第十八章：

数据挖掘

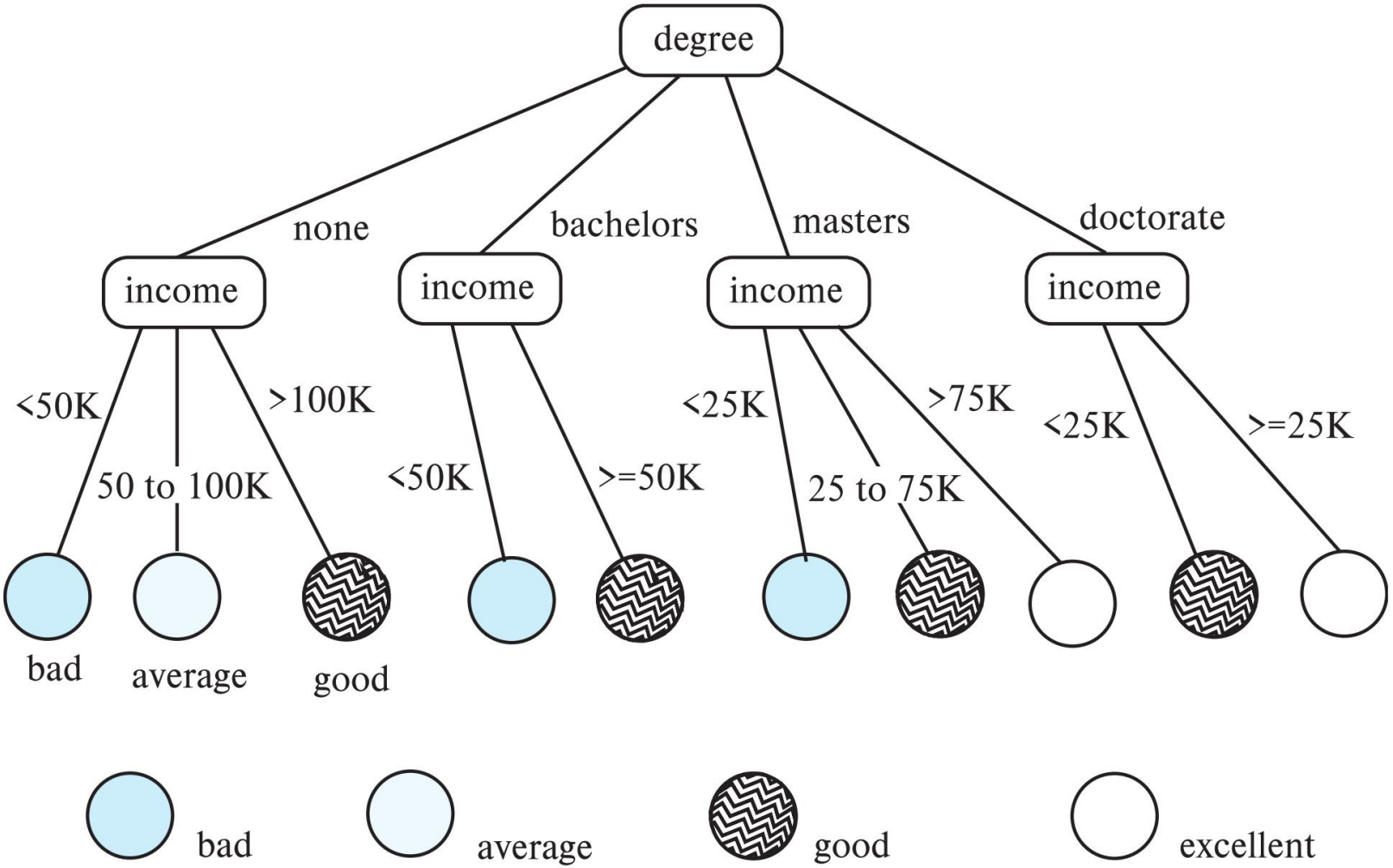
数据挖掘

- 数据挖掘是半自动分析大型数据库以找到有用模式的过程
  - 与机器学习类似的目标,但数据量非常大
- 也称为数据库中的知识发现(KDD)
- 某些类型的知识可以表示为规则
- 更一般地说,知识是通过对过去的数据实例应用机器学习技术来发现的,从而形成一个模型
  - 然后使用模型对新实例进行预测

