

# 数据挖掘复习题（201807）

前言：

## 1. 数据挖掘的特点 ★★★

- (1) **数据挖掘的数据必须是真实的。**数据挖掘所处理的数据通常是已经存在的真实数据，而不是为了进行数据分析而专门收集的数据。因此，数据收集本身不属于数据挖掘所关注的焦点，这是数据挖掘区别于大多数统计任务的特征之一。
- (2) **数据挖掘所处理的数据必须是海量的。**如果数据集很小的话，采用单纯的统计分析方法就可以了。但是，当数据集很大时，会面临许多新的问题，诸如数据的有效存储、快速访问、合理表示等。
- (3) **查询一般是决策制定者（用户）提出的随机查询。**查询要求灵活，往往不能形成精确的查询要求，要靠数据挖掘技术来寻找可能的查询结果。
- (4) **挖掘出来的知识一般是不能预知的，数据挖掘发现的是潜在的、新颖的知识。**这些知识在特定环境下是可以接受、可以理解、可以运用的，但不是放之四海皆准的。（书 110-7.1）

## 2. 数据挖掘的分类

- 1) 根据挖掘的数据库类型分类，2) 根据挖掘的知识类型分类，3) 根据所用的技术分类，4) 根据挖掘的应用领域分类

## 3. 数据挖掘的组件化思想（五大组件）★★★

数据挖掘算法都是由 5 个“标准组件”构成的，即模型或模式结构、数据挖掘任务、评分函数、搜索和优化方法、数据管理策略。

每一种组件都蕴含着一些非常通用的系统原理，掌握了每一种组件的基本原理之后，再来理解由不同组件“装配”起来的算法就变得相对轻松一些。而且，不同算法之间的比较也变得更加容易，因为能从组件这个层面看出算法之间的异同。

- 1)通过数据挖掘过程所得到的知识通常被称为模型（model）或模式(pattern)。模型是全局的，模式是局部的。
  - 2)根据数据分析者的目标，可以将数据挖掘任务分为：模式挖掘，模型挖掘(描述建模,预测建模)
  - 3)评分函数用来对数据集与模型（模式）的拟合程度进行评估。常用的评分函数有：似然（likelihood）函数、误差平方和、准确率等。
  - 4)搜索和优化的目标是确定模型（模式）的结构及其参数值，以使评分函数达到最小值（或最大值）。
  - 5)数据管理策略应该设计有效的数据组织和索引技术，或者通过采样、近似等手段，来减少数据的扫描次数，从而提高数据挖掘算法的效率。
- 确定模型(模式)结构和评分函数的过程通常由人来完成,而优化评分函数的过程通常需要计算机辅助来实现。(书 P116 7.2)

## 4. 三个经典数据挖掘算法的组件（书 P120）

	Apriori	ID3	K-means
任务	规则模式发现	分类	聚类
模型（模式）	关联规则	决策树	聚类
评分函数	支持度、置信度	分类准确度、信息增益	误差平方和
搜索方法	宽度优先搜索（带剪枝）	贪婪搜索	梯度下降
数据管理策略	未指定	未指定	未指定
提出年代	Agarawal 1994 年	Quin lan 20 世纪 80 年代	Mac Queen1967 年

## 一. 关联规则 Association: 频繁模式 (FP, Frequent Pattern)

[定义]Frequent pattern: a pattern (a set of items, subsequences, substructured, etc.)that occurs frequently in a data set.

关联规则: Apriori[AgSr94](BFS), FP-Growth(DFS); Sequential Pattern 序列模式: GSP,PrefixSpan; Sub-Graph 频繁子图: AGM,FSG;

### 1.1 什么是关联规则的支持度? 什么是关联规则的可信度? (书 P123) ★★★

(1) 给定关联规则  $X \Rightarrow Y$ , 支持度指项集  $X$  和  $Y$  在数据库  $D$  中同时出现的概率, 即  $\Pr(XUY)$ 。

(2) 给定关联规则  $X \Rightarrow Y$ , 可信度指项集  $X$  出现的情况下, 项集  $Y$  在数据库  $D$  中同时出现的条件概率, 即  $\Pr(X|Y) = \Pr(XUY) / \Pr(X)$ 。

### 1.2 Apriori 算法【经典】(考大题 20 分) ★★★★★


重要公理: 如果一个项目集  $S$  是频繁的 (项目集  $S$  的出现频度大于最小支持度  $s$ ), 那么  $S$  的任意子集也是频繁的。

例题 1: 现有如下事物数据库, 设  $\min\_sup=40\%$ ,  $\min\_conf=80\%$

(1) 请用 Apriori 算法找出所有的频繁项目集;

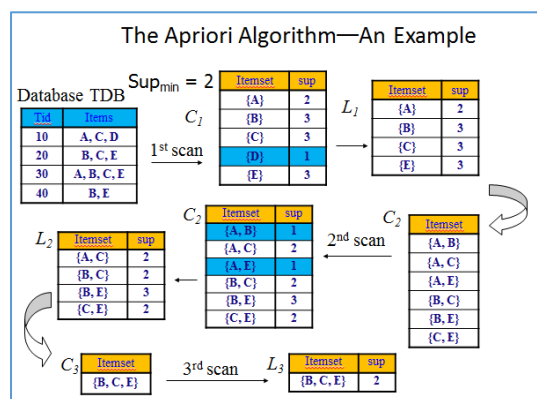
(2) 请写出所有的关联规则。

Association Rules



Rule	Support	Confidence
$A \Rightarrow D$	2/5	2/3
$C \Rightarrow A$	2/5	2/4
$A \Rightarrow C$	2/5	2/3
$B \& C \Rightarrow D$	1/5	1/3

