第一部分：逻辑（logic）

1-1、概述

逻辑是人对于知识的定义和解释，主要分为两种

命题逻辑（propositional logic）：比较基础，适用于一些人工智能问题

一阶逻辑（first order logic）：更有表达能力，在人工智能领域使用较为广泛

1-2、命题逻辑（propositional logic）

1-2-1、组成

逻辑常量：真值，假值

命题符号：表示原子命题，语义由使用者定义

括号

连接符：析取（∧），合取（∨），蕴含（⇒），等值（⇔），非（¬）

1-2-2、真值表（truth table）

真值表用来说明命题中的原子命题为真值或假值的情况下命题的真假

当命题中的原子命题的真值确定时命题的真值也随之确定

1-2-3、模型与知识库（model and knowledge base）

知识库：多个命题的集合

模型：使得知识库中全部命题都为真时的原子命题真值组合

1-2-4、恒真命题，矛盾与导出（tautology, contradiction and entail）

恒真命题：不论命题中的原子命题的真值，命题都为真

矛盾：不论命题中的原子命题的真值，命题都为假

P导出Q（记为P |= Q）：当命题P为真时，命题Q也为真

此时命题P的模型也是命题Q的模型

1-2-5、优缺点

优点：命题的真假与上下文无关

缺点：表达能力有限

1-3、一阶谓词逻辑（first-order predicate logic）

1-3-1、组成

对象：表示一个具体对象

谓词符号：表示对象的属性

量词：全称量词（∀），存在量词（∃）

1-3-2、谓词逻辑的作用域

1、如果量词后面有括号就作用在括号范围内

$$\forall x(F(x) \iff F(h))$$

2、如果量词后面没有括号就负责到连接符之前

$$\forall x F(x) \iff F(h)$$

3、如果量词后面还是量词就作用到后面那个量词的作用域

$$\forall x \exists y R(x, y)$$

1-3-3、命题证明

一般通过转换成对应比较好证明的等价命题进行证明

第二部分：数学基础（foundation of mathematics）

2-1、概率论基础（probability）

乘法定律：

$$P(A,\ B) = P(B \mid A)P(A) = P(A \mid B)P(B)$$

加法定律：

$$P(A) = \sum_{i=1}^{n} P(A,\ B_i) = \sum_{i=1}^{n} P(A \mid B_i)P(B_i)$$

2-2、期望，中位数，众数与方差（expectation, median, mode and variance）

2-2-1、期望

离散型变量的期望：

$$E[X] = \sum_i x_i P\{X = x_i\}$$

连续型变量的期望：

$$E[X] = \int_{-\infty}^{\infty} x f(x)\,dx$$

期望的性质：

$$E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i]$$

2-2-2、几何平均数（geometric mean）

$$G_m = \sqrt[n]{\prod_{i=1}^{n} X_i}$$

2-2-3、中位数

当X(1..n)为有序时，在n为奇数的情况下：

$$median = X((n+1)/2)$$

在n为偶数的情况下：

$$median = (X(n/2)+X(n/2+1))/2$$

2-2-4、众数

2-2-5、方差

$$Var(X) = E[(X-E(X))^2] = E[X^2]-(E[X])^2$$

2-2-6、协方差（covariance），协方差矩阵与相关系数（correlation coefficient）

$$Cov(X,\ Y) = E[(X-\mu_x)(Y-\mu_y)] = E[XY] - E[X]E[Y]$$

$$var(X+Y) = var(X) + var(Y) + 2cov(X,Y)$$

假设多变量 $C = (C_1, C_2, \dots, C_n)$ 下的m组数据

$$S1 = (S1_{C_1}, S1_{C_2}, \dots, S1_{C_n})$$
$$S2 = (S2_{C_1}, S2_{C_2}, \dots, S2_{C_n})$$
$$\dots$$
$$Sm = (Sm_{C_1}, Sm_{C_2}, \dots, Sm_{C_n})$$

协方差矩阵为：

$$cov(Cj,Ck) = \frac{1}{m-1}\left[\sum_{i=1}^{m}(Si_{Cj} - E(C_j))(Si_{Ck} - E(C_k))\right]$$

$$cov\ matrix = \frac{1}{m-1}\begin{bmatrix} cov(C1,C1) & cov(C1,C2) & \dots & cov(C1,Cn) \\ cov(C2,C1) & cov(C2,C2) & \dots & cov(C2,Cn) \\ \dots & \dots & \dots & \dots \\ cov(Cn,C1) & cov(Cn,C2) & \dots & cov(Cn,Cn) \end{bmatrix}$$

