数增长。

***Apriori 算法原理：**

先验原理（Apriori）：使用支持度对候选项集进行剪枝；Apriori 旨在寻找频繁项集，以帮助数据挖掘者发现数据见的关联规则。通过连接生产候选项及其支持度然后剪枝生成频繁项集。

如果一个项集是频繁的，则它的所有子集也一定是频繁的，相反，如果一个项集是非频繁的，则它所有的超集都是非频繁的，这种基于支持度度量修剪指数搜索空间的策略称为基于支持度的剪枝，依赖于一个性质，即一个项集的支持度决不会超过它的自己的支持度，这个性质称为反之尺度度量的反单调性。【此处重点**】**

※Apriori 算法步骤：

（做大题建议先穷举，得出结果后再依照下面步骤匹配数据作答）

Step01 列出候选项集：*先列所有 1-项集并标记他们的支持度计数；*

Step02 剪枝：*根据支持度阈值剪掉所有非频繁项集；*

Step03 列出候选项集：*根据方法{(K-1)-项集*1-项集 或者 (K-1)-项集*(K-1)-项集}列出 2-项集并标记支持度计数；*

Step04 剪枝：*再次根据支持度阈值剪掉所有非频繁项集；*

Step05 重复 step03、step04 直到 k-项集的频繁项集为空；

生产候选项的方法：（所有项集内的项都要按一个顺序排列，避免答题时写出多个重复的）

※Apriori 算法涉及的参数：【重要考点】

事务：*每一行相当于一个事务；**项：**除第一列外，每一列的列名是一个项；**项集：**项的集合（0~n）；*

K-项集：*包含 k 个项的项集；*

关联分析：发现隐藏在大数据中的有意义的数据联系；

※支持度计数：包含指定项集的事务的总个数，就 σ(X)；

※支持度：支持度计数/事务总数 ；

※置信度：X->Y 的支持度计数/X 的支持度计数 ；

阈值：临界值；

※频繁项集：支持度大于等于支持度阈值的项集；

强规则：置信度大于等于置信度阈值的频繁项集；

※极大频繁项集：一个频繁项集的直接超集都是非频繁项集时，本频繁项集为极大频繁项集；

※闭项集：项集 X 所有直接超集的支持度计数都与项集 X 的支持度计数不同；

※闭频繁项集：一个闭项集是频繁项集时；

前件：*X->Y （X,Y 在同一个项集中），X 是前件；**后件：**同理，Y 是后件；*

※优缺点：

优点：Apriori 算法使用 Apriori 性质来生产候选项集的方法，大大压缩了频繁集的大小， 取得了很好的性能。

缺点：每一次迭代都要遍历整个数据进行计数计算和判断，运行效率低。

四，FP 模式增长算法：

※FP 算法原理：算法思想：

FP 树是一种输入数据的压缩表示。它逐条读入事务，并将事务映射到 FP 树中的一条路径上。由于不同事务可能存在若干相同的项，因此它们的路径可能部分重叠。越多的路径相互重叠，使用 FP 树结构获得的压缩效果越好。当 FP 树足够小时，就可以将其放入内存，而不必重复读取硬盘中的数据，进而加速频繁项集的计算。如果构成的 FP 树还是很大，可以采用分区投影的方法。采用自底向上、分治思想。

※FP 算法优缺点：

优点：项的排列方式从大到小，压缩的路径越大，越节省空间。

缺点：如果项的排列方式为从小到大，则花费的空间比原始数据还大（因为数组的表示还需要指针和附加空间）

第四章

一，分类与预测(回归)的概念：

(1)分类和预测(回归)的异同：*（这里的表述我更倾向于回归，数据预测分成两块，对离散数据是分类，对连续数据是回归，有时候也简称为预测）*

分类和预测(回归)是两种使用**数据进行预测的方式**，**可用来确定未来的结果。**

分类是用于预测数据对象的离散类别的，需要预测的属性值是离散的、无序的。

预测(回归)则是用于预测数据对象的连续取值的，需要预测的属性值是连续的、有序的。

(2)分类概念：分类算法反映的是如何找出同类事物的共同性质的特征型知识和不同事物之间的差异性特征知识。分类是通过有指导的学习训练建立分类模型，并使用模型对未知分类的实例进行分类。分类输出属性是离散的、无序的。

分类过程是：第一步是模型建立阶段，或者称为训练阶段，第二步是评估阶段。

(3)预测(回归)概念：可以看作一个映射或者函数 y=f(x)，其中，x 是输入元组，输出 y 是连续的或有序的值。与分类算法不同的是，预测算法所需要预测的属性值是连续的、有序的，分类所需要预测的属性值是离散的、无序的。【课本 p65~p66-3.1 课本】