答: 1)求频繁项集



所有频繁项集:  $L_1, L_2, L_3$

2)求关联规则

L1	没关联规则，因为左右没东西。		L3	B=>CE	BCE/B=2/3=0.67<0.8
L2	A=>C	AC/A=2/2=1		C=>BE	BCE/C=2/3=0.67<0.8
	C=>A	AC/C=2/3=0.67<0.8		E=>BC	BCE/E=2/3=0.67<0.8
	B=>C	BC/B=2/3=0.67<0.8		BC=>E	BCE/BC=2/2=1
	C=>B	BC/C=2/3=0.67<0.8		BE=>C	BCE/BE=2/3=0.67<0.8
	B=>E	BE/B=3/3=1		CE=>B	BCE/CE=2/2=1
	E=>B	BE/E=3/3=1			
	C=>E	CE/C=2/3=0.67<0.8			
	E=>C	CE/E=2/3=0.67<0.8			

输出所有 confidence  $> 0.8$  的规则有:  $A \Rightarrow C, B \Rightarrow E, E \Rightarrow B, BC \Rightarrow E, CE \Rightarrow B$

例题 2：现有如下事物数据库，设 min sup=70%, min conf=75%

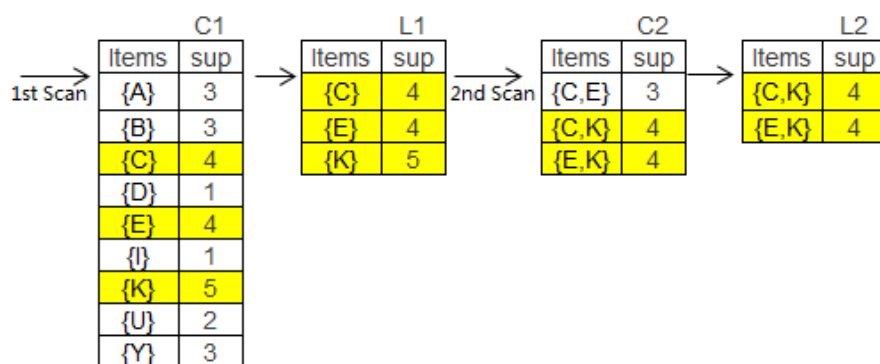
(1) 请用 Apriori 算法找出所有的频繁项目集。（12 分）

(2) 请写出所有的关联规则（8 分）

TID	items bought
T100	{A, B, C, K, E, Y}
T200	{D, B, C, K, E, Y}
T300	{A, K, E}
T400	{A, U, C, K, Y}
T500	{C, B, U, K, I, E}

答：1)求频繁项集

Min\_sup=70%，频度=3.5



所有频繁项集：1-项集（L1）：{C},{E},{K}

2-项集（L2）：{C,K},{E,K}

2)求关联规则

min\_conf=75%=0.75，则

L1	没关联规则，因为左右没东西。	
L2	C=>K	CK/C=4/4=1
	K=>C	CK/K=4/5=0.8
	E=>K	EK/E=4/4=1
	K=>E	EK/K=4/5=0.8

输出所有 confidence>0.75 的规则有：C=>K，K=>C，E=>K，K=>E

## 二、分类和回归 Classification and Regression）：预测建模 有监督的学习（两阶段：学习+预测）

[定义]分类是对有类别对象的数据集进行学习，概括主要特征构建分类模型，用该模型预测未知对象的类别，是有监督学习。

1、什么是分类？常用的分类模型有哪些？

当被预测的属性是范畴型（或离散数据）时,称为分类。常用的模型有 1) 判别模型，如决策树 2) 概率模型，如贝叶斯

2、什么是回归？常用的回归模型有哪些？

当被预测的属性是数量型（或连续数据）时,称为回归。常用的模型有 1)线性回归模型 2)非线性回归模型 3)分段线性模型

3、聚类和分类的区别是什么？他们之间有什么联系？

分类：是对有类别对象的数据集进行学习，概括主要特征构建分类模型，用该模型预测未知对象的类别，是有监督学习。

分类模型用于预测未知记录的类标签。

聚类：是根据给定一组对象的描述信息，发现具有共同特性的对象构成簇，是无监督学习，也可作为其他算法的预处理步骤。

聚类是通过观察学习的过程，而分类是通过例子学习的过程。

这里的观察指的是定义并计算对象间的相似性的过程，而例子指的是训练集。

#### 4、重要的评价指标（3个）

- 「准确率」（P, precision）
- 「召回率」（R, recall）
- F-Measure

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}}$$

$$F_1 = \frac{2PR}{P+R}$$

		Human	
		True	False
Classifier	Yes	a	b
	No	c	d

准确率  $P = a/(a+b)$

召回率  $R = a/(a+c)$

#### 2.1 决策树算法(DT, Decision Tree) (不考算法，只考描述)★★★★

##### 1) 决策树的构造方法和步骤（书 P156）

决策树的生成是一个从根节点开始，从上到下的递归过程。一般采用分而治之的方法，通过不断地将训练样本划分成子集来构造决策树。

假设给定的训练集  $T$  总共有  $m$  个类别。则针对  $T$  构造决策树时，会出现以下三种情况：

- (1) 如果  $T$  中所有样本的类别相同，那么决策树只有一个叶子结点。
- (2) 如果  $T$  中没有可用于继续分裂的变量，则将  $T$  中出现频率最高的类别作为当前结点的类别。
- (3) 如果  $T$  包含的样本属于不同的类别，根据变量选择策略，选择最佳的变量和划分方式将  $T$  分为几个子集  $T_1, T_2, \dots, T_k$ ，每个数据子集构成一个内部结点。