注：对于样本数据集来说要除的是m - 1，而不是m

变量Cj和Ck的相关系数为：

$$corr(Cj,Ck) = \frac{cov(Cj,Ck)}{\sqrt{var(Cj)var(Ck)}}$$

注：相关系数只对于两个变量而言

例题一：假设数据样本

$$(2, 19), (9, 6), (7, 15), (5, 12)$$

其协方差矩阵为

$$\bar{x} = \frac{2+9+7+5}{4} = 5.75, \quad \bar{y} = \frac{19+6+15+12}{4} = 13$$

$$\sigma_x^2 = \frac{1}{3}[(2-5.75)^2 + (9-5.75)^2 + (7-5.75)^2 + (5-5.75)^2] = 8.917$$

$$\sigma_y^2 = \frac{1}{3}[(19-13)^2 + (6-13)^2 + (15-13)^2 + (12-13)^2] = 30$$

$$\text{cov}(x, y) = \frac{1}{3}[(2-5.75)(19-13) + (9-5.75)(6-13) +$$
$$(7-5.75)(15-13) + (5-5.75)(12-13)] = -14$$

The covariance matrix is given by $\begin{bmatrix} 8.917 & -14 \\ -14 & 30 \end{bmatrix}$

相关系数为

$$r_{xy} = \frac{-14}{\sqrt{8.917}\sqrt{30}} = -0.86$$

2-3、数据质量及数据类型

2-3-1、数据质量

精度：数据的聚合程度，一般通过标准差进行衡量
偏移量：通过与均值进行比较

2-3-2、随机变量类型

离散型随机变量：

伯努利随机变量

$$P\{X = i\} = \frac{n!}{i!(n-i)!}p^i(1-p)^{n-i} \quad i = 0,1,...,n$$

$$E[X] = np$$

$$var(X) = np(1-p)$$

泊松随机变量

$$P\{X = i\} = e^{-\lambda}\frac{\lambda^i}{i!} \quad i = 0,1,...,n$$

$$E[X] = \lambda$$

$$var(X) = \lambda$$

$$\lambda = \frac{\sum_{i=1}^{n} i}{n}$$

连续型随机变量：

均匀分布随机变量

$$f(x) = \begin{cases} \frac{1}{b-a} & a < x < b \\ 0 & otherwise \end{cases}$$

$$E(X) = \frac{a+b}{2}$$

$$var(X) = \frac{1}{12}(b-a)^2$$

正态分布随机变量

$$f(x) = -\frac{1}{\sqrt{2\pi}\,\sigma}e^{-(x-\mu)^2/2\sigma^2}$$

$$E(X) = \mu$$

$$var(X) = \sigma^2$$

2-3-3、数据类型

标定数据(nominal data)：仅具有名称上的含义，没有数学上的含义
有序数据(ordinal data)：具有顺序含义的数据
区间数据(interval data)：某个区间段的数据
比例数据(ratio data)：连续型数据

2-4、距离

Minkowski距离

$$d(x,y) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{\frac{1}{r}}$$

city-block距离

$$d(x,y) = \sum_{k=1}^{n} |x_k - y_k|$$

Euclidean距离

$$d(x,y) = \sqrt{\sum_{k=1}^{n} |x_k - y_k|^2}$$

Supermum距离

$$d(x,y) = \sqrt[\infty]{\sum_{k=1}^{n} |x_k - y_k|^{\infty}}$$

2-5、熵（entropy）

信息熵（单位：bit）

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

联合熵与条件熵（单位：bit）

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(x,y)$$

$$H(Y|X) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(y|x)$$

第三部分：分类（classification）

3-1、概述

传统分类主要分成两步：

1、训练：将已知类别标签的数据作为训练集对模型（分类器）进行训练

2、测试：将未知标签的数据作为测试集测试模型（分类器）的准确度

衡量指标：准确性，速度，鲁棒性，可扩展性，可解释性

3-2、决策树(decision tree)

