《数据挖掘》复习整理

1.整理基于老师上课的 PPT，我和书（中文第三版）作了对照，在后面标有页码。
2.绝大部分名词后有英文，便于和 PPT 对照记忆理解。
3.部分内容直接剪切于 PPT。水平有限，不足之处请及时指正。——王晋东

第一章 数据仓库（Data_warehouse.pdf，涉及书上第 4、5 章）

1.数据仓库四个特点：面向主题的、集成的、时变的、非易失的，支持管理者的决策过程。（P83）.

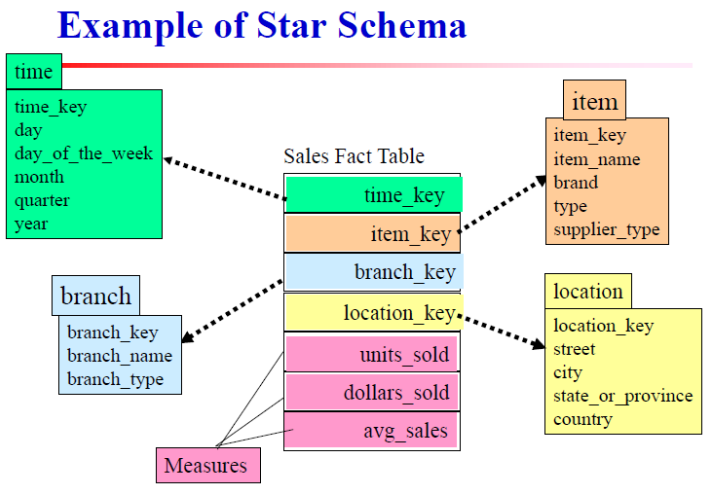
2.OLTP：联机事务处理，执行日常事务和查询；OLAP：联机分析处理，高层次分析数据。区别（P85）。

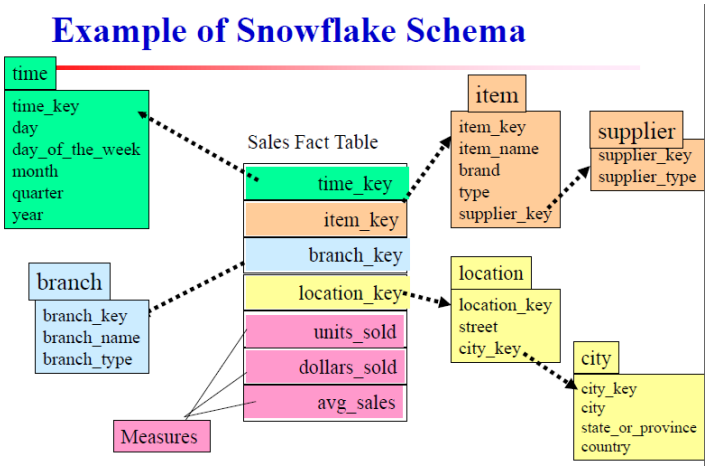
3.数据仓库和传统异构数据库的区别：数据仓库是更新驱动，异构数据库是查询驱动。（P83）。

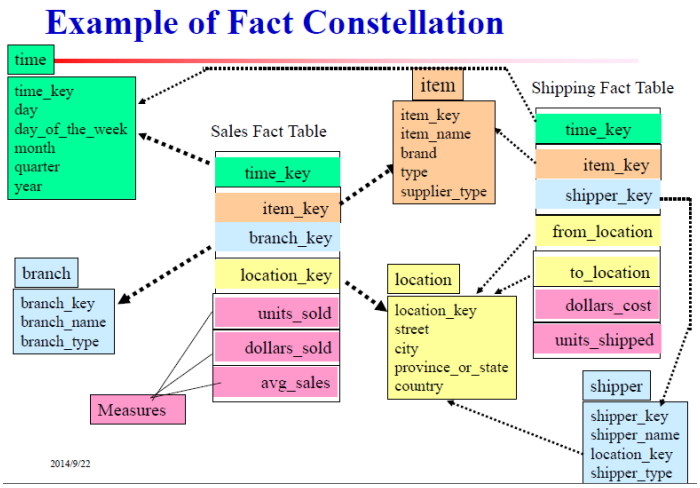
4.为什么要分离的数据仓库？（P85，3 点）。

5.数据仓库的三层结构：（1）底层是仓库数据库服务器，几乎总是一个关系数据库系统；（2）中间层是 OLAP 服务器；（2）顶层是前端客户层。（P86）

6.数据仓库和 OLAP 工具基于多给数据模型。数据立方体允许对多维数据进行建模和观察，它由维和事实定义。三种模式：星形、雪花形和事实星座。







7.DMQL 语句：

（1）创建 CUBE：Create cube <cube name> [<dimension list>]:[measure list]

（2）维表定义：define dimension <dimension name> as (<attribute_or_subdimension_list>)

（3）共享的维表定义：define dimension <dimension_name> as <dimension_name_first_time> in cube <cube_name_first_time>