对于某个内部结点继续进行判断，重复上述操作，直到满足决策树的终止条件为止。终止条件就是,结点对应的所有样本属于同一个类别，或者  $T$  中没有可用于进一步分裂的变量。

##### 决策树构建算法 Generate\_decision\_tree。

输入：训练集  $T$ ，输入变量集  $A$ ，目标（类别）变量  $Y$

输出：决策树 Tree

Generate\_decision\_tree( $T, A, Y$ )

- 1:如果  $T$  为空，返回出错信息；
- 2:如果  $T$  的所有样本都属于同一个类别  $C$ ，则用  $C$  标识当前节点并返回；
- 3:如果没有可分的变量，则用  $T$  中出现频率最高的类别标识当前结点并返回；
- 4:根据变量选择策略选择最佳变量  $X$  将  $T$  分为  $k$  个子集 ( $T_1, T_2, \dots, T_k$ )；
- 5:用  $X$  标识当前结点；
- 6:对  $T$  的每一个子集  $T_i$
- 7:NewNode= Generate\_decision\_tree( $T_i, A-X, Y$ )； //递归操作
- 8:生成一个分枝，该分枝由结点  $X$  指向 NewNode；
- 9:返回当前结点。

在上述算法中，结点分裂（第 4 步）是生成决策树的重要步骤。只有根据不同的变量将单个结点分裂成多个结点，方能形成多个类别，因此整个问题的核心就是如何选择分裂变量。

## 2) 决策树算法举例 (ID3)

### Attribute Selection Measure: Information Gain (ID3/C4.5)

- 选择具有最大信息收益的属性
- 用S表示训练集，假设分类属性具有m个不同的值，也就是说共有m个不同的分类  $C_i (i = 1, \dots, m)$ ，用  $s_i$  表示S中属于分类  $C_i$  的样本的个数
- 则信息收益可以用如下三步求出

- 求information:  $I(s_1, s_2, \dots, s_m) = -\sum_{i=1}^m \frac{s_i}{S} \log_2 \frac{s_i}{S}$

- 对每个属性求entropy, 假设属性A的值为  $\{a_1, a_2, \dots, a_v\}$

$$E(A) = \sum_{j=1}^v \frac{s_{1j} + \dots + s_{mj}}{S} I(s_{1j}, \dots, s_{mj})$$

- 对每个属性求information gain:

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A)$$

15

### Attribute Selection by Information Gain Computation

- Class P: buys\_computer = "yes"
- Class N: buys\_computer = "no"
- $I(p, n) = I(9, 5) = 0.940$
- Compute the entropy for age:

age	$p_i$	$n_i$	$I(p_i, n_i)$
$\leq 30$	2	3	0.971
30...40	4	0	0
$> 40$	3	2	0.971

$$E(\text{age}) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$  means "age  $\leq 30$ " has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(\text{age}) = I(p, n) - E(\text{age}) = 0.246$$

Similarly,

$$Gain(\text{income}) = 0.029$$

$$Gain(\text{student}) = 0.151$$

$$Gain(\text{credit\_rating}) = 0.048$$

16

1) 求总信息量  $I(9, 5) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940$

2) 求每个属性的熵

$$\begin{cases} E(\text{age}) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = \frac{5}{14} \left( -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) + \frac{4}{14} \left( -\frac{4}{4} \log_2 \frac{4}{4} + 0 \right) + \frac{5}{14} \left( -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) = 0.694 \\ E(\text{income}) = 0.911 \\ E(\text{student}) = 0.789 \\ E(\text{credit\_rating}) = 0.892 \end{cases}$$

3) 为每个属性求信息收益，选择具有最大信息收益的属性做树根

$$Gain(\text{age}) = I(9, 5) - E(\text{age}) = 0.940 - 0.694 = 0.246$$

$$Gain(\text{income}) = I(9, 5) - E(\text{income}) = 0.940 - 0.911 = 0.029$$

$$Gain(\text{student}) = I(9, 5) - E(\text{student}) = 0.940 - 0.789 = 0.151$$

$$Gain(\text{credit\_rating}) = I(9, 5) - E(\text{credit\_rating}) = 0.940 - 0.892 = 0.048$$

由于 age 的信息收益最大，选它作为树根，用同样的方法求每一个中间节点，直到构造出决策树模型。

## 2.2 贝叶斯算法 (Bayes) (考大题 20 分) ★★★★★

### 1) 贝叶斯公式 (会变形)

$$P(B|A) = P(AB)/P(A) \Rightarrow P(AB)=P(B|A)*P(A) \Rightarrow P(AB)=P(A|B)*P(B)$$

$$P(B|A) = P(A|B) * P(B) / P(A) \Rightarrow P(A|B)=P(B|A)*P(A)/P(B)$$

### 2) 为什么朴素贝叶斯的算法是朴素的? (Naïve Bayes, NB)

朴素贝叶斯分类之所以称之为“朴素”的,是因为在分类的计算过程中做了一个朴素的假设,假定属性值之间是相互独立的。

该假设称作类条件独立,做此假设的目的是为了简化计算。(书 P164)

例题 3: 下面数据表已离散化,请问如果用朴素贝叶斯分类法的话,给定新元组的类标签应该是什么?请写出计算步骤

Training dataset

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Class: C1:buys\_computer='yes'

C2:buys\_computer='no'

Data sample: X =(age<=30,Income=medium,Student=yes,Credit\_rating=Fair)

答: NewData

输入向量 X				Y
age<=30	medium	Student	Fair	?