二，决策树分类：*（决策树就是一个有智力的东西**决定对自己接下来要干什么的一种树**：遇到什么情况应该干什么，接下来遇到什么情况应该干什么）*

(1)决策树定义：决策树归纳是从类标记的训练数据构建决策树，属于分类领域。遍历根节点到全部叶节点的路径，每条路径都属于一个元组分类。整棵决策树形成分类规则。

ID3 算法中根据特征选择和信息增益评估，每次选择信**息增益最大**的**特征作为分支标准。**

ID3 算法可用于划分标称型数据集，没有剪枝的过程，为了去除过度数据匹配的问题，可通过裁剪合并相邻的无法产生大量信息增益的叶子节点。

【课本 p69-3.3 决策树分类器】

(2)ID3 算法：

※基本思想是：首先计算出原始数据集的**信息熵**，然后**依次**将数据中的**每一个特征作为分支标准**，并计算其相对于原始数据的**信息增益**，选择**最大信息增益的分支标准来划分数据**，因为信息增益越大，区分样本的能力就越强，越具有代表性。重复上述过程从而生成一棵决策树，很显然这是一种自顶向下的贪心策略。【书上没有这块假如问**思想是什么**就上面这坨了】

※求属性的信息熵（度量一个属性的信息量）：

案例：一个属性（表中一列）假如有两种可能情况。要么 a 要么 b。一共 14 行，其中有 9 行是 a，有 5 行是 b。有如下公式：

$$Entropy(S) = Entropy(p_1, p_2, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

$$Entropy(S) = Entropy(\frac{9}{14}, \frac{5}{14}) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$

其中该属性几种可能性就几项，每项的规则如上，比葫芦画瓢吧。注意每项开头都有负号，log2 是固定的无论多少项情况怎么变都是 log2

（熵是数据的混乱程度，熵越小表明目标属性混乱程度低，反之高。先求训练集输出属性的信息熵）

※求信息增益：

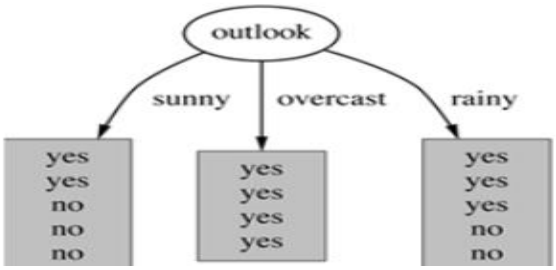
案例：具体情况比葫芦画瓢吧，这个实在不好描述😂

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rain	mild	high	false	yes
rain	cool	normal	false	yes
rain	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rain	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rain	mild	high	true	no

接下来以天气的例子来说明 ID3 算法。左图是样本数据集，每个样本包括 4 个特征“Outlook”，“Temperature”,”Humidity”和“Windy”， 模型的分类目标是 play 或者 not play。

*表中一共包含 14 个样本，包括 9 个正样本和 5 个负样本，并且是一个二分类问题，要么玩要么不玩，那么当前**输出值 Play 的信息熵**的计算如下：**上一个分支讲过这块了——怎么求信息熵的***

$$Ent(D) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.940286$$



*接下来以表中的 **Outlook** 属性作为分支标准，根据 **sunny**、**overcast**、**rain** 这三个属性值可将数据分为三类，如左图所示：*

引入该分支标准后，数据被分成了 3 个分支，每个分支的信息熵计算如下：

$$H(D^{sunny}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.970951$$

$$H(D^{overcast}) = -\frac{4}{4}log_2\frac{4}{4} = 0$$

$$H(D^{rain}) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5} = 0.970951$$

因此，基于该分支标准 T 所带来的信息增益为：

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V}\frac{|D^v|}{|D|}Ent(D^v) = 0.940286 - \frac{5}{14}\cdot 0.970951 + \frac{4}{14}\cdot 0 + \frac{5}{14}\cdot 0.970951 = 0.24675$$

【课本 p73-3.3.3-1 求熵，求基尼系数，求分类误差 课本 p77-3.3.3-7 求增益率 （上面是对关键的两个值的运算)】

※ID3 算法的流程：

还是上面的案例

该活动无法进行的概率是：5/14

该活动可以进行的概率是：9/14

因此样本集合的信息熵是：-5/14log(5/14) - 9/14log(9/14) = 0.940

接下来我们再看属性 outlook 信息熵的计算：

outlook 为 sunny 时，

该活动无法进行的概率是：3/5

该活动可以进行的概率是：2/5

因此 sunny 的信息熵是：-3/5log(3/5) - 2/5log(2/5) = 0.971

同理可以计算 outlook 属性取其他值时候的信息熵：

outlook 为 overcast 时的信息熵：0

outlook 为 rain 时的信息熵：0.971

---

*属性 outlook 的信息增益：gain(outlook) = 0.940 - (5/14*0.971 + 4/14*0 + 5/14*0.971) = 0.246*