8.典型的 OLAP 操作（P96-97）：

（1）上卷（roll-up）：沿一个维的概念向上爬升进行聚集。

（2）下钻（drill-down）：和上卷相反，由不详细到详细。

（3）切片和切块（slice, dice）：选择投影一个维（切片）或者多个维（切块）数据。

（4）转轴（pivot）：转动数据的视角，目测没用过。

9.三种数据仓库模型：企业模型、数据集市、虚拟仓库。

10.元数据：数据的数据，是定义仓库对象的数据。（P88）

11.OLAP 的服务器架构：关系 OLAP 服务器（ROLAP）、多维 OLAP 服务器（MOLAP）、混合 OLAP 服务器（HOLAP）、特殊 OLAP 服务器。（P107-108）

12.数据立方体计算：在 n 维立方体中的 cuboids 有 2^n 个。

第二章 数据预处理（Preprocessing.pdf，涉及书上第 2、3 章）

13.为什么要进行数据预处理：脏数据、不完整、噪声、不连续、重复数据（P55-56）.

14.数据预处理主要内容：数据清理、数据集成、数据规约、数据变换（P56-57）.

15.数据描述度量：

中心趋势度量（P30）：

（1）平均数（mean）：注意算术平均和加权平均。

（2）中位数（median）：计算方法见书上描述。

（3）众数（mode）：mean – mode = 3 * (mean - median)

度量数据散布（P32）：

（4）*IQR*：*IQR* = *Q*3 - *Q*1。

（5）极差：max – min。

（6）五数概括：min、*Q*1、median、*Q*3、max。

（7）方差和标准差：

$$s^2=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2=\frac{1}{n-1}[\sum_{i=1}^{n}x_i^2-\frac{1}{n}(\sum_{i=1}^{n}x_i)^2] \qquad \sigma^2=\frac{1}{N}\sum_{i=1}^{n}(x_i-\mu)^2=\frac{1}{N}\sum_{i=1}^{n}x_i^2-\mu^2$$

16.盒图（Boxplot，P33）：

（1）端点在四分位数上，使得盒的长度是四分位数极差 *IQR*。

（2）中位数用盒内的线标记。

（3）胡须延伸至最小和最大值。

17.qq 图、q 图、直方图、散点图（P34-36）。

18.数据清理：填充缺失值、检测边界光滑噪声、改正不连续数据、清除冗余。

对缺失数据处理、噪声处理（P58-59）.

19.分箱方法：等宽（每个箱子区间长度一样）、等频（每个箱子数的个数一样）。（P59）

20.数据变换（P75-76）：

（1）最小-最大规范化（min-max normalization）：

（2）z-score normalization：

（3）小数定标规范化（normalization by decimal scaling）：

Min-max normalization: to [new_min$_A$, new_max$_A$]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,600 is mapped to $\frac{73,600-12,000}{98,000-12,000}(1.0-0)+0=0.716$

Z-score normalization (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

$\frac{73,600-54,000}{16,000}=1.225$

- Ex. Let μ = 54,000, σ = 16,000. Then

Normalization by decimal scaling

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max(|v'|) < 1

- Ex. (-986, 917) => (-0.986, 0.917), j=3

21.数据集成（P62-63）：

（1）数值数据（Numerical Data）的相关系数：Pearson's product moment coefficient

$$r_{A,B} = \frac{\sum (a_i - \bar{A})(b_i - \bar{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum (a_i b_i) - n\bar{A}\bar{B}}{(n-1)\sigma_A\sigma_B}$$

根据结果和 0 的关系来判断是正相关、负相关还是不相关。

（2）标称数据（Categorical Data）的卡方检验：

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

22.数据规约：

（1）属性子集选择（Attribute Subset Selection）：目标是找出最小属性集。常用的启发式方法：逐步向前选择、逐步向后选择、逐步向前和逐步向后删除的组合、决策树归纳。（P69）

（2）数据压缩（Data Compression）：常用方法：离散小波变换（DWT，适合高维数据）、主

成分分析（PCA，更好地处理稀疏数据）。（P66-68）
（3）回归模型、直方图、聚类、抽样（P69-71）。
23.数据离散化和概念分层：离散化是将数据分配到指定的区间，减少给定连续属性值的个数；概念分层是使用高层的概念（如老年、青年）来代替实际数据（如实际年龄）。
（1）常用的离散化模型：分箱、直方图、聚类（P76）。
（2）基于熵（Entropy）的离散化：
熵的定义：对于集合 S 中的 m 个类来说，

$$Entropy(S) = -\sum_{i=1}^{m} p_i \log_2(p_i) \quad \text{where } p_i \text{ is the probability of class } i \text{ in } S$$

如果将 S 用边界 T 分成两个类 S1 和 S2，则分后的熵是：

$$Entropy(S,T) = \frac{|S_1|}{|S|} Entropy(S_1) + \frac{|S_2|}{|S|} Entropy(S_2)$$

所有可能分的 T 中，使得信息增益最大化的 T 被选中来作为二类分类边界：

$$Gain(S,T) = Entropy(S) - Entropy(S,T)$$

（3）3-4-5 规则：如果一个区间最高有效位上包含 3，6，7 或 9 个不同的值，就将该区间划分为 3 个等宽子区间；(7->2-3-2)；如果一个区间最高有效位上包含 2，4，或 8 个不同的值，就将该区间划分为 4 个等宽子区间；如果一个区间最高有效位上包含 1，5，或 10 个不同的值，就将该区间划分为 5 个等宽子区间；将该规则递归的应用于每个子区间，产生给定数值属性的概念分层。
（4）分类数据的层次概念分层：有四种方法（P77-78）。