根据贝叶斯公式 (1) 假设 Buys\_computer=yes 时,  $P(\text{Yes}|X) = \frac{P(X|\text{Yes})P(\text{Yes})}{P(X)}$

(2) 假设 Buys\_computer=no 时,  $P(\text{No}|X) = \frac{P(X|\text{No})P(\text{No})}{P(X)}$

利用上表作为训练数据,使用朴素贝叶斯分类法(属性直接的条件独立假设),为每一个类别算一个概率

$$P(\text{Yes}|X) \propto P(X|\text{Yes})P(\text{Yes}) = P(\text{age} \leq 30|\text{Yes}) P(\text{medium}|\text{Yes}) P(\text{Student}|\text{Yes}) P(\text{Fair}|\text{Yes}) P(\text{Yes}) = \frac{2}{9} * \frac{4}{9} * \frac{6}{9} * \frac{6}{9} * \frac{9}{14} = 0.028$$

$$P(\text{No}|X) \propto P(X|\text{No})P(\text{No}) = P(\text{age} \leq 30|\text{No}) P(\text{medium}|\text{No}) P(\text{Student}|\text{No}) P(\text{Fair}|\text{No}) P(\text{No}) = \frac{3}{5} * \frac{2}{5} * \frac{1}{5} * \frac{2}{5} * \frac{5}{14} = 0.007$$

$P(\text{Yes}|X) > P(\text{No}|X)$ , 故该新元组类别属于 C1: buys\_computer='yes'

### Naïve Bayesian Classifier: Example 1

#### ■ Compute $P(X|C_i)$ for each class

$$P(\text{buys\_computer}=\text{"yes"}) = 9/14=0.643$$

$$P(\text{buys\_computer}=\text{"no"}) = 5/14=0.357$$

$$P(\text{age} < 30 | \text{buys\_computer}=\text{"yes"}) = 2/9=0.222$$

$$P(\text{income}=\text{"medium"} | \text{buys\_computer}=\text{"yes"}) = 4/9 = 0.444$$

$$P(\text{student}=\text{"yes"} | \text{buys\_computer}=\text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit\_rating}=\text{"fair"} | \text{buys\_computer}=\text{"yes"}) = 6/9=0.667$$

$$P(\text{age} < 30 | \text{buys\_computer}=\text{"no"}) = 3/5 = 0.6$$

$$P(\text{income}=\text{"medium"} | \text{buys\_computer}=\text{"no"}) = 2/5 = 0.4$$

$$P(\text{student}=\text{"yes"} | \text{buys\_computer}=\text{"no"}) = 1/5=0.2$$

$$P(\text{credit\_rating}=\text{"fair"} | \text{buys\_computer}=\text{"no"}) = 2/5=0.4$$

X=(age<=30, income=medium, student=yes, credit\_rating=fair)

$$P(X|C_1) : P(X|\text{buys\_computer}=\text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$

$$P(X|\text{buys\_computer}=\text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$

$$P(X|C_1)*P(C_1) : P(X|\text{buys\_computer}=\text{"yes"}) * P(\text{buys\_computer}=\text{"yes"}) = 0.044 \times 0.643 = 0.028$$

$$P(X|\text{buys\_computer}=\text{"no"}) * P(\text{buys\_computer}=\text{"no"}) = 0.019 \times 0.357 = 0.007$$

Therefore, X belongs to class "buys\_computer=yes"

注: 此图为老师 ppt 解法

例题 4：在下面的雇员数据表中，数据已经被离散化，如“31...35”代表年龄在 31 到 35 之间的人。

Count 表示该行在数据集重复出现的次数。假设 Status 是类标签属性，给定新元组“Systems, 31...35, 41k...45k”，请问如果用朴素贝叶斯分类法的话，该元组的类标签应该是什么？请写出计算步骤（20 分）

department	status	age	salary	count
sales	senior	31...35	46k...50k	30
sales	junior	26...30	26k...30k	40
sales	junior	31...35	31k...35k	40
systems	junior	21...25	46k...50k	20
systems	senior	31...35	66k...70k	5
systems	junior	26...30	46k...50k	3
systems	senior	41...45	66k...70k	3
marketing	senior	36...40	46k...50k	10
marketing	junior	31...35	41k...45k	4
secretary	senior	46...50	36k...40k	4
secretary	junior	26...30	26k...30k	6

答：

Class: C1: status=senior

C2: status =junior

输入向量 X=( Systems, 31...35, 41k...45k)

每个类别的  $P(C_i)$ ， $i=1,2$ 。根据训练样本计算  $P(\text{Senior}) = \frac{52}{165}$   $P(\text{Junior}) = \frac{113}{165}$