相似的方法可以计算其他属性的信息增益：

gain(temperature) = 0.029

gain(humidity) = 0.151

gain(windy) = 0.048

信息增益最大的属性是 outlook。

根据 outlook 把样本分成 3 个子集，然后把这 3 个子集和余下的属性

作为输入递归执行算法。

（上述内容大致逻辑就是以信息增益/信息熵为度量，用于决策树结点的属性选择的标准，每次优先选取信息量最多（信息增益最大）的属性，即信息熵值最小的属性，以构造一颗熵值下降最快的决策树，到叶子节点处的熵值为 0。）***

【课本上没有需要考试之间看明白流程，题目就是上一小节的，上一小节计算的是熵和信息增益，这一小节才是具体步骤】

※ID3 的优缺点：

优点有：

ID3 算法避免了搜索不完整假设空间的一个主要风险；ID3 算法在搜索的每一步都使用当前的所有训练样例，大大降低了对个别训练样例错误的敏感性；ID3 算法在搜索过程中不进行回溯。所以，它易受无回溯的爬山搜索中的常见风险影响；*（收敛到局部最优而不是全局最优）*

缺点有：

ID3 算法只能处理离散值的属性；信息增益度量存在一个内在偏置，它偏袒具有较多值的属性；ID3 算法增长树的每一个分支的深度，直到恰好能对训练样例完美地分类，存在决策树过度拟合；【书上没有 ID3，问到了照抄这段即可】

(3)C4.5 算法：

※多算一个————信息增益率

※C4.5 算法流程：

学号	性别	学生干部	综合成绩	毕业论文	就业情况
1	男	是	70-79	优	已
2	女	是	80-89	中	已
3	男	不是	60-69	不及格	未
4	男	是	60-69	良	已
5	男	是	70-79	中	已
6	男	不是	70-79	良	未
7	女	是	60-69	良	已
8	男	是	60-69	良	已
9	女	是	70-79	中	未
10	男	不是	60-69	及格	已
11	男	是	80-89	及格	已
12	男	是	70-79	良	已
13	男	不是	70-79	及格	未
14	男	不是	60-69	及格	已
15	男	是	70-79	良	已
16	男	不是	70-79	良	未
17	男	不是	80-89	良	未
18	女	是	70-79	良	已
19	男	不是	70-79	不及格	未
20	男	不是	70-79	良	未
21	女	是	60-69	优	已
22	男	是	60-69	良	已

Step01

*求最后结果"就业情况"的**结果总信息熵**：*

$$\text{Entropy}(\text{就业情况}) = -\frac{14}{22}\log_2\frac{14}{22} - \frac{8}{22}\log_2\frac{8}{22} = 0.94566$$

（就业情况自己的混乱程度）

求性别中，男的对应所有结果（就业情况）的信息熵。女的对应的所有结果（就业情况）的信息熵：

$$\text{Entropy}(\text{男}) = -\frac{10}{17}\log_2\frac{10}{17} - \frac{7}{17}\log_2\frac{7}{17} = 0.97742$$

（男的抽出来组成表，然后其就业情况的混乱程度）

$$\text{Entropy}(\text{女}) = -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5} = 0.72193$$

（同上，女的抽出来，其就业情况的混乱程度）

*求对性别本身**男女的信息熵的加权平均**：*

$$\text{Entropy}(\text{性别}) = \frac{17}{22} * 0.97742 + \frac{5}{22} * 0.72193 = 0.91935$$

（上一步的男女信息熵乘上男女概率）

*求性别**信息增益**：*

$$\text{Gain}(\text{性别}) = 0.94566 - 0.91935 = 0.02631$$

*（就是总的综合信息熵减去**上一步的男女信息熵加权**）*

*求**分裂信息**：*

$$\text{Split\_Info}(\text{性别}) = -\frac{17}{22}\log_2\frac{17}{22} - \frac{5}{22}\log_2\frac{5}{22} = 0.77323$$

*（**就是对男女这一列求信息熵**）*

*求信息**增益率**：*

$$\text{Gain\_Ratio}(\text{性别}) = 0.02631/0.77323 = 0.03403$$

*（**分裂信息除以当前列的信息熵**）*

Step02** 求下一个属性（学生干部）**的上述各值以此类推

$$\text{Gain\_Ratio}(\text{学生干部}) = 0.40184/0.97602 = 0.41171$$

$$\text{Gain\_Ratio}(\text{综合成绩}) = 0.125763/1.422675 = 0.088391$$

$$\text{Gain\_Ratio}(\text{毕业论文}) = 0.200103/2.00103 = 0.10167158$$

Step03** 选择**最大**的**信息增益率**的属性，**为当前层的判断依据

※C4.5 算法优缺点：

优点：C4.5 是 ID3 的改进，采用信息增益率进行特征选择；C4.5 采用单点离散化的思想，用信息增益率来进行连续值特征的属性值选择；与 ID3 相类似，只不过将信息增益改为信息增益比，以解决偏向取值较多的属性的问题，另外它可以处理连续型属性。