3-2-1、概述

特征：树状结构，内部节点表示基于某个属性的分裂，分支表示不同的属性，叶节点表示最终的数据分布

训练过程：初始化所有数据分布在一个节点（根节点）上，然后这个根节点通过选择属性进行分裂。直到没有多余的可以分裂的属性或每个叶节点的样本都是一个类别的。

测试过程：去掉不合理的可以导致噪声的分支

使用算法：通过自顶而下递归分治的做法将数据样本进行划分

3-2-2、ID3

概述：通过衡量属性信息增益进行分裂属性的选择（选择信息增益大的）

关键：信息增益等于划分前数据集的熵减去划分后数据集的条件熵

$$information\ \ gain(D,A) \ = \ H(D) \ - \ H(D|A)$$

其中D为划分前的数据样本，A为本次划分的属性

其中假设属性A将原来的样本D划分成了n个样本

$$H(D|A) \ = \ - \sum_{x \in A} \sum_{y \in Y} p(x)p(y|x) \log_2 p(y|x)$$

其中Y表示Y个不同的类别，A表示A个划分的属性

问题：只衡量增益不衡量分类后的分支数，很容易出现无意义的属性标签划分（比如说：索引）

例题二、假设有如下的数据样本

Instance	a_1	a_2	a_3	Target Class
1	T	T	1	+
2	T	T	6	+
3	T	F	5	-
4	F	F	4	+
5	F	T	7	-
6	F	T	3	-
7	F	F	8	-
8	T	F	7	+
9	F	T	5	-

其原始的信息熵为：

$$-\frac{4}{9}\log_2\frac{4}{9}-\frac{5}{9}\log_2\frac{5}{9}=0.991 \quad \text{bit}$$

如果通过a1属性进行划分，对应的信息熵为：

$$\frac{4}{9}(-\frac{3}{4}\log_2\frac{3}{4}-\frac{1}{4}\log_2\frac{1}{4})+\frac{5}{9}(-\frac{1}{5}\log_2\frac{1}{5}-\frac{4}{5}\log_2\frac{4}{5})=0.762 \quad \text{bit}$$

信息增益为：

$$0.991-0.762=0.229 \text{ bit}.$$

如果通过a2属性进行划分，对应的信息熵为：

$$\frac{5}{9}(-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5})+\frac{4}{9}(-\frac{2}{4}\log_2\frac{2}{4}-\frac{2}{4}\log_2\frac{2}{4})=0.984 \quad \text{bit}$$

信息增益为：

$$0.991-0.984=0.007 \text{ bit}.$$

原始的划分误差为：

$$1-\max(\frac{4}{9},\frac{5}{9})=\frac{4}{9}$$

根据a1进行划分后的划分误差为：

$$\frac{4}{9}[1-\max(\frac{3}{4},\frac{1}{4})]+\frac{5}{9}[1-\max(\frac{1}{5},\frac{4}{5})]=\frac{2}{9}$$

误差减少为：

$$\triangle \mathrm{E}(a_1)=4/9-2/9=2/9$$

根据a2进行划分后的划分误差为：

$$\frac{5}{9}[1-\max(\frac{2}{5},\frac{3}{5})]+\frac{4}{9}[1-\max(\frac{2}{4},\frac{2}{4})]=\frac{4}{9}$$

误差减少为：

$$\triangle \mathrm{E}(a_2)=4/9-4/9=0.$$

如果通过a3属性进行划分，根据信息增益求最好的划分点

a_3	Class label	Split point	Entropy	Info gain
1	+	2.0	0.848	0.143
3	-	3.5	0.989	0.002
4	+	4.5	0.918	0.073
5 5	- -	5.5	0.984	0.007
6	+	6.5	0.973	0.018
7 7	+ -	7.5	0.889	0.102
8	-			

3-2-3、C4.5

概述：通过衡量属性信息增益率进行分裂属性的选择（选择信息增益率大的）

关键：信息增益率的计算

$$information\ \ gain\ \ ratio(D,A) = \frac{information\ \ gain(D,A)}{splitinfo(D,A)}$$

$$= \frac{H(D) - H(D|A)}{-\sum_{j=1}^{v} \frac{|D_j|}{|D|} \log_2\left(\frac{|D_j|}{|D|}\right)}$$

3-2-4、CART

概述：通过衡量gini划分系数进行分裂属性的选择（选择gini划分系数小的）

关键：gini划分系数的计算

$$gini\ \ split(D,A) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} gini(D_j)$$

$$gini(D_j) = 1 - \sum_{i=1}^{n} p(D_{j_i}|D_j)^2$$

例题三、假设有如下的数据样本

Instance	a_1	a_2	a_3	Target Class
1	T	T	1	+
2	T	T	6	+
3	T	F	5	-
4	F	F	4	+
5	F	T	7	-
6	F	T	3	-
7	F	F	8	-
8	T	F	7	+
9	F	T	5	-