为了计算  $P(X|C_i)$ ， $i=1,2$  需要先计算下列条件概率：

$$P(\text{Systems}|\text{Senior}) = \frac{8}{52} \quad P(31\cdots35|\text{Senior}) = \frac{35}{52} \quad P(41k\cdots45k|\text{Senior}) = \frac{0}{52}$$

$$P(\text{Systems}|\text{Junior}) = \frac{23}{113} \quad P(31\cdots35|\text{Junior}) = \frac{44}{113} \quad P(41k\cdots45k|\text{Junior}) = \frac{4}{113}$$

根据贝叶斯公式：（1）假设 status=senior 时， $P(\text{senior}|X) = \frac{P(X|\text{senior})P(\text{senior})}{P(X)}$

（2）假设 status =junior 时， $P(\text{junior}|X) = \frac{P(X|\text{junior})P(\text{junior})}{P(X)}$

利用上表作为训练数据，使用朴素贝叶斯分类法（属性直接的条件独立假设），为每一个类别算一个概率

$P(\text{Senior}|X) \propto P(X|\text{Senior})P(\text{Senior}) = P(\text{Systems}|\text{Senior}) P(31\cdots35|\text{Senior}) P(41k\cdots45k|\text{Senior}) P(\text{Senior})$

$$= \frac{8}{52} * \frac{35}{52} * \frac{0}{52} * \frac{52}{165} = 0$$

$P(\text{Junior}|X) \propto P(X|\text{Junior})P(\text{Junior}) = P(\text{Systems}|\text{Junior}) P(31\cdots35|\text{Junior}) P(41k\cdots45k|\text{Junior}) P(\text{Junior})$

$$= \frac{23}{113} * \frac{44}{113} * \frac{4}{113} * \frac{113}{165} = 0.002$$

由于 salary 在训练集中没有出现结果为 0，所以采用平滑方法：

$$P(x_j|c_i) = \frac{(\text{count}(x_j, c_i)) + mp}{(\text{count}(c_i)) + m} = \frac{(\text{count}(x_j, c_i)) + 1}{(\text{count}(c_i)) + 2} \quad \text{其中 } m=|C|=2, p=1/|C|=1/2$$

那么，

$P(\text{Senior}|X) \propto P(X|\text{Senior})P(\text{Senior}) = P(\text{Systems}|\text{Senior}) P(31\cdots35|\text{Senior}) P(41k\cdots45k|\text{Senior}) P(\text{Senior})$

$$= \frac{8+1}{52+2} * \frac{35+1}{52+2} * \frac{0+1}{52+2} * \frac{52}{165} = 0.0006$$

$P(\text{Junior}|X) \propto P(X|\text{Junior})P(\text{Junior}) = P(\text{Systems}|\text{Junior}) P(31\cdots35|\text{Junior}) P(41k\cdots45k|\text{Junior}) P(\text{Junior})$

$$= \frac{23+1}{113+2} * \frac{44+1}{113+2} * \frac{4+1}{113+2} * \frac{113}{165} = 0.0024$$

$P(\text{Senior}|X) < P(\text{Junior}|X)$ ，故该新元组类别属于 C2: status= Junior

2.3 支持向量机（SVM，Support Vector Machine）（复杂一般不考）

任务是寻找一种不同类别间的差异最大化的函数；核心思路是通过构造分割面将数据进行分离。  
能做分类（对离散数据进行预测），也能做回归（对连续数据进行预测）

2.4 人工神经网络（ANN，Artificial Neural Networks）（复杂一般不考）

后向传播算法（BP，Back-Propagation）[经典]：迭代的对训练集中的每个样本进行处理。  
学习过程：

- 1) 初始化权（只作一次）
- 2) 向前传播输入（迭代 n 次）
- 3) 向后传播误差（迭代 n 次）

2.5 邻近算法（KNN，k-NearestNeighbor）（复杂一般不考）

每个训练样本都看作 n 维空间中的一个点。  
给定一个未知样本（类似于查询点 q）， 首先找到该样本的 k 个近邻，将这 k 个近邻按照类标号进行分组，未知样本最终被分到组员最多的那个组。

KNN 和 K-Means 的区别	
KNN	K-Means
1.KNN 是分类算法	1.K-Means 是聚类算法
2.监督学习	2.非监督学习
3.喂给它的数据集是带 label 的数据，已经是完全正确的数据	3.喂给它的数据集是无 label 的数据，是杂乱无章的，经过聚类后才变得有点顺序，先无序，后有序
没有明显的前期训练过程，属于 memory-based learning	有明显的前期训练过程
K 的含义：来了一个样本 x，要给它分类，即求出它的 y，就从数据集中，在 x 附近找离它最近的 K 个数据点，这 K 个数据点，类别 c 占的个数最多，就把 x 的 label 设为 c	K 的含义：K 是人工固定好的数字，假设数据集可以分为 K 个簇，由于是依靠人工定好,需要一点先验知识
相似点：都包含这样的过程，给定一个点，在数据集中找离它最近的点。即二者都用到了 NN(Nears Neighbor)算法，一般用 KD 树来实现 NN。	

2.6 回归问题（也叫预测，是等价的） - 常用方法是最小二乘法求回归系数（太基础一般不考）

最小二乘法(又称最小平方法，Least squares)是一种数学优化技术。它通过最小化误差的平方和寻找数据的最佳函数匹配。  
利用最小二乘法可以简便地求得未知的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小。  
最小二乘法还可用于曲线拟合。  
 $Y = \alpha + \beta X$  （ $\alpha$ 、 $\beta$ 是回归系数）