缺点：C4.5 时间耗费大；C4.5 没解决回归问题；

三，贝叶斯分类算法：

※总结一下核心算法思想：通过对概率的比较，直接根据某些已知条件得出某个结论。同时

贝叶斯算法是一种利用概率统计知识进行分类的算法。在许多场合，朴素贝叶斯(Naïve Bayes，NB)分类算法可以与决策树和神经网络分类算法相媲美，该算法能运用到大型数据库中，而且方法简单、分类准确率高、速度快。*(比如一个人有三个属性，国籍年龄，是否打疫苗。那么计算得出一个印度+老人+没打疫苗的情况下是新冠的概率奇高——简称叠 buff。那么我们在遇见一个印度，老人，没打疫苗的。就可以直接认为得了新冠)*

※算法流程：

案例【比葫芦画瓢】

症状	职业	疾病
打喷嚏	护士	感冒
打喷嚏	农夫	过敏
头痛	建筑工人	脑震荡
头痛	建筑工人	感冒
打喷嚏	建筑工人	过敏
打喷嚏	教师	感冒
头痛	教师	脑震荡
打喷嚏	教师	过敏

例子：*某个医院早上收了八个门诊病人，如下表。*

现在又来了第九个病人，是一个打喷嚏的建筑工人。请问他患上感冒的概率有多大?

根据贝叶斯定理:

P(A|B) = P(B|A) P(A) / P(B)

可得满足"打喷嚏"和"建筑工人"两个条件下，感冒的概率如下:

P(感冒|打喷嚏 x 建筑工人)

= P(打喷嚏 x 建筑工人|感冒) x P(感冒) / P(打喷嚏 x 建筑工人)

假定"打喷嚏"和"建筑工人"这两个特征是独立的（即这两个条件没有相关性，比如不存在说他是建筑工人他打喷嚏的概率比较大或者比较小这种关系），因此，上面的等式就变成了。

P(感冒|打喷嚏 x 建筑工人)

= P(打喷嚏|感冒) x P(建筑工人|感冒) x P(感冒) / P(打喷嚏) x P(建筑工人)

通过统计可得:

P(感冒|打喷嚏 x 建筑工人)

= (2/3) x (1/3) x (3/8) / (5/8) x (3/8)

= (16/45)

通过贝叶斯公式算出了满足条件下感冒的概率，那么现在贝叶斯分类器如何实现呢?

接上面的例子，从上面我们得出了 P(感冒|打喷嚏 x 建筑工人）的值，那么我们可以再算出

P(不感冒|打喷嚏 x 建筑工人）的值，计算结果如下:

P(不感冒|打喷嚏 x 建筑工人)

= P(打喷嚏|不感冒) x P(建筑工人|不感冒) x P(不感冒) / P(打喷嚏) x P(建筑工人)

= (3/5) x (2/5) x (5/8) / (5/8) x (3/8)

= (16/25)

OK，现在我们知道来如果新来一个打喷嚏的建筑工人，他患感冒的几率是 P(感冒|打喷嚏 x 建筑工人)= (16/45)。不患感冒的几率是 P(不感冒|打喷嚏 x 建筑工人)= (16/25)。

最后通过对概率的比较，我们就可以将打喷嚏的建筑工人分类到"不感冒"人群中（不感冒的概率比较大)。

【课本 p126-1 贝叶斯定理 计算贝叶斯的公式 p130-例题 4.5 朴素贝叶斯分类器（算法流程，如果不清楚流程了一定要看 130 页!!!)】

※贝叶斯算法优缺点：

朴素贝叶斯的思想十分简单，对于给出的待分类项，求出在此项出现的条件下各个类别出现的概率，以概率大小确定分类项属于哪个类别。

优点有：朴素贝叶斯模型发源于古典数学理论，因此有着坚实的数学基础，以及稳定的分类效率；算法较简单，常用于文本分类；对小规模的数据表现很好，能够处理多分类任务，适合增量式训练。