第三章 关联规则挖掘（ARM.pdf，涉及书上第 6、7 章）

24.基本概念（P159）：
（1）项集（item set）、支持度（*support*）、置信度（*confidence*）、频繁模式（frequent pattern）等都较简单，见 P158-159。
（2）最小支持度阈值（*min_support*）、最小置信度阈值（*min_confidence*）。
（3）置信度计算：

$$\text{Confidence}(\alpha \Rightarrow \beta) = P(\beta/\alpha) = \frac{P(\alpha \cap \beta)}{P(\alpha)} = \frac{\text{count}(\alpha \cap \beta)}{\text{count}(\alpha)}$$

（4）关联事件的度量（*lift*）计算：

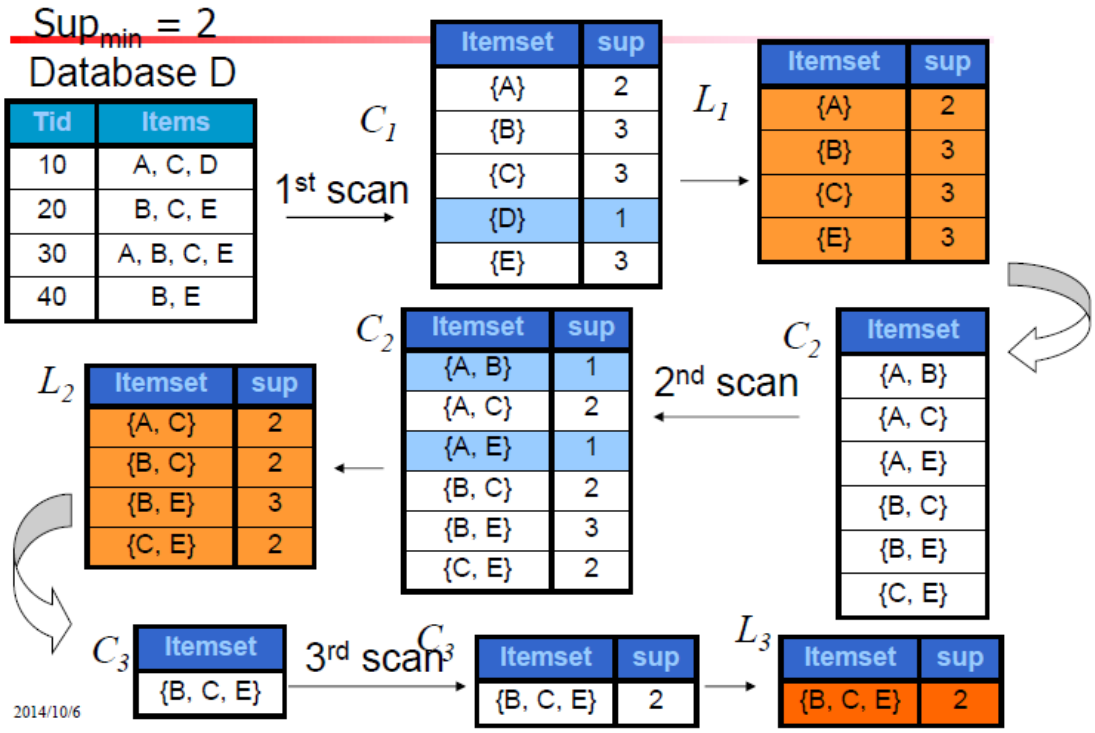
$$lift = \frac{P(A \cap B)}{P(A)P(B)}$$

（5）几种关联模式：
a.布尔类型：仅考虑到该项出现还是不出现，如 buys(x, “SQLServer”) ^ buys(x, “DMBook”) => buys(x, “DBMiner”) [0.2%, 60%]。
b.数值类型：考虑连续的数，如 age(x, “30…39”) ^ income(x, “42…48K”) => buys(x, “high resolution TV”) [1%, 75%]。
c.单层-多层关联：age(x, “30…39”) => buys(x, “laptop computer”)，age(x, “30…39”) => buys(x, “computer”)。
d.单维-多维关联等。

25.*Apriori* 算法（P160 起）：

（1）先验性质：频繁项集的所有非空子集也一定是频繁的。挖掘频繁模式的基本方法。

（2）算法具体过程：见书上所述，注意每一步连接到要考虑 *min_support*，把不满足条件的除掉。一个例子：



（3）对算法的优化：散列技术、事务压缩、划分、抽样及动态项集计数（P165-166）。

（4）算法缺点：需要多次扫描数据库，代价太大。

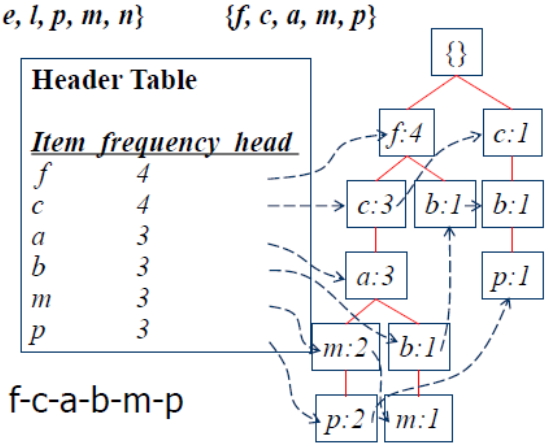
26.频繁模式增长（*FP*-growth）：（P166 起）

（1）基本思想：构造 *FP* 树。首先按照项的 *support* 从大到小排序，第二次扫描数据库时构造 FP 树。借助构造好的树来挖掘频繁模式基（Conditional Pattern Base）和频繁项集。