### 三、聚类 Clustering: 描述建模 无监督的学习 (没有明显的两阶段, 没有 train data)

[定义] 聚类是根据数据的特征将数据分组, 使具有最大的组内相似性和最小的组间相似性, 是无监督学习。

#### 1、聚类的两种方法

(1) 基于相似度法 (硬聚类): K-Means, BRICH 等

[基于相似度方法] 适合于大数据量聚类的 BRICH 算法, 有 3 大贡献: (不考)

- ① 提出了聚类特征 CF (Clustering Feature)    ② 聚类特征 CF 满足线性可加性    ③ 用 B-tree 的性质实现了 CF-tree
- 基于密度法 (硬聚类): DBSCAN, OPTICS (基于密度也可以理解成基于相似度)

(2) 基于模型法[即概率法] (软聚类): GMM 等

#### 2、对象间的相似性是聚类分析的核心

(1) 对象的属性分类

- 1) 区间标度型变量    2) 二元变量(0 或 1)    3) 分类型变量    4) 序数型变量    5) 比例型变量    6) 混合型

(2) 区间标度型对象之间的相似度 (或相异度) 计算

是基于对象间距离来计算的, 通常用: ① 名考斯基距离, ② 当  $q=1$  时,  $d$  称为曼哈顿距离; ③ 当  $q=2$  时,  $d$  称为欧几里得距离

通常使用明考斯基距离

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

其中  $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ ,  $j = (x_{j1}, x_{j2}, \dots, x_{jp})$   
分别代表两个  $p$ -维的对象,  $q$  是一个正整数

$q = 1$  的时候,  $d$  称为曼哈顿距离

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

当  $q=2$  表示欧几里得距离

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

(最常用) 曼哈顿函数表示两个点在标准坐标系上的绝对轴距总和, 坐标  $(x1, y1)$  的  $i$  点与坐标  $(x2, y2)$  的  $j$  点的曼哈顿距离为:  $d(i, j) = |X1 - X2| + |Y1 - Y2|$ 。找曼哈顿距离最近的那个点就是新的中心点。

#### 3、主要的聚类算法 (5 种)

1) 基于划分的算法: 给定一个  $n$  个对象或元组的数据库, 将数据划分为  $k$  个组 ( $k$  是事先给定的,  $k \leq n$ )。如 K-Means  
[定义] K-Means 方法是 MacQueen 1967 年提出的。给定一个数据集  $X$  和一个整数  $K (K \leq n)$ , K-Means 方法是将  $X$  分成  $K$  个聚类并使得在每个聚类中所有值与该聚类中心距离的总和最小。

2) 基于层次的算法: 是把数据对象排列成一个聚类树, 在需要的层次上对其进行切割, 关联的部分构成一个 cluster。  
有两种类型 ① 聚合层次聚类; ② 划分层次聚类。如 BRICH

3) 基于密度的算法: 绝大多数划分方法基于对象之间的距离进行聚类, 只能发现凸状的簇。如 DBSCAN, OPTICS

4) 基于方格的算法: 把多维空间划分成一定数目的单元(cell), 然后在这种数据结构上进行聚类操作, 速度快, 但不精确

5) 基于模型的算法: ① 基于神经网络的方法

② 基于概率统计的方法, 典型方法有高斯混合模型 GMM (Gaussian Mixture Models), 典型应用有话题检测 (EM)

#### 4、好的聚类算法特点 Requirements of Clustering in Data Mining

- 可伸缩性
- 能够处理各种不同类型的属性
- 能够发现任意形状的聚类
- 在决定输入参数的时候, 对领域知识的需求要小
- 能够处理噪声和异常点
- 对输入数据的顺序不敏感
- 可以处理高维数据
- 可以和用户制定的限定条件相结合
- 可解释性和使用性好

### 3.1 K-Means 算法【经典】（考大题 20 分）★★★★★

1) K-Means 聚类方法分为以下几步：

- [1] 给 K 个 cluster 选择最初的中心点，称为 K 个 Means。
- [2] 计算每个对象和每个中心点之间的距离。（计算  $n*k$  这么多个距离）
- [3] 把每个对象分配给距它最近的中心点所属的 cluster。
- [4] 重新计算每个 cluster 的中心点。
- [5] 重复 2, 3, 4 步，直到算法收敛。

ppt 例题（一维的数据，一般不考）

Cluster		C1	C2	C3
Centroid Value		1	20	40
P1	1	0	19	39
P2	3	2	17	37
P3	5	4	15	35
P4	8	7	12	32
P5	9	8	11	31
P6	11	10	9	29
P7	12	11	8	28
P8	13	12	7	27
P9	37	36	17	3
P10	43	42	23	3
P11	45	44	25	5
P12	49	48	29	9
P13	51	50	31	11
P14	65	64	45	25

Cluster		C1	C2	C3
Centroid Value		5	12	48
P1	1	4	11	47
P2	3	2	9	45
P3	5	0	7	43
P4	8	3	4	40
P5	9	4	3	39
P6	11	6	1	37
P7	12	7	0	36
P8	13	8	1	35
P9	37	32	25	11
P10	43	38	31	5
P11	45	40	33	3
P12	49	44	37	1
P13	51	46	39	3
P14	65	60	53	17