缺点有：需要计算先验概率；对输入数据的表达形式很敏感；分类决策存在错误率。

四，神经网络分类：

※算法流程：

案例【比葫芦画瓢】

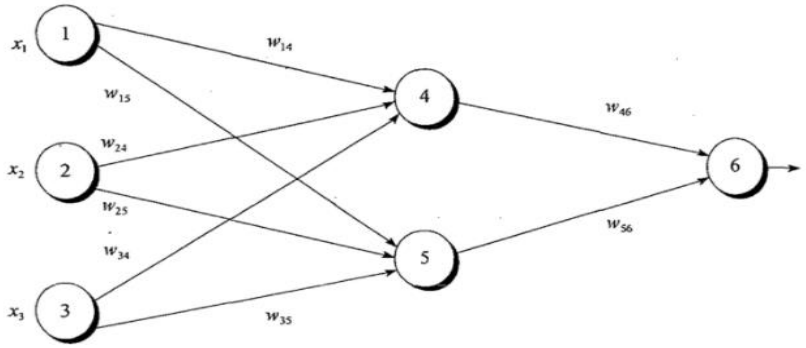
算法：后向传播。使用后向传播算法,学习分类或预测的神经网络。
输入：

- D：由训练元组和其相关联的目标值组成的数据集；
- L：学习率；
- *network*：多层前馈网络。

输出：训练后的神经网络。
方法：

```
(1)  初始化network的所有权重和偏倚。
(2)  while 终止条件不满足 {
(3)   for D中每个训练元组X {
(4)    // 前向传播输入
(5)    for每个输入层单元j {
(6)      O_j=I_j;    // 输入单元的输出是它的实际输入值
(7)    for 隐藏或输出层的每个单元j {
(8)      I_j=∑_i w_ij O_i+θ_j; // 关于前一层i，计算单元j的净输入
(9)      O_j=1/(1+e^-I_j) ; } // 计算单元j的输出
(10)   // 后向传播误差
(11)   for 输出层的每个单元j
(12)     Err_j=O_j(1-O_j)(T_j-O_j) ; // 计算误差
(13)   for 由最后一个到第一个隐藏层，对于隐藏层的每个单元j
(14)     Err_j=O_j(1-O_j) ∑_k Err_k w_jk;// 计算关于下一个较高层k的误差
(15)   for network中的每个权w_ij {
(16)     ΔW_ij=(l) Err_j O_i;  // 权重增量
(17)     W_ij=W_ij+ΔW_ij; } //权重更新
(18)   for network中每个偏倚θ_j {
(19)     Δθ_j=(l) Err_j;   //偏倚增量
(20)     θ_j=θ_j+Δθ_j; } //偏倚更新
(21) }}
```

以一个案例为准就行说明；以第一个训练元祖 X={1，0，1} 其类标号为 1



Step01 *初始化 network 的所有权重和偏倚；网络的权重一般初始为小**随机数**（例如 -1.0 到 1.0），具体初始数据如下：（开始的时候需要随机初始化各层权重和偏倚）*

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1

Step02 *在终止循环条件下循环每个训练元祖，设 X={1，0，1}，其类标号 Y 为 1；（就是循环执行此步骤之后的一切步骤）*

Step03 *前向传播*

单元 j	净输入 I_j	输出 O_j
4	$0.2+0-0.5-0.4=-0.7$	$1+(1+e^{0.7})=0.33$
5	$-0.3+0+0.2+0.2=0.1$	$1+(1+e^{-0.1})=0.525$
6	$-(0.3)(0.332)-(0.2)(0.525)+0.1=-0.105$	$1+(1+e^{0.105})=0.474$

（很明显上表损失函数是 sigmoid 函数）【p144-4.7.1 感知机 4.7.2 多层神经网络】

***Step04** 损失函数计算*

求激活函数的导数：

上面提到了激活函数是 sigmoid 函数

$$O_j = \frac{1}{1+e^{-I_j}}$$

求导之后为：

$O_j(1-O_j)$ *（这也就是为啥要找<u>可导</u>的激活函数用于神经网络）*

输出层有：

$$Err_j = O_j(1-O_j)(T_j-O_j) \tag{9.6}$$

其中，O_j 是单元 j 的实际输出，而 T_j 是 j 给定训练元组的已知目标值。注意，$O_j(1-O_j)$ 是逻辑斯缔函数的导数。

隐藏层有：

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk} \tag{9.7}$$

其中，w_{ik}是由下一较高层中单元 k 到单元 j 的连接权重，而 Err_k 是单元 k 的误差。

计算结果如下：

单元 j	Err_j
6	$(0.474)(1-0.474)(1-0.474)=0.1311$
5	$(0.525)(1-0.525)(0.1311)(-0.2)=-0.0065$
4	$(0.332)(1-0.332)(0.1311)(-0.3)=-0.0087$

***Step05** 权值更新*

更新权重和偏倚，以反映误差的传播。权重用下式更新，其中，Δw_{ij}是权 w_{ij}的改变量。

$$\Delta w_{ij} = (l)Err_j O_i \tag{9.8}$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \tag{9.9}$$

（l 就是学习率）

结果如下：

权重或偏差	新值
w_{46}	$-0.3+(0.9)(0.1311)(0.332)=-0.261$
w_{56}	$-0.2+(0.9)(0.1311)(0.525)=-0.138$
w_{14}	$0.2+(0.9)(-0.0087)(1)=0.192$
w_{15}	$-0.3+(0.9)(-0.0065)(1)=-0.306$
w_{24}	$0.4+(0.9)(-0.0087)(0)=0.4$
w_{25}	$0.1+(0.9)(-0.0065)(0)=0.1$
w_{34}	$-0.5+(0.9)(-0.0087)(1)=-0.508$
w_{35}	$0.2+(0.9)(-0.0065)(1)=0.194$
θ_6	$0.1+(0.9)(0.1311)=0.218$
θ_5	$0.2+(0.9)(-0.0065)=0.194$
θ_4	$-0.4+(0.9)(-0.0087)=-0.408$