其原始的gini系数为：

$$1-(\frac{4}{9})^2-(\frac{5}{9})^2=0.494$$

如果通过a1属性进行划分，对应的gini系数为：

$$\frac{4}{9}[1-(\frac{3}{4})^2-(\frac{1}{4})^2]+\frac{5}{9}[1-(\frac{1}{5})^2-(\frac{4}{5})^2]=0.344$$

gini系数的变化量为：

$$0.494 - 0.344 = 0.15.$$

如果通过a2属性进行划分，对应的gini系数为：

$$\frac{5}{9}[1-(\frac{2}{5})^2-(\frac{3}{5})^2]+\frac{4}{9}[1-(\frac{2}{4})^2-(\frac{2}{4})^2]=0.489$$

gini系数的变化量为：

$$0.494 - 0.489 = 0.005.$$

3-2-5、过拟合（over-fitting）

过拟合有两种情况：

1、数据样本太少：出现训练误差极小但测试误差极大地情况

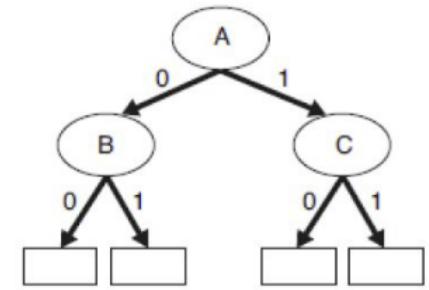
解决办法：增大训练集

2、数据样本太多：随着树的增大，训练误差会不断减小，但模型复杂度随之增大并且拟合了很多的噪声数据导致测试误差大

解决办法：在训练误差计算时引入惩罚项

$$e = \frac{number\ of\ miss\text{-}classified\ samples + number\ of\ branches \times penalty\ term}{number\ of\ samples}$$

例题四、给定下面这棵决策树



Training:

Instance	A	B	C	Class
1	0	0	0	+
2	0	0	1	+
3	0	1	0	+
4	0	1	1	–
5	1	0	0	+
6	1	0	0	+
7	1	1	0	–
8	1	0	1	+
9	1	1	0	–
10	1	1	0	–