Cluster		C1	C2	C3
Centroid Value		4	1	48
P1	1	3	10	47
P2	3	1	8	45
P3	5	1	6	43
P4	8	4	3	40
P5	9	5	2	39
P6	11	7	0	37
P7	12	8	1	36
P8	13	9	2	35
P9	37	33	26	11
P10	43	39	32	5
P11	45	41	34	3
P12	49	45	38	1
P13	51	47	40	3
P14	65	61	54	17

Cluster		C1	C2	C3
Centroid Value		3	10	48
P1	1	2	9	47
P2	3	0	7	45
P3	5	2	5	43
P4	8	5	2	40
P5	9	6	1	39
P6	11	8	1	37
P7	12	9	2	36
P8	13	10	3	35
P9	37	34	27	11
P10	43	40	33	5
P11	45	42	35	3
P12	49	46	39	1
P13	51	48	41	3
P14	65	62	55	17

第一步

- 把 14 个人分成 3 组，只有一个属性，年龄
- 初始的 centroids 是 1, 20, 40；右边的表是完成步骤 1, 2 后的结果

第二步

- 重新计算 centroid，得到 5, 12 和 48
- 重新计算每个实例与 3 个 Cluster 的距离；P5 更接近 C2；
- 需要重新计算 C1 和 C2 的 centroid，C3 没有变化不需要重新计算

第三步

- 3 个 Cluster 的 centroid 是 4, 11 和 48
- 计算每个实例到 Cluster 的距离；P4 更接近 C2；
- 需要重新计算 C1 和 C2 的 centroid，C3 没有变化不需要重新计算

第四步

- 3 个 Cluster 的 centroid 是 3, 10 和 48
- 计算每个实例到 Cluster 的距离；没有任何变化，算法不再迭代

例题 5：（二维的数据，考试一般考 2 次迭代）

假设数据挖掘的任务是将如下的 8 个点（用(x, y)代表位置）聚类为三个类。

A1(2,3), A2(3,5), A3(10,7), A4(1,6), A5(3,8), A6(6,4), A7(5,1), A8(9,2), 距离函数是曼哈顿函数。

假设初始我们选择 A2, A5, A7 为每个聚类的中心, 请用 k-means 算法给出第一次、第二次循环执行后三个聚类中心。

答：1、第一次迭代：1) 选择三个初始中心：A2, A5, A7

2) 计算 8\*3 个距离

		A1(2,3)	A2(3,5)	A3(10,7)	A4(1,6)	A5(3,8)	A6(6,4)	A7(5,1)	A8(9,2)
C1	A2(3,5)	3	0	9	3	3	4	6	9
C2	A5(3,8)	6	3	8	4	0	7	9	12
C3	A7(5,1)	5	6	11	9	9	4	0	5

注：找曼哈顿距离最近的那个点就是新中心点， $d(i,j)=|X1-X2|+|Y1-Y2|$ 。如 A1(2,3)到 A2(3,5)的  $d=|2-3|+|3-5|=3$ , 红色为距中心点最近的

3) 将每一个对象分配给离自己最近的 cluster（中心），第一次循环结果：

Cluster1: A1、A2、A4

Cluster2: A3、A5

Cluster3: A6、A7、A8

2、第二次迭代：1) 算出每个 cluster 的新中心

Cluster1 的中心： $(\frac{2+3+1}{3}, \frac{3+5+6}{3}) = (2, 4)$

Cluster2 的中心： $(6, 7)$

Cluster3 的中心： $(\frac{6+5+9}{3}, \frac{4+1+2}{3}) = (6, 2)$

2) 计算 8\*3 个距离

		A1(2,3)	A2(3,5)	A3(10,7)	A4(1,6)	A5(3,8)	A6(6,4)	A7(5,1)	A8(9,2)
C1	(2,4)	1	2	11	3	5	4	6	9
C2	(6,7)	8	5	4	6	4	3	7	8
C3	(6,2)	5	6	9	9	9	2	2	3

3) 重新分组，第二次循环结果：

Cluster1: A1、A2、A4

Cluster2: A3、A5

Cluster3: A6、A7、A8

因为第一次和第二次结果一样，直接就收敛了，结束了。

例题 6：假设数据挖掘的任务是将如下的 8 个点（用(x, y)代表位置）聚类为三个类。

A1(1,9), A2(2,8), A3(5,4), B1(6,8), B2(10,5), B3(6,8), C1(10,2), C2(4,8), C3(2,6) 距离函数是曼哈顿函数。

假设初始我们选择 A1, B1 和 C1 为每个聚类的中心, 请用 k-means 算法给出在第一次循环执行后的三个聚类中心。(20 分)