（2）算法具体过程见书上所述。一个例子：

TID	*Items bought*	*(ordered) frequent items*
100	{*f, a, c, d, g, i, m, p*}	{*f, c, a, m, p*}
200	{*a, b, c, f, l, m, o*}	{*f, c, a, b, m*}
300	{*b, f, h, j, o, w*}	{*f, b*}
400	{*b, c, k, s, p*}	{*c, b, p*}
500	{*a, f, c, e, l, p, m, n*}	{*f, c, a, m, p*}

min_support = 3



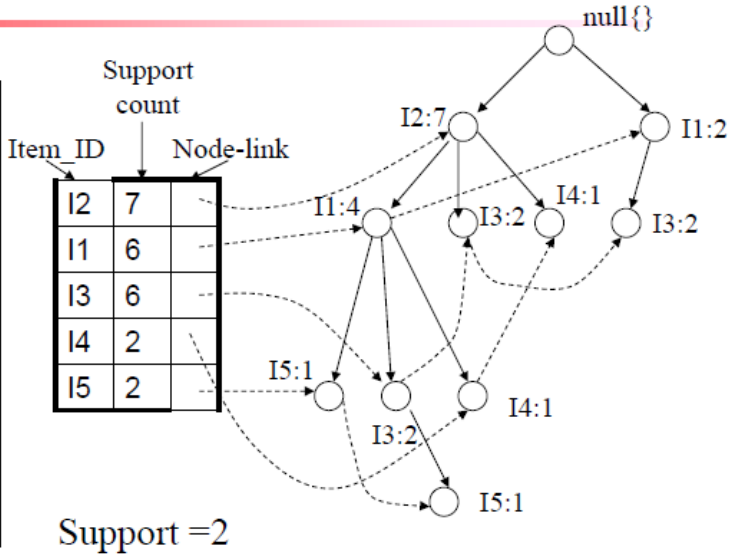
（3）一个完整的例题：

3. A database has 9 transactions. Let *min_sup* = 20%. Please construct the FP-tree for the database, the conditional FP-trees, and all the frequent itemsets.

TID	List of items_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

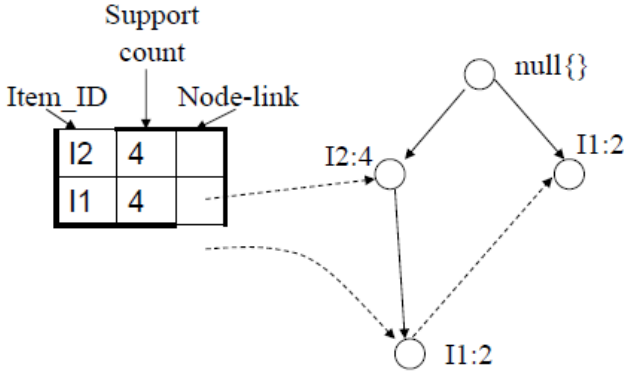
2014/10/6

TID	List of items_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3



Support =2

item	*conditional pattern base*	*conditional FP-tree*	*frequent patterns generated*
I5	{{I2,I1: 1}, {I2,I1,I3: 1}}	⟨I2: 2, I1: 2⟩	{I2,I5: 2}, {I1,I5: 2}, {I2,I1,I5: 2}
I4	{{I2,I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2,I4: 2}
I3	{{I2,I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2,I3: 4}, {I1,I3: 4}, {I2,I1,I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2,I1: 4}



27.挖掘多层关联规则：高层较抽象，希望下钻得到更准确的描述（P182-184）。
（1）一致支持度方法：对于所有层使用一致的最小支持度（一致支持度）；
（2）递减支持度方法：在较低层使用递减的最小支持度（递减支持度）；
（3）使用基于项或基于分组的最小支持度（基于分组的支持度）；
（4）自顶自下执行 Apriori 算法。如果祖先是不频繁的，那么对于祖先的分支下的搜索停止。
28.挖掘多维关联规则：涉及到两个或者以上维或谓词的关联规则（P186）。如 age(X,"19-25") ∧ occupation(X,"student") ∧ buys(X, "coke").
（1）用概念分层模型；
（2）数值数据被范围代替；
（3）数据立方体被证明是有效的方法。
挖掘量化关联规则：（P186-187）
（1）数值数据被动态离散化；
（2）2D 量化关联规则：$A_{quan1} \wedge A_{quan2} \Rightarrow A_{cat}$；