【p148-149 损失值计算】

※神经网络分类算法优缺点：

优点：分类准确率高；并行处理能力强；分布式存储和学习能力强；鲁棒性较强，不易受噪声影响；

缺点：需要大量参数（网络拓扑、阀值、阈值）；结果难以解释；训练时间过长；

五，欠拟合过拟合解决&十折交叉：

(1)解决过拟合欠拟合：

※重新清洗数据，导致过拟合的一个原因也有可能是数据不纯导致的，如果出现了过拟合就需要我们重新清洗数据。

※增大数据的训练量，还有一个原因就是我们用于训练的数据量太小导致的，训练数据占总数据的比例过小。

※采用 dropout 方法。这个方法在神经网络里面很常用。dropout 方法是 ImageNet 中提出的一种方法，通俗一点讲就是 dropout 方法在训练的时候让神经元以一定的概率不工作。

(2)十折交叉：我们将数据集随机分成 10 份，使用其中 9 份进行训练而将另外 1 份用作测试。该过程可以重复 10 次，每次使用的测试数据不同。【课本上没有别翻了】*（就是 9：1 分组后轮流着作为机器学习的训练集和验证集）*

第五章

一，聚类分析概念：

聚类分析指将物理或抽象对象的集合分组为由类似的对象组成的多个类的分析过程。聚类分析的目标就是在相似的基础上收集数据来分类。

聚类技术通常又被称为无监督学习，与监督学习不同的是，在簇中那些表示数据类别的分类或者分组信息是没有的。数据之间的相似性是通过定义一个距离或者相似性系数来判别的。聚类分析就是把数据进行分组划分成一些有意义或者有用的组（簇），便于在数据汇总中减少数据量。聚类分析的主要目标就是将数据对象分组，同组的是相关的，不同组是不相关的，组内相关性更高，不同组的差别越大，聚类效果就越好。*(聚类实际上可以看作是一个分类，但它与分类也有区别，聚类要求划分的类是不知道的。或者理解为，聚类就是为了把相似的东西放在一起。例如，有三个人其中两个双胞胎，外加一个路人。已知我们需要把这三个人分成两组，聚类就会自动的将长得相似的双胞胎化成一组，路人另外一组）*【p307-7.1.1 什么是聚类分析】

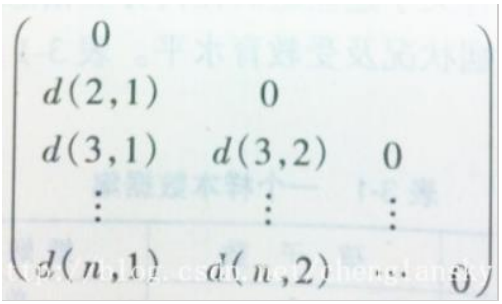
二，聚类分析中涉及到的数据类型：

(1)相异度：

欧几里得距离 *（用欧几里得距离——**向量间的距离** 表示两个向量的相异程度）*

$$d(x,y) := \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}.$$

(2)相异度矩阵：



存放 n 个对象两两之间的相异度的 n*n 个矩阵
其中 d(i,j)表示对象 i 和对象 j 之间的相异度的数值，越相似越接近于 0；反之，其值越大。
显然为下三角矩阵，d(i,j) =d(j,i)
(3)区间标度度量:是一个粗略线性标度的连续度量。*(比如重量、高度)*
(4)二元变量:就是只有两种状态：0 或者 1。*(比如：一个人能有没有打疫苗，0 就是没打，1 就是打了，且一个对象可以包含多个二元变量)*
(5)标称变量:是二元变量的推广，它可以具有不只是两个的状态值。*(比如说红、绿、蓝、黄。对于标称型变量，值排列顺序不重要，可以不去管。)*

三，聚类算法的类型：

(1)基于划分的聚类方法:
基于划分的聚类方法是一种自顶向下的方法，对于给定的 n 个数据对象的数据集 D，将数据对象组织成 k(k≤n) 个分区，其中，每个分区代表一个簇。
(2) 基于层次的聚类方法:
基于层次的聚类方法是指对给定的数据进行层次分解，直到满足某种条件为止。
(3) 基于密度的聚类方法
基于密度的聚类方法的主要目标是寻找被低密度区域分离的高密度区域。与基于距离的聚类算法不同的是，基于距离的聚类算法的聚类结果是球状的簇，而基于密度的聚类算法可以发现任意形状的簇。基于密度的聚类方法是从数据对象分布区域的密度着手的。如果给定类中的数据对象在给定的范围区域中，则数据对象的密度超过某一阈值就继续聚类。