答：曼哈顿距离：如 2 维： $d=|x1-x2|+|y1-y2|$

族	C1	C2	C3	族	C1	C2	C3	族	C1	C2	C3
族中心点	1, 9	6, 8	10, 2	族中心点	2, 8	5, 7	10, 4	族中心点	2, 8	6, 7	10, 4
A1	1	9	0	A1	1	9	2	A1	1	9	2
A2	2	8	2	A2	2	8	0	A2	2	8	0
A3	5	4	9	A3	5	4	7	A3	5	4	7
B1	6	8	6	B1	6	8	4	B1	6	8	4
B2	10	5	13	B2	10	5	11	B2	10	5	11
B3	6	8	6	B3	6	8	4	B3	6	8	4
C1	10	2	16	C1	10	2	14	C1	10	2	14
C2	4	8	4	C2	4	8	2	C2	4	8	2
C3	2	6	4	C3	2	6	2	C3	2	6	2
第一族中心	$(1+2+2)/3$			第一族中心	$(1+2+4+2)/4$			第一族中心	$(1+2+8+6)/4$		
	2				2				2		
第二族中心	$(5+6+6+4)/4$			第二族中心	$(5+6+6)/3$			第二族中心	$(4+8+8)/3$		
	5				6				7		
第三族中心	$(10+10)/2$			第三族中心	$(10+10)/2$			第三族中心	$(5+2)/2$		
	10				10				4		

第一次循环后三个族聚类中心分别为 (2, 8)、(5, 7)、(10, 4)

第二次循环后三个族聚类中心分别为 (2, 8)、(6, 7)、(10, 4)（注：这部分以下题目没要求可以不写）

第三次循环后三个族聚类中心分别为 (2, 8)、(6, 7)、(10, 4)

因为第二次和第三次结果一样，直接就收敛了，结束了。

## 2) K-means 算法的优缺点 (书 P188) ★★★

**k-Means 方法具有下面的优点:**

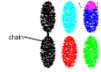
- (1) 对于处理大数据量具有可扩充性和高效率。算法的复杂度是  $O(tkn)$ , 其中  $n$  是对象的个数,  $k$  是 cluster 的个数,  $t$  是循环的次数, 通常  $k, t \ll n$ 。
- (2) 可以实现局部最优化。

**K-Means 方法也有以下缺点:**

- (1) 族的个数  $K$  必须事先确定。在有些应用中, 事先确定族的个数非常难。
- (2) 无法找出具有特殊形状的族 (如图 10.4 所示)。
- (3) 必须给出  $k$  个初始中心点。如果这些初始中心点选择不好的话, 聚类的质量将会非常差。如图 10.5 所示, 由于初始中心点选得不好, 最后形成的聚类结果明显很差。
- (4) 对异常数据过于敏感。异常数据的存在将对中心点的计算产生极大影响。
- (5) 求中心点的时候, 需要计算算术平均。无法适用于具有分类属性的数据。



(书图 10.4 和 10.5)



(书图 10.10)



(书图 10.11 核心距离和可达距离示例)

## 3.2 DBSCAN 算法的优缺点 (书 P191) ★★★

**DBSCAN 方法的优点是:**

- (1) 不需要事先确定族的个数;
- (2) 聚类速度快, 使用索引 (例如  $R^*$ -tree) 时, DBSCAN 的时间复杂度为  $O(n \log n)$ ,  $n$  为数据库中数据对象的个数, 否则, DBSCAN 的时间复杂度为  $O(n^2)$ ;
- (3) 对噪声数据不敏感;
- (4) 能发现任意形状的族, 例如, DBSCAN 可以找出如图 10.4 所示的族。

**DBSCAN 的缺点是:**

- (1) 输入参数  $\epsilon$  和  $\text{MinPts}$  的值较难确定。
- (2) 当数据库中数据对象的密度分布不均匀时, 用相同的参数值可能得不到好的聚类结果。
- (3) 可能会产生“链条”现象。如图 10.10 所示, 左边的上、下两个本应独立的族连接在了一起, 产生了类似“链条”的现象。
- (4) 使用  $R^*$ -tree 索引时, 由于  $R^*$ -tree 在高维空间中不够有效, 导致 DBSCAN 算法在处理高维数据时性能下降。

## 3.3 OPTIC 算法 (复杂一般不考)

**核心理想:** 为每个数据对象计算出一个顺序值 (ordering)。这些值代表了数据对象的基于密度的族结构, 位于同一个族的数据对象具有相近的顺序值。根据这些顺序值将全体数据对象用一个图示的方式排列出来, 根据排列的结果就可以得到不同层次的族。

基于这个思想, 每个数据对象需要存储两个值, 一个是核心距离 (core-distance), 另一个是可达距离 (reach-distance)。

**核心距离:** 给定一个数据对象集合  $D$ , 两个参数  $\epsilon$  和  $\text{MinPts}$ , 一个对象  $O$ , 如果  $O$  是一个核心对象, 则  $O$  的核心距离 (core-dist) 是使得  $O$  能成为核心对象的最小半径值 (该值小于等于  $\epsilon$ )。如果  $O$  不是核心对象, 则  $O$  的核心距离没有定义。

**可达距离:** 给定一个数据对象集合  $D$ , 两个参数  $\epsilon$  和  $\text{MinPts}$ , 一个对象  $O$ , 如果  $O$  是一个核心对象, 则  $O$  与另一个对象  $p$  间的可达距离 (reachability-distance) 是  $O$  的核心距离和  $O$  与  $p$  的欧几里得距离之间的较大值。如果  $O$  不是一个核心对象,  $O$  与  $p$  之间的可达距离没有定义。

## 3.4 常用聚类方法对比 (K-means, DBSCAN, OPTICS) ★★★★★

	K-means	DBSCAN	OPTICS
参数	$K$	$\epsilon, \text{MinPts}$	$\epsilon, \text{MinPts}$
形状	凸状的簇	任意形状	任意形状
密度变化	无法应对	无法应对	可应对
鲁棒	对异常值敏感	对异常值不敏感	对异常值不敏感