第四章 分类（Classification.pdf，涉及书上第 8、9 章）

29.分类的一些基本概念（P211-213）
（1）数值预测、预测器、回归分析、分类器；学习阶段、训练集、监督学习（分类）、无监督学习（聚类）、过拟合、校验集、准确率。
（2）分类有两大步骤：建立模型和使用模型来预测。
30.分类的一些问题：
（1）数据准备：数据清理（消除噪声、处理缺失值）、特征选择（消除不相关或冗余属性）、数据变换（数据正则化）；
（2）评估分类方法：准确率（分类器和预测器）、速度（建立模型的时间、使用模型的时间）、鲁棒性（处理噪声和缺失值）、扩展性、可解释性等。
30.决策树归纳（P213 起）：
（1）一些基本概念（P213、P214-215）：决策树、分枝、结点、根、分裂属性、分裂点、分裂子集等。
（2）算法的基本思想是贪心，以自顶向下的方式（递归，非回溯）构造。详细见 P214-215.
（3）分裂准则划分的元组有三种可能性：离散值（结点的输出直接对应类别）、连续值（找到分裂点，大于小于关系来分枝）、离散值且必须产生二叉树（用集合的思想来判定是否属于）。
31.属性选择度量（P217 起）：
（1）信息增益（Information Gain）：ID3、C4.5 使用，增益越大越优先选择。对于一个集合 D，有 m 类，p_i 是第 i 类的概率

首先计算对 D 中元组进行分类的期望信息：$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$ ，

然后假定要计算属性 A 的增益，其中 A 属性有 v 个不同的值 $\{a_1, a_2, \ldots, a_v\}$，用 A 将 D 划分为 v 个分区 $\{D_1, D_2, \ldots, D_v\}$。则再计算下式：

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

最后，属性 A 的信息增益为：$Gain(A) = Info(D) - Info_A(D)$.

这种定义是按照书上来的，例子见书上所述。还有一种定义是 ppt 里的：

- Assume that using attribute A have v distinct values, $\{a_1, a_2, \ldots, a_v\}$
- Training set S will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$
  - If S_i contains p_i examples of P and n_i examples of N, the entropy, or the expected information based on the partitioning into subsets by attribute A is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

- Information gain of A

$$Gain(A) = I(p, n) - E(A)$$

一个例子：

- 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	$I(p_i, n_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

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

Hence

$$Gain(age) = I(p, n) - E(age) = 0.246$$

22

对于连续值，需要将各个属性值先排序，然后针对相邻两个值计算中点，对每一个中点计算信息增益，从而选择最好的分裂点。

（2）增益率（Gain Ratio）：C4.5 使用，越大越优先使用。对属性 A 计算分裂信息：

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

则增益率为 $GainRate(A) = \frac{Gain(A)}{SplitInfo_A(D)}$ 。

（3）基尼指数（Gini Index）：在 CART 中使用，值越小越优先使用。计算方式：

首先计算不纯度 $Gini(D)=1-\sum_{i=1}^{m} p_i^{\,2}$ ，

如果 D 被属性 A 分成两个子集 D_1 和 D_2，则 D 的基尼指数为：

$$Gini_A(D)=\frac{|D_1|}{|D|}\times Gini(D_1)+\frac{|D_2|}{|D|}\times Gini(D_2)$$

32.剪枝（pruning,P223）：

（1）先剪枝（prepruning）：提前停止树的构建（在给定的结点不再分裂或划分训练元组的子集）。

（2）后剪枝（postpruning）：由完全生长的枝剪去子树。

（3）还有悲观剪枝等。

33.贝叶斯分类（Bayesian），朴素贝叶斯（naïve Bayesian）（P227）：

（1）原理：基于贝叶斯公式 $P(H\mid X)=\frac{P(X\mid H)P(H)}{P(X)}$ 。基于这样的假设：给定的类之间是相互独立的。

（2）朴素贝叶斯分类器的具体过程和公式都是基于贝叶斯定理，见书上所述。遇到 0 概率时用拉普拉斯平滑。属性值是分类属性和连续属性时有不同的概率处理方法。

34.后向传播分类（Backpropagation，P258 起）：

（1）一种神经网络（neural network）方法。基本概念：输入层、输出层、隐藏层、输入单元、输出单元、神经节点、前馈网络等。

（2）定义网络拓扑：

输入输出个数，隐藏层数，隐藏层的节点数等；

规范化每个属性值到 0 和 1 之间；

一个输入单元一个值，通常初始化为 0；

分类时多于两类则一个输出节点表示一个值，两类则一个节点表示；

反复构建神经网络直到准确率可以被接受。

（3）构建算法见 P260，概述：初始化权值为随机数，计算隐藏层和输出层每个单元的净输入和输出，训练样本提供给网络的输入层，隐藏层输出层每个单元的净输入用其输入的线性组合计算。

（4）一些值的计算：

a.给出层和输出层的单元 j，到单元 j 的净输入 I_j 为 $I_j=\sum_i w_{ij}O_i+\theta_j$ ，wij 是上一层单元 i 到 j 的连接权重，O_i 是上一层的单元 i 的输出，θ 是单元 j 的偏倚（bias）。

b.隐藏层和输入层的每个单元取其净输入，然后将激活函数作用于它，使用 logistic 或者 sigmoid 函数。给定单元 j 的净 I_j，则单元 j 的输出 O_j 为 $O_j=\frac{1}{1+e^{-I_j}}$ 。又叫挤压函数。

c.对于输出层单元 j，误差 Err_j 为 $Err_j=O_j(1-O_j)(T_j-O_j)$ ，T_j 是 j 给定训练元组的已知目标值。

d.对于隐藏层单元 j 的误差，考虑下一层中连接 j 的单元的误差加权和，计算方法：

$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$ ，w_{jk} 是下一较高层中单元 k 到单元 j 的连接权重，Err_k 是单元 k 的误差。