【p308-7.1.2 聚类的不同类型】

四，K-means/k-medoide：

(1)K-means 聚类算：
※K-means 聚类算法思想：
k 均值聚类算法（k-means clustering algorithm）是一种迭代求解的聚类分析算法；
聚类是一个将数据集中在某些方面相似的数据成员进行分类组织的过程，聚类就是一种发现这种内在结构的技术，聚类技术经常被称为无监督学习。
k 均值聚类是最著名的划分聚类算法，由于简洁和效率使得他成为所有聚类算法中最广泛使用的。给定一个数据点集合和需要的聚类数目 k，k 由用户指定，k 均值算法根据某个距离函数反复把数据分入 k 个聚类中。
先随机选取 K 个对象作为初始的聚类中心。然后计算每个对象与各个种子聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心。聚类中心以及分配给它们的对象就代表一

个聚类。一旦全部对象都被分配了，每个聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复直到满足某个终止条件。

※K-means 聚类算法聚类过程：

Step01 用户确定簇个数 k。 *（计划将数据划分为 k 个帮派，还没分，是计划分 k 个帮派）*

Step02 随机确定 k 个初始点作为质心。 *（先在所有人中找 k 个人当老大）*

Step03 *对每个数据实例依次计算到 k 个质心的距离，选择最小距离的质心，并将其分配给该质心所对应的簇，直到数据集中的所有数据全都分配给 k 个簇。（没选上老大的人当小弟，小弟找离自己最近的老大当自己老大）*

Step04 更新 k 个簇的质心为该簇所有点的平均值。 *（已经确定的 k 个帮派内部选重新进行选举，确定一个不存在的点为精神寄托的虚空老大——也就是该帮派内部所有点 x 坐标加在一起求个均值，y 坐标加在一起求个均值，这个点在数学上就是质心。这个点是不在所有点里的单纯就是对于整个帮派来说"最中间的位置"）*

Step05 重新进行 step04 和 step04 操作并循环执行，直到 k 个质心位置不再发生变化。 *（选出新的老大后将这 k 个帮派解散，然后小弟重新找离自己最近的老大，一直循环直到各组成员稳定下来）*

※K-means 聚类算法优缺点：

优点：原理简单，实现容易；容易理解，聚类效果不错，虽然是局部最优， 但往往局部最优就够了；处理大数据集的时候，该算法可以保证较好的伸缩性；当簇近似高斯分布的时候，效果非常不错；算法复杂度低。

缺点：收敛较慢；算法时间复杂度比较高 O(nkt)；不能发现非凸形状的簇；需要事先确定超参数 K；对噪声和离群点敏感；结果不一定是全局最优，只能保证局部最优；K 值需要人为设定，不同 K 值得到的结果不一样；对初始的簇中心敏感，不同选取方式会得到不同结果；对异常值敏感；样本只能归为一类，不适合多分类任务；不适合太离散的分类、样本类别不平衡的分类、非凸形状的分类。

(2)K-medoide 聚类算法：

※算法步骤：

Step01 *确定聚类的个数 K。*

Step02 *在所有数据集合中选择 K 个点作为各个聚簇的中心点。*

Step03 *计算其余所有点到 K 个中心点的距离，并把每个点到 K 个中心点最短的聚簇作为自己所属的聚簇。*

Step04 *在每个聚簇中按照顺序依次选取点，计算该点到当前聚簇中所有点距离之和，最终距离之和最小的点，则视为新的中心点。*

Step05 *重复 step02step03 步骤，直到各个聚簇的中心点不再改变。*

*（和 K-means 差不多，**只是原本是求一个中心点 K 现在是找一个点在本族内正中间而已**）*

※算法优缺点：

优点：不容易受到那些由于误差之类的原因产生的脏数据的影响。

缺点：计算量显然要比 K-means 要大，一般只适合小数据量。

(3)聚类划分 k 值的选取：

k 值为分组的个数，计算方法很多（k 不唯一，可以动态改变，可以人为经验预设反之没个统一标准，选择最暴力的开方法反正都对）：假设我们有 m 个样本，该方法认为 K=（就是需要分类的东西的总数 m；**开根号就是 k 的值——即为分组个数**）

五，层次方法聚类：

※层次聚类算法思想：通过某种相似性测度计算节点之间的相似性，并按相似度由高到低排序，逐步重新连接个节点。

算法最初将每个对象作为一个簇，然后这些簇根据某些准则被一步步地合并两个簇间的相似；度由这两个不同簇中距离最近的数据点对的相似度来确定；聚类的合并过程反复进行直到所有的对象最终满足簇数目