数据挖掘任务的类型

- 数据挖掘任务示例：
  - 分类
    - 项目（具有相关属性）属于多个类别之一
    - 训练实例具有提供的属性值和类
    - 给定一个类别未知的新项目，预测它属于哪个类别
      属于基于其属性值
  - 协会
    - 查找“相似”客户经常购买的书籍。如果一个新的这样的客户买了一本这样的书，也推荐其他的。
    - 关联可用作检测因果关系的第一步
      - 例如，接触化学物质 X 与癌症之间的关联
  - 聚类
    - 例如，伤寒病例聚集在污染井周围的区域
    - 集群检测对于发现流行病仍然很重要

决策树分类器 Decision Tree Classifiers



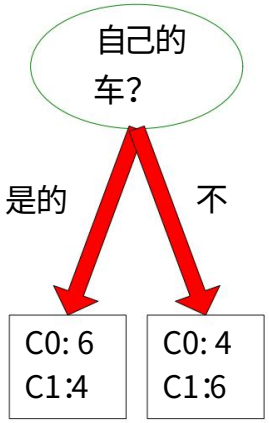
决策树 sion Trees

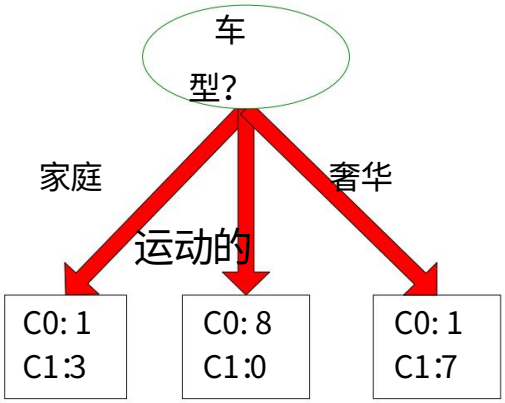
- 树的每个内部节点根据分区或拆分属性以及节点的分区条件将数据划分为组

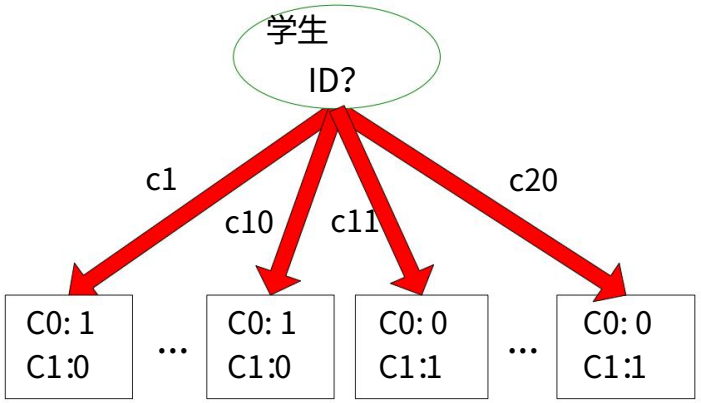
- 节点纯度：
  - 节点上的所有（或大部分）项目属于同一类
- 从顶部遍历树进行预测

分裂

拆分前:0类（C0)的10条记录，
10 类 1 (C1) 记录







哪个最适合预测信用价值?

如何确定一个好的拆分 Good Split

- 贪婪的方法：
  - 具有均匀或纯类分布的节点是首选
- 需要测量节点杂质：

C0: 5 C1:5	C0: 9 C1:1
非均质 杂质度高	同质 杂质度低

杂质测量:GINI

- 给定节点t 的基尼指数，

$$= 1 - \#^{-1}_{=0} ( )^2$$

其中是类在节点总类数的相对频率 ( ) ，并且是

- 对于 2 类问题(p, q),其中p表示相对
  第 1 类的频率,并且q = (1-p)
  - 基尼系数 = 1 - p^2 – $(1 – p)^2$ = 2p (1-p) = 2pq
- 最大值 (1 – 1/c)当记录在所有类别中平均分布时,这意味着分类最不利的情况