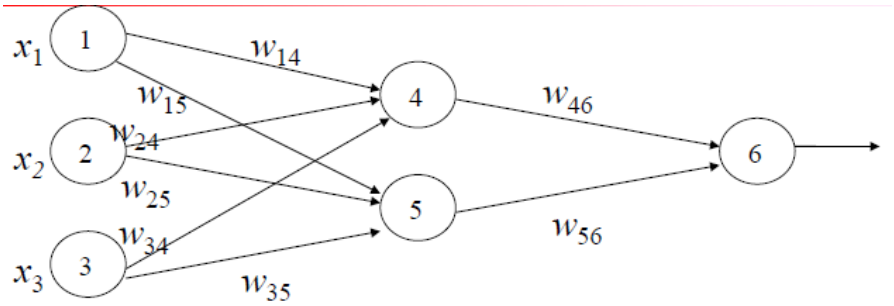
e.更新权重和偏倚， Δw_{ij} 是权 w_{ij} 的改变量，l 是学习率。

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

偏倚由下式更新，其中 $\Delta\theta_j$ 是偏倚 θ_j 的改变量：

$$\Delta\theta_j = (l) Err_j$$
$$\theta_j = \theta_j + \Delta\theta_j$$

（5）一个例子：



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

Unit j	Net input, I_j	Output, O_j
4	$0.2 + 0 - 0.5 - 0.4 = -0.7$	$1/(1 + e^{0.7}) = 0.332$
5	$-0.3 + 0 + 0.2 + 0.2 = 0.1$	$1/(1 + e^{-0.1}) = 0.525$
6	$(-0.3)(0.332) - (0.2)(0.525) + 0.1 = -0.105$	$1/(1 + e^{0.105}) = 0.474$

Unit 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	$(0.332)(1 - 0.332)(0.1311)(-0.3) = -0.0087$

Weight or bias	New value
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	$-0.4 + (0.9)(-0.0087) = -0.408$

（5）缺点：训练时间长，需要经验化大量的参数（拓扑结构），可解释性弱。

优点：对噪声效果好，对连续值输入输出效果好等。

35.其他分类方法：

（1）*K* 近邻（k-nearest neighbor, *k*-NN，P275-276）：用欧几里得距离度量两个点之间的距离。

（2）组合分类方法（Ensemble Methods）：由多个分类器组成，基于投票机制返回效果最好的那个分类器。常用方法有装袋（Bagging，投票机制）、提升（Boosting，动态更新权值）、随机森林。（P245-249）

36.预测：通常采用回归模型，不同于分类模型。

（1）线性回归：

Linear regression: a response variable y and a single predictor variable x

$y = w_0 + w_1 x$

where w_0 (intercept) and w_1 (slope) are regression coefficient

Method of least squares: estimates the best-fitting straight line

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2} \qquad w_0 = \bar{y} - w_1 \bar{x}$$

Multiple linear regression: more than one predictor variable

- Training data is of the form $(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_{|D|}, y_{|D|})$
- Ex. For 2-D data, we may have: $y = w_0 + w_1 x_1 + w_2 x_2$
- Solvable by extension of least square method or using SAS, S-Plus, R, Matlab
- Many nonlinear functions can be transformed into the above

4/10/26 69

（2）非线性回归：

A polynomial regression model can be transformed into linear regression model. For example,

$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$

convertible to linear with new variables: $x_2 = x^2$, $x_3 = x^3$

$y = w_0 + w_1 x + w_2 x_2 + w_3 x_3$

Some models are intractable nonlinear (e.g., sum of exponential terms)

- possible to obtain least square estimates through extensive calculation on more complex formulae

（3）其他回归：

Generalized linear model:

- Foundation on which linear regression can be applied to modeling categorical response variables
- Logistic regression: models the prob. of some event occurring as a linear function of a set of predictor variables

  $\text{Log}(p/1\text{-}p) = W_0 + W_1 X + W_2 X_2 + \ldots + W_3 X_3$, p is probability Y=1
- Poisson regression: models the data that exhibit a Poisson distribution

Log-linear models: (for categorical data)

- Approximate discrete multidimensional prob. distributions
- Also useful for data compression and smoothing

$$\log(y) = W_0 + W_1 X + W_2 X_2 + \ldots + W_n X_n$$

第五章 聚类（clustering.pdf，涉及书上第 2、10、12 章）

37.对聚类分析的要求（P289-291）.