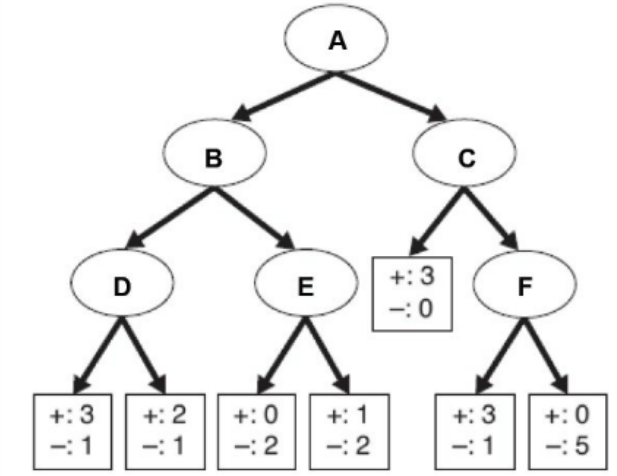
在不引入惩罚项的情况下的误差为

$$\frac{0+1+2+0}{10}=\frac{3}{10}$$

在对每个叶节点引入惩罚项0.5的情况下的误差为

$$\frac{0+1+2+0+0.5\times4}{10}=\frac{5}{10}$$

例题五、给定下面这棵决策树并假设对每个叶节点引入惩罚项1.5



如果在F节点处进行叶节点合并误差为

$$\frac{6+1.5\times 6}{24}=\frac{15}{24}$$

如果在D、E节点处同时进行叶节点合并误差为

$$\frac{4+1.5\times 5}{24}=\frac{11.5}{24}$$

如果在D、E、F节点处同时进行叶节点合并误差为

$$\frac{6+1.5\times 4}{24}=\frac{12}{24}$$

3-2-6、分析

特点：对定性离散型变量进行划分（对于连续型变量要划分区间段）
　　单属性决策，不同属性的决策顺序影响最终结果

优势：便于理解

劣势：存在过拟合的问题

3-3、PLA

3-2-1、概述

通过划线的方法对连续性定量变量进行划分，可以进行多个属性下的划分

3-2-2、方法

假设有d个属性，对于一个数据样本x来说通过下式计算h(x)的值

$$h(x) = sign\left(\sum_{i=1}^{d} w_i d_i - threshold\right)$$

h(x)为1的为一类，其他的为另一类

其中最核心的问题是w的值怎么选定

常见做法是先随机初始化w的值，然后看一下有没有错分的点对（也就是居于直线同侧的本属于不同类别的点，然后进行下列操作）

假设 (x, y) 为错分的点对，则通过如下方式更新w的值

$$when\ \ sign(\boldsymbol{w}_t^T\boldsymbol{x})\ =\ 1:\ \boldsymbol{w}_{t+1}\ =\ \boldsymbol{w}_t\ -\ \boldsymbol{x}$$
$$when\ \ sign(\boldsymbol{w}_t^T\boldsymbol{x})\ =\ -1:\ \boldsymbol{w}_{t+1}\ =\ \boldsymbol{w}_t\ +\ \boldsymbol{x}$$

3-2-3、分析
优势：可以对于连续型变量进行划分，并可以同时考虑多个属性
劣势：不一定收敛，无法应对非线性可分的情况，解不唯一（随机初始化权重w；错分时选点的顺序不同）

3-4、朴素贝叶斯(naive Bayes)

3-4-1、概述
一种基于统计学的数据预测模型，假设属性与属性之间相互独立
3-4-2、方法
假设有n个属性的数据样本X = (x1, x2, …, xn)，并有m个类别(C1, C2, …, Cm)，通过贝叶斯公式计算先验概率

$$p(C_i|X)\ =\ \frac{p(X|C_i)p(C_i)}{p(X)}$$

其中p(X)为固定值，因此只需要求分子取得最大值时对应的类别
由于n个属性相互独立，因此

$$p(X|C_i)\ =\ \prod_{k=1}^{n} p(X_k|C_i)$$

注：有些情况下对于稀疏数据样本一般通过平滑或者词带的方法进行解决
3-4-3、评价
优势：易于实现，一般能得到比较好的结果
劣势：要求属性相互独立忽略了属性的关联性

例题六、假设存在下面的文本

ID	Input review text	Class label
1	Good, thanks	Positive
2	No impressive, thanks	Negative
3	Impressive good	Positive

用朴素贝叶斯预测No, thanks的类标签
先将文本进行离散化：

ID	good	thanks	no	impressive	Class label
1	1	1	0	0	Positive
2	0	1	1	1	Negative
3	1	0	0	1	Positive
4	0	1	1	0	?

然后根据贝叶斯公式分别求出为positive或者为negative的后验概率
1、positive
表达式为：

$$P(\text{Class label}=\text{“Positive”}|\text{ID}=4)$$

$$= P(\text{Class label}=\text{“Positive”})P(\text{ID}=4|\text{Class label}=\text{“Positive”}) / P(\text{ID}=4)$$
$$= P(\text{Class label}=\text{“Positive”})P(\text{“thanks”, “no”}|\text{Class label}=\text{“Positive”}) / P(\text{ID}=4)$$

由于在贝叶斯公式中属性间相互独立，因此有：

$$P(\text{“thanks”, “no”}|\text{Class label}=\text{“Positive”})$$
$$= P(\text{“thanks”}|\text{Class label}=\text{“Positive”})P(\text{“no”}|\text{Class label}=\text{“Positive”})$$

由题意我们可知

$$P(\text{Class label}=\text{“Positive”}) = 2/3,$$
$$P(\text{“thanks”}|\text{Class label}=\text{“Positive”}) = 1/4,$$
$$P(\text{“no”}|\text{Class label}=\text{“Positive”}) = 0.$$

因此可求得

$$P(\text{Class label}=\text{“Positive”}|\text{ID}=4) = (2/3) * (1/4) * 0 / P(\text{ID}=4) = 0 / P(\text{ID}=4)$$

2、negative

$$P(\text{Class label}=\text{“Negative”}|\text{ID}=4)$$
$$= P(\text{Class label}=\text{“Negative”})P(\text{ID}=4|\text{Class label}=\text{“Negative”}) / P(\text{ID}=4)$$
$$= P(\text{Class label}=\text{“Negative”})P(\text{“thanks”, “no”}|\text{Class label}=\text{“Negative”}) / P(\text{ID}=4)$$
$$= (1/3) * (1/3) * (1/3) / P(\text{ID}=4) = (1/27) / P(\text{ID}=4)$$