※层次算法步骤：

序号	属性1	属性2
1	1	1
2	1	2
3	2	1
4	2	2
5	3	4
6	3	5
7	4	4
8	4	5

step01 *根据初始簇计算每个簇之间的距离，随机找出距离最小的两个簇进行合并，最小距离为 1，合并后 1,2 两个点合并为一个簇。*

Step02 *对上一次合并后的簇计算簇间距离，找出距离最近的两个簇进行合并，合并后 3,4 两个点成为一个簇，* ***（欧几里得距离）***

Step03 *重复第 2 步，5,6 点成为一个簇。*

Step04 *重复第 2 步，7,8 点成为一个簇。*

Step05 *合并{1,2}、{3,4}，使之成为一个包含 4 个点的簇。*

Step06 *合并{5,6}、{7,8}，由于合并后的簇的数目达到用户输入的终止条件，程序终止。*

过程表格如下：

步骤	最近的簇距离	最近的两个簇	合并后的新簇
1	1	{1}、{2}	{1、2}、{3}、{4}、{5}、{6}、{7}、{8}
2	1	{3}、{4}	{1、2}、{3、4}、{5}、{6}、{7}、{8}
3	1	{5}、{6}	{1、2}、{3、4}、{5、6}、{7}、{8}
4	1	{7}、{8}	{1、2}、{3、4}、{5、6}、{7、8}
5	1	{1、2}、{3、4}	{1、2、3、4}、{5、6}、{7、8}
6	1	{5、6}、{7、8}	{1、2、3、4}、{5、6、7、8}

※上一个案例第 5 步计算组合点集距离：

将数据点 B 与数据点 C 进行组合后，重新计算各类别数据点间的距离矩阵。数据点间的距离计算方式与之前的方法一样。这里需要说明的是组合数据点(B,C)与其他数据点间的计算方法。当我们计算(B,C)到 A 的距离时，需要**分别计算 B 到 A 和 C 到 A 的距离均值**。

$$D = \frac{\sqrt{(B-A)^2} + \sqrt{(C-A)^2}}{2} = \frac{21.6 + 22.6}{2}$$

计算组合数据点(A,F)到(B,C)的距离，这里**分别计算了(A,F)和(B,C)两两间距离的均值**。

$$D = \frac{\sqrt{(A-B)^2} + \sqrt{(A-C)^2} + \sqrt{(F-B)^2} + \sqrt{(F-C)^2}}{4}$$

※层次法优缺点：

优点：距离和规则的相似度容易定义，限制少；不需要预先制定聚类数；可以发现类的层次关系；

缺点：计算复杂度太高；奇异值也能产生很大影响；算法很可能聚类成链状；

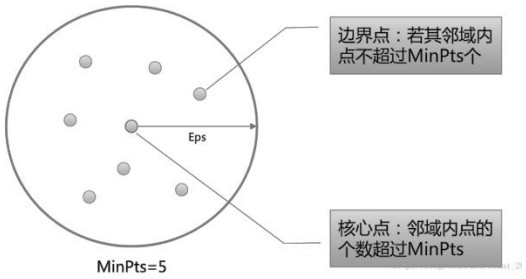
六，DBSCAN 聚类：

※算法思想：

由密度可达关系导出的最大密度相连的样本集合，即为我们最终聚类的一个类别，或者说一个簇。DBSCAN 是一种基于密度的聚类算法，这类密度聚类算法一般假定类别可以通过样本分布的紧密程度决定。同一类别的样本，他们之间的紧密相连的，也就是说，在该类别任意样本周围不远处一定有同类别的样本存在。

通过将紧密相连的样本划为一类，这样就得到了一个聚类类别。通过将所有各组紧密相连的样本划为各个不同的类别，则我们就得到了最终的所有聚类类别结果。

※算法步骤：



Step01：找两类点：<u>只需找出核心点和边界点即可</u>

<u>DBSCAN 算法将数据点分为三类：</u>

核心点：*在半斤 EPS 内含有超过 MinPts 数目的点*

边界点：*在半斤 EPS 内点的数量小于 MinPts，但是落在核心点的领域内*

（噪音点：既不是核心点也不是边界点的点）

Step02 ：*对所有点计算其领域 Eps=distance 内的点的集合*

Step03：*集合内的点个数超过 MinPt3=count 的点为核心点*

Step04：*查看剩余点是否在核心点的领域内，若在则为边界点，否则为噪声点*

Step05：*循环上面步骤*

※DBSCAN 优缺点：

优点：聚类速度快且能够有效处理噪声点和发现任意形状的空间聚类；与 K-MEANS 比较起来，不需要输入要划分的聚类个数；聚类簇的形状没有偏倚；可以在需要时输入过滤噪声的参数。

缺点：当数据量增大时，要求较大的内存支持 I/O 消耗也很大；当空间聚类的密度不均匀、聚类间距差相差很大时，聚类质量较差，因为这种情况下参数 MinPts 和 Eps 选取困难。算法聚类效果依赖与距离公式选取，实际应用中常用欧式距离，对于高维数据，存在“维数灾难”。

2021/6/20 海南大学 郑昊