38.数据类型（P45 起）：

（1）数据矩阵；

（2）相异性矩阵（Dissimilarity Matrix）：又叫对象-对象结构：存放 *n* 个对象两两之间的邻近度，通常用一个 *n***n* 矩阵表示：

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \ldots & \ldots & 0 \end{bmatrix}$$

（3）区间标度变量（Interval-valued variables）：计算平均绝对差值：

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

$$\text{where} \quad m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf})$$

或者计算标准度量（z-score）：

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

（4）闵诺夫斯基距离（Minkowski Distance）：

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

当 q=1 时就变成曼哈顿距离（Manhattan Distance）：

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

当 q=2 时就变成欧几里得距离（Euclidean Distance）：

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

（5）二值属性邻近度度量（Binary Variables）：根据列联表（Contingency Table）计算，有对称和非对称的二元相异性：

列联表：

		Object j		
		1	0	*sum*
Object i	1	a	b	$a+b$
	0	c	d	$c+d$
	sum	$a+c$	$b+d$	p

对称：$d(i,j)=\frac{b+c}{a+b+c+d}$，非对称：$d(i,j)=\frac{b+c}{a+b+c}$。

非对称的二元相似性（Jaccard 系数）：$sim(i,j)=\frac{q}{q+r+s}=1-d(i,j)$

（6）标称属性邻近度度量（Nominal Variables）：m 是匹配的数目，p 是刻画对象的属性总数，则邻近度为：$d(i,j)=\frac{p-m}{p}$。

（7）序数属性相异性（Ordinal Variables）：用排序 r_{if} 取代 x_{if}，进行规格化：$z_{if}=\frac{r_{if}-1}{M_f-1}$，属性有 M_f 个有序的状态。

（8）比例标度变量（ratio-scaled variables）：或者像区间标度数据一样（不好），或者用 log 函数 $y_{if}=\log(x_{if})$；或者把它们当作连续序数或者它们的序当作比例标度。

（9）混合属性相异性（Mixed-Types）：

One may use a weighted formula to combine their effects $d(i,j)=\frac{\sum_{f=1}^{p}\delta_{ij}^{(f)}d_{ij}^{(f)}}{\sum_{f=1}^{p}\delta_{ij}^{(f)}}$

- $\delta^{(f)}_{ij}$ = 0 if x_{if} or x_{jf} is missing, or x_{if} = x_{jf} =0 and f is asymmetric attribute; otherwise, $\delta^{(f)}_{ij}$ = 1
- *f* is binary or nominal:
  $d_{ij}^{(f)}$ = 0 if x_{if} = x_{jf} , or $d_{ij}^{(f)}$ = 1 otherwise
- *f* is interval-based: use the normalized distance
- *f* is ordinal
  - compute ranks r_{if} and treat z_{if} as interval-scaled $z_{if}=\frac{r_{if}-1}{M_f-1}$
- *f* is ratio-scaled
  - Transform *f*, and treat *f* as interval-scaled. y_{if} = *log*(x_{if})

39.划分方法（P293 起）：常用方法：*k*-means, *k*-medoids, CLARANS。

k-均值（*k*-Means）：

（1）复杂度 $O(tkn)$，n 是物体的数量，k 是簇的数量，t 是迭代的次数。

（2）具体方法：

随机选择 k 个对象，每个对象初始地代表一个簇的平均值或中心。

对剩余的每个对象，根据其与各个簇中心的距离，将它赋给最近的簇。

重新计算每个簇的平均值，并用该平均值代表相应的簇；

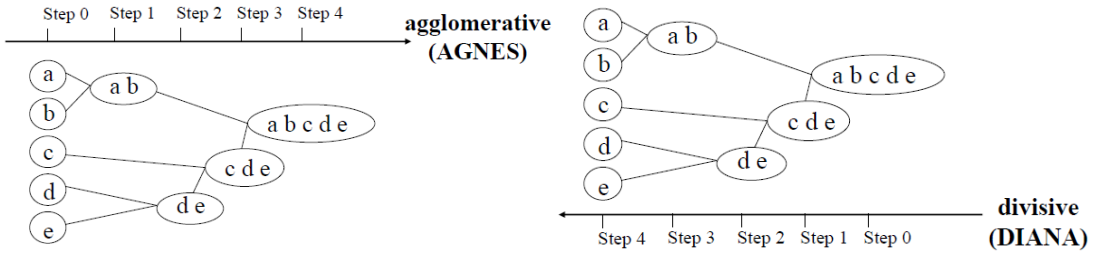
将每个对象根据其与各个簇中心的距离，重新分配到与它最近的簇中；

回到第三步，直到不再有新的分配发生。

（3）缺点：对离群点敏感，此时要用 *k*-中心点（*k*-medoids）。对分类数据时用 *k*-众数代替。

40.层次方法（Hierarchical Clustering）：采用距离作为衡量聚类的标准。该方法不再需要指定聚类的个数，但用户可以指定希望得到的簇的数目作为一个结束条件。代表方法：BIRCH，ROCK，变色龙（CHAMELEON）。

（1）AGNES（Agglomerative NESting）是凝聚方法，DIANA（Divisive ANAlysis）是分裂方法，二者相反。复杂度 $O(n^2)$，n 是物体的数量。



（2）BIRCH（P301）：基于聚类特征树的多阶段聚类。构造一棵 *CF*-tree（clustering feature）。类的 *CF* 是一个三维向量：*CF*=<*n*,*LS*,*SS*>，*LS* 是 n 个点的线性和，*SS* 是数据点的平方和。复杂度 $O(n)$。