通过比较上面两式可以得知应该预测为negative

3-5、k近邻(k-nearest neighbor)

3-5-1、概述

通过查看最近的k个点所属的类别判断样本点所属的类别

3-5-2、关键问题

用什么方法进行点与点之间距离的测算
k值应该取多少
邻居的权重是否有所不同

3-5-3、评价

优势：实现简单
劣势：速度慢

例题七、假设存在下面的文本

ID	Input review text	Class label
1	Good, thanks	Positive
2	No impressive, thanks	Negative
3	Impressive good	Positive

用KNN（k设为1）预测No, thanks的类标签

$$d(\text{ID=4},\text{ID=1})=\sqrt{(0-1)^2+(1-1)^2+(1-0)^2+(0-0)^2}=\sqrt{2}$$

$$d(\text{ID=4},\text{ID=2})=\sqrt{(0-0)^2+(1-1)^2+(1-1)^2+(0-1)^2}=1$$

$$d(\text{ID=4},\text{ID=3})=\sqrt{(0-1)^2+(1-0)^2+(1-0)^2+(0-1)^2}=2$$

从此可以看出，离最近的k（k为1）个样本中，negative的多于positive的
因此应该预测为negative

3-6、回归（仅作了解）

通过曲线的办法尽量拟合尽可能多的数据样本点
使用最小二乘法
容易受到噪声点的影响

3-7、神经网络（仅作了解）

3-7-1、概述

通过模拟生物学上神经元学习的模式，使用多层网络结构的分类器

3-7-2、方法

先通过前向迭代的方法从输入层经过逻辑变换得到输出层
将得到的结果与真实结果进行比较将误差用梯度下降法反向迭代至输入层，调整变换函数
迭代直至训练完毕（收敛）

3-7-3、评价

优势：鲁棒性强，抗噪，非常适合于于连续型变量，能解决非线性可分的问题（通过增加决策层数或进行逻辑变换）
劣势：训练时间长，需要初始化很多的变量，解释性不强

3-8、支持向量机（仅作了解）

通过支持向量函数进行分类，可以适应非线性和高维的定量变量，解决了PLA中的划线不唯一的问题

3-9、分类器的评价指标

准确率（precision）：分类器认为属于某一类的样本里面实际属于这一类样本的比重
召回率（recall）：实际属于某一类的样本里面分类器认为属于这一类样本的比重
F值（F-Measure）：对于准确率和召回率的综合评价

$$F=\frac{2\times precision\times recall}{precision+recall}$$

第四部分：关联规则分析（association analysis）

4-1、概述

通过数据的关联性进行分析挖掘，适合用于大数据样本中
项集：包含0个或多个项的集合。包含k个项就是一个k项集
事务：表示事件
举例：每次去商场购买东西是一次事务而实际购买到的东西就是项集

4-1-1、支持度与置信度（support and confidence）

支持度：说明给定数据集的频繁程度
置信度：说明推理的可靠程度

$$support(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{N}$$

$$confidence(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{\sigma(X)}$$

在小型数据集下会先设定支持度和置信度的阈值然后再根据阈值进行挖掘
但在大数据集下这样做代价太高（每种情况都计算一遍还要存储非常麻烦）
因此一般情况下会分成两步：频繁项集生成和规则生成
　　频繁项集生成(frequent item set generation)：找到支持度高的项集
　　规则生成(rule generation)：找到置信度高的项集

4-2、频繁项集生成(frequent item set generation)：Apriori算法

4-2-1、原理

　　如果一个项集是频繁的，则它的所有子集也是频繁的
　　也就是说如果一个项集不频繁，那么它的超集也一定不频繁，因此可以从子集往上生成
　　因此我们就可以进行候选项集的产生与剪枝操作

4-2-2、方法

方法1：暴力法
　　枚举所有k-项集的情况，计算支持度，淘汰支持度小于阈值的
方法2：Fk-1和F1方法
　　1、扫描所有1-项集的支持度，去除支持度小于阈值的项集
　　2、迭代：通过扫描k-项集的支持度，选取支持度大于阈值的k-项集和1-项集生成k+1-项集
　　3、直到所有项都覆盖到
方法3：Fk-1和Fk-1方法
　　通过扫描k-项集的支持度，选取支持度大于阈值的k-项集和与这个k-项集只有一个项的差别的k-项集生成k+1-项集

4-2-3、支持度计数

　　支持度数：在事务中某个项集出现的次数
　　方法：以树状结构的方法进行表示，从顶层开始往下层，每往下一层的项集对应的支持度数加1

4-3、最大频繁项集与闭频繁项集(maximal frequent item set and closed item set)

4-3-1、最大频繁项集

定义：最大频繁项集的所有直接超集都是不频繁的
性质：最大频繁项集和它的所有子集都是频繁项集
问题：缺失支持度信息

4-3-2、闭频繁项集

闭项集定义：闭频繁项集具有和它的所有直接超集不同的支持度数
闭频繁项集定义：在闭项集的基础上支持度达到支持度阈值的项集
性质：非闭频繁项集的支持度必定是它的所有直接超集中支持度的最大值
最大频繁项集一定是闭频繁项集

4-4、关联模式的评估

4-4-1、支持度和置信度的局限性

支持度：许多潜在的有意义的模式由于包含支持度小的项而删去
置信度：忽略了规则后键（箭头后面那个项）的支持度，忽略了逆关系

4-4-2、提升度(lift)

背景：置信度忽略了规则后键的支持度
定义：

$$lift(X \rightarrow Y) = \frac{confidence(X \rightarrow Y)}{support(Y)}$$

4-4-3、兴趣因子(interest factor)

对于二元变量来说衡量提升度相当于衡量兴趣因子，其定义为：

$$I(A, B) = \frac{support(A, B)}{support(A) \times support(B)}$$

并具有下列的性质：

$$I(A, B)\begin{cases} = 1 & independence \\ > 1 & positive\ correlation \\ < 1 & negative\ correlation \end{cases}$$

第五部分：聚类（cluster）

5-1、概述

5-1-1、概述

聚类是将没有标签的数据聚在不同的类别里面，在同一类别里面的数据要尽量相似，类与类之间的数据要尽量相异

5-1-2、聚类类型

1、层次的与划分的（hierarchical and partitional）

层次聚类：形成树状结构，类似于分治法进行分层划分

划分聚类：简单地将对象扔在不同的类别里面

2、互斥的、重叠的与模糊的（exclusive, overlapping and fuzzy）

互斥聚类：每个对象只属于一个类

重叠聚类：每个对象可以属于多个类

模糊聚类：一部分属于一个类，又有一部分属于其他类

3、完全的与部分的（complete and partial）

完全聚类：所有对象都要被聚类

部分聚类：不要求所有对象都要被聚类

5-2、K均值聚类（K-means）

5-2-1、概述

最传统的聚类算法，通过初始化指定K个样本中心点进行聚类

5-2-2、算法

1、初始化选择K个点作为K个聚类样本中心

2、迭代：

计算每个样本点到中心点的距离，根据距离判断每个样本点所属类别

各聚类根据属于自己这一类的样本点的均值更新聚类中心

3、直到每个样本点所属类别不再变化

5-2-3、算法分析

在以欧氏距离作为邻近性度量时，使用误差的平方和(sum of the squared error, SSE)，而K均值算法在这种情况下可以得到全局最优的聚类

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} d(x, C_i)^2$$

离群点（outlier）问题：离群点将极大影响聚类中心的位置，通过寻找样本数很少的聚类在一定程度上找到离群点

算法后处理：在算法使用过后还可以对聚类进行进一步的分裂、合并或直接拆散或引进新的聚类中心等方法以达到更好的效果

5-2-4、二分K均值聚类（仅作了解）

通过分裂聚类的方法得到K个聚类

5-2-5、算法局限性

初始化对结果的影响大

对非球形样本的聚类效果不好

例题八、假设有下面六个样本点

A: (1, 3), B: (2, 1), C: (2, 2), D: (3, 5), E: (4, 4), F: (3, 3)

1、初始化将B、D、E三点设为三个聚类的中心点，求经过一次迭代后每个点属于的类别和第一次迭代后的聚类中心点

The first cluster is {A, B, C}, and its centroid is (5/3, 2).

The second cluster is {D}, and its centroid is (3, 5).

The third cluster is {E, F}, and its centroid is (3.5, 3.5).

2、假设要聚成两个类，并初始化认为ADE是一个类的，BCF是一个类的，求最终的聚类情况

b) Initially, the first cluster "C1" is {A, D, E}, and its centroid is (8/3, 4).