（3）CURE：复杂度 $O(n^2\log n)$基本算法：

开始时每个点都是单独的一个簇

归并最近的簇直到每个簇包含超过 c 个点

对于每个簇，用 c 个散点作为代表

如果有超过 k 个簇，最近的距离的簇要合并，更新合并簇的代表点。

选择代表点和归并的方法：

Choose representatives

- the point farthest from the mean of the cluster
- for 2 to *c* do
  the point farthest from the previously chosen point
- Shrink the scattered points toward the mean by a fraction α
  for each scattered point *p* do
  representative = $p + \alpha$ * (mean – *p*)

Merge

- Euclidian distance

$$dist(u,v) = \min_{p \in u.\text{rep}, q \in v.\text{rep}} dist(p,q)$$

- Closest clusters –- minimum distance between representative points from two clusters

41.密度方法（Density based methods，P306）：复杂度 $O(n^2)$.常用方法：DBSCAN, OPTICS, DenClue。

（1）一些概念：

ε-neighborhood: neighborhood within a radius ε of a point

MinPts: Min number of points in ε-neighborhood of a point

core object: If the number of points in ε-neighborhood of point p exceeds MinPts

直接密度可达（directly density reachable）：p 对 q 可达，如果 p 是 q 的近邻或者 q 是 core object。

密度可达（density reachable）：有链条关系到达。

密度相连（density connected）：如果 p 和 q 都对 o 密度可达。

（2）DBSCAN 算法：将具有足够高密度的区域划分为簇，并可以在带有“噪声”的空间数据库中发现任意形状的聚类。算法过程：

任意选择没有加簇标签的点 p

找到从 p 关于 ε 和 MinPts 密度可达的所有点

如果$|N_\varepsilon(q)|>=MinPts$ ，则 p 是核心对象,形成一个新的簇，给簇内所有的点加簇标签

如果 p 是边界点，则处理数据库的下一点

重复上述过程，直到所有的点处理完毕

42.网格方法（Grid-Based Method，P312）：基本有 STING，CLIQUE，WaveCluster 等。

STING：空间区域被分成矩形单元，每个高层单元可以被分为多个低层单元。需要 min，max，s，count，分布信息等。复杂度：查询时间 $O(k)$，k 是最底层的单元数；生成簇时间 $O(n)$。不足之处是，所有的簇边界都得是水平或垂直，处理时间取决于每个网格的尺寸。

43.离群点检测（Outlier Discovery，P356 起）：常用方法

（1）统计方法：为数据点设想满足一个分布，然后发现边界。不足：一般只能单个属性有用，并且数据分布通常是不知道的。

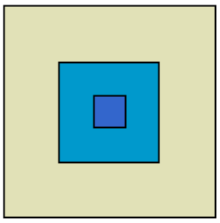
（2）基于距离：

a. Index-Based 方法，找一个物体周围 d 半径范围的点。复杂度是 $O(kn^2)$。

b. Cell-Based 方法:

Cell partition

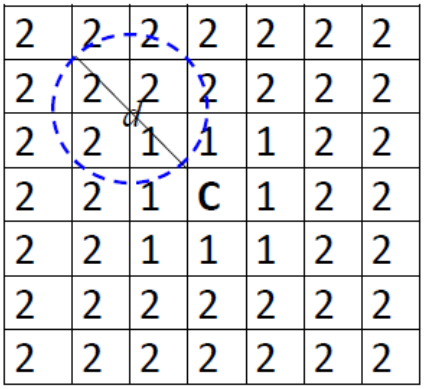
- Partition data space into cells, side length $d/2k^{1/2}$
- Each cell has two layers around it
  - First layer one cell thick
  - Second layer $(2k^{1/2}-1)$ cells thick



Outlier detection

- If count of the first layer >M, no outlier in this cell
- If count of the second layer<=M, all objects are outliers
- Otherwise, examine every object in the cell

Good for large-scale data set



k=2, *k* dimensionality

Layer-1 property: given any point *x* in cell *C*, and any point *y* in layer-1 cell, $dist(x,y) \leq d$

Layer-2 property: given any point *x* in cell *C*, and any point *y* out of layer-2 cell, $dist(x,y) > d$

（3）基于邻近的方法：通过检查一个组内的主要数据特征来判断离群点。

一个练习答案：

Suppose that a large store has a transaction database that is *distributed* among four locations. Transactions in each component database have the same format, namely $T_j : \{i_1, \ldots, i_m\}$, where T_j is a transaction identifier, and i_k $(1 \leq k \leq m)$ is the identifier of an item purchased in the transaction. Propose an efficient algorithm to mine global association rules (without considering multilevel associations). You may present your algorithm in the form of an outline. Your algorithm should not require shipping all of the data to one site and should not cause excessive network communication overhead.

Answer:

An algorithm to mine global association rules is as follows:

- Find the local frequent itemsets in each store. Let CF be the union of all of the local frequent itemsets in the four stores.
- In each store, find the local (absolute) support for each itemset in CF.
- Now we must determine the global (absolute) support for each itemset in CF. This can be done by summing up, for each itemset, the local support of that itemset in the four stores. Doing this for each itemset in CF will give us their global supports. Itemsets whose global supports pass the support threshold are global frequent itemsets.
- Derive strong association rules from the global frequent itemsets.