The second cluster "C2" is {B, C, F}, and its centroid is (7/3, 2).

After the first iteration, the first cluster "C1" is {D, E, F}, and its centroid is (4, 4).

The second cluster "C2" is {A, B, C}, and its centroid is (5/3, 2).

Then, the k-Means algorithm is convergence.

5-3、层次聚类（hierarchical clustering）

5-3-1、概述

通过形成树状结构，进行分层划分式的聚类

分类：

凝聚（agglomerative）型层次聚类：从底层往上层进行聚类

分裂（divisive）型层次聚类：从上层往下层开始分裂

5-3-2、算法：

1、初始化距离矩阵

2、迭代

合并距离最近的两个类

更新距离矩阵

3、直到聚合成一个类

5-3-3、类与类之间的距离测算

1、单链接（single link）：取两个类中任意两个样本点的距离的最小值

适合非椭圆形样本

对噪声点和离群点敏感

2、全链接（complete link）：取两个类任意两个样本点的距离的最大值

结果偏向于球状

对噪声点和离群点不敏感

3、组平均（group average）：取两个类任意两个样本点的距离的平均值

5-3-4、算法评价

这个算法很便于我们得到样本的层次结构

局限性在于时间空间复杂性大，而且传统方法下一旦被聚类就无法还原

例题九、假设有下面四个样本点并以欧氏距离作为距离测量标准

$$\text{A: (2, 2), B: (2, 3), C: (3, 5), D: (4, 3)}$$

1、在单链接的情况下各样本点的合并顺序

$$d(A,B)=1,\ d(A,C)=\sqrt{10},\ d(A,D)=\sqrt{5},$$

$$d(B,C)=\sqrt{5},\ d(B,D)=2,\ d(C,D)=\sqrt{5}.$$

A and B are merged firstly. We denote the cluster containing A and B by C1.

Then, for C, D, C1:

$$d(C,C1)=\sqrt{5},\ d(D,C1)=2,\ d(C,D)=\sqrt{5}.$$

Thus, D and C1 (A and B) are then merged, which is denoted by C2.

Finally, C and C2 (A, B and D) are merged.

2、在全连接的情况下各样本点的合并顺序

$$d(A,B)=1,\ d(A,C)=\sqrt{10},\ d(A,D)=\sqrt{5},$$

$$d(B,C)=\sqrt{5},\ d(B,D)=2,\ d(C,D)=\sqrt{5}.$$

A and B are merged firstly. We denote the cluster containing A and B by C1.

Then, for C, D, C1:

$$d(C,C1)=\sqrt{10},\ d(D,C1)=\sqrt{5},\ d(C,D)=\sqrt{5}.$$

Thus, we have the following two orders:

(a) D and C1 (A and B) are then merged, which is denoted by C2.

Finally, C and C2 (A, B and D) are merged.

(b) C and D are then merged, which is denoted by C2.

Finally, C1 (A and B) and C2 (C and D) are merged.

5-4、DBSCAN

5-4-1、概述

通过衡量样本的密度进行聚类

5-4-2、核心点，边界点和噪声点（core point, border point and noise point）

核心点：在给定半径下以自身为圆心作圆，圆形包含的点数超过阈值的就是核心点
边界点：在给定半径下以自身为圆心作圆，圆形包含的点数没有超过阈值的但是包含了核心点的为边界点
噪声点：在给定半径下以自身为圆心作圆，圆形包含的点数没有超过阈值且没有包含核心点的为噪声点

5-4-3、算法

1、找到所有核心点
2、找到所有边界点和噪声点
3、将距离在给定半径下的核心点中间连一条线
4、每组连通的核心点形成一个聚类
5、将每个边界点分配到对应的聚类中

5-4-4、算法评价
对噪声点和离群点不敏感，可以适应多种形状的聚类

例题十、假设有下面六个样本点并给定半径为2，点数为3

p1: (5, 9), p2: (5, 8), p3: (3, 8), p4: (1, 2), p5: (2, 1), p6: (4, 4)

以上六个点中哪些是核心点，哪些是边界点，哪些是噪声点

The neighborhood of each point is as follows:

N(p1)={p1, p2}, N(p2)={p1, p2, p3}, N(p3)={p2, p3},

N(p4)={p4, p5}, N(p5)={p4, p5}, N(p6)={p6}. Thus,

The core point is p2.

The border points are p1, p3.

The noise points are p4, p5, p6.