- 当所有记录都属于一个类时,最小值 (0),表示大多数
  分类的有利情况

C1	0
C2	6
基尼=0.000	

C1	1
C2	5
基尼=0.278	

C1	2
C2	4
基尼=0.444	

C1	3
C2	3
基尼=0.500	

计算单个节点的基尼指数 Computing Gini of a Single Node

$$= 1 - \sum_{=0}^{-1} \#\ (\ )^2$$

C1	0
C2	6

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

基尼系数 = 1 – P(C1)2 – P(C2)2 = 1 – 0 – 1 = 0

C1	1
C2	5

P(C1) = 1/6　　P(C2) = 5/6

基尼系数 = 1 – (1/6)2 – (5/6)2 = 0.278

C1	2
C2	4

P(C1) = 2/6　　P(C2) = 4/6

基尼系数 = 1 – (2/6)2 – (4/6)2 = 0.444

基于 GINI 的拆分

- 当一个节点p被分裂成k个分区（子)时,分裂的质量是
  计算为，

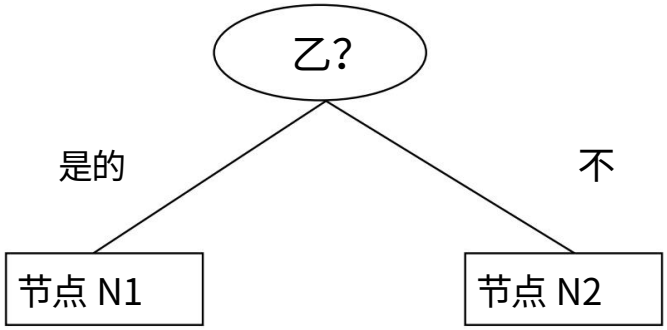
$$\text{基尼分}_{\text{裂}} = \sum_{i=1}^{k} \frac{n_i}{n} \text{基尼指数}(i)$$

在哪里，　ni = 子i 处的记录数， n = 节点p 处的记录数。

计算基尼指数

- 分成两个分区
- 称重分区的影响：
  - 寻求更大、更纯净的分区　降低基尼系数

	家长
C1	6
C2	6
基尼 = 0.500	



C1:5; C2:2

C1:1; C2:4

基尼系数
(N1) = 1 – (5/7)2 – (2/7)2
= 1-0.51 – 0.082

= 0.408

基尼系数（儿童）= 7/12 * 0.408 + 5/12 * 0.32

= 0.238 + 0.133

= 0.371（从 0.5 减少）

基尼系数
(N2) = 1 – (1/5)2 – (4/5)2
= 1 – 0.04 – 0.64

= 0.32

	N1	N2
C1	5	1
C2	2	4
基尼=0.371		

基于熵的分割准则 Based on Entropy

- 给定节点t 的熵，

$$= -\sum_{=0}^{-1} ( ) \quad 2 ( )$$

班级数量 ( ) 是节点处类的相对频率 ，并且是总数

- 衡量节点缺乏同质性（或无序性）
  - 当记录平均分布在所有类别中时的最大值 (log2c) ,这意味着对分类最不利的情况
  - 最小值 (0),当所有记录都属于一个类时,意味着最多分类的有利情况
- 从logba*logax = logbx,我们有 log2x = log2e*logex = 1.44 logex
- 基于熵的计算类似于基尼指数计算

计算熵 Computing Entropy

$$= \sum_{=0}^{-1} ( ) \quad 2 ( )$$

注意 log2x = log2e*logex = 1.44 logex
P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

熵 = – 0 log20 – 1 log21 = – 0 – 0 = 0

C1	0
C2	6

P(C1) = 1/6 P(C2) = 5/6

熵 = – (1/6) log2 (1/6) – (5/6) log2 (5/6) = 0.65

C1	1
C2	5

P(C1) = 2/6 P(C2) = 4/6

熵 = – (2/6) log2 (2/6) – (4/6) log2 (4/6) = 0.92

C1	2
C2	4

杂质测量:错误分类错误 Misclassification Error

- 节点的分类错误

$$( ) = 1 - \text{最大} [\quad ( )]$$

其中最 ( ) 是节点处类的相对频率 ，和
大值被所有类c

- 我们希望 maxi [pi (t)] 大,因此 1- maxi [pi (t)] 小
- 当记录在所有类别中平均分布时,最大值为 (1 – 1/c) ,这意味着分类最不利的情况
- 当所有记录都属于一个类时,最小值为 0,暗示最有利于分类的情况

单节点计算误差

Computing Error of a Single Node

( ) = 1 - 最大 [( )]

C1	0
C2	6

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

误差 = 1 – 最大值 (0, 1) = 1 – 1 = 0

C1	1
C2	5

P(C1) = 1/6 P(C2) = 5/6

误差 = 1 – 最大值 (1/6, 5/6) = 1 – 5/6 = 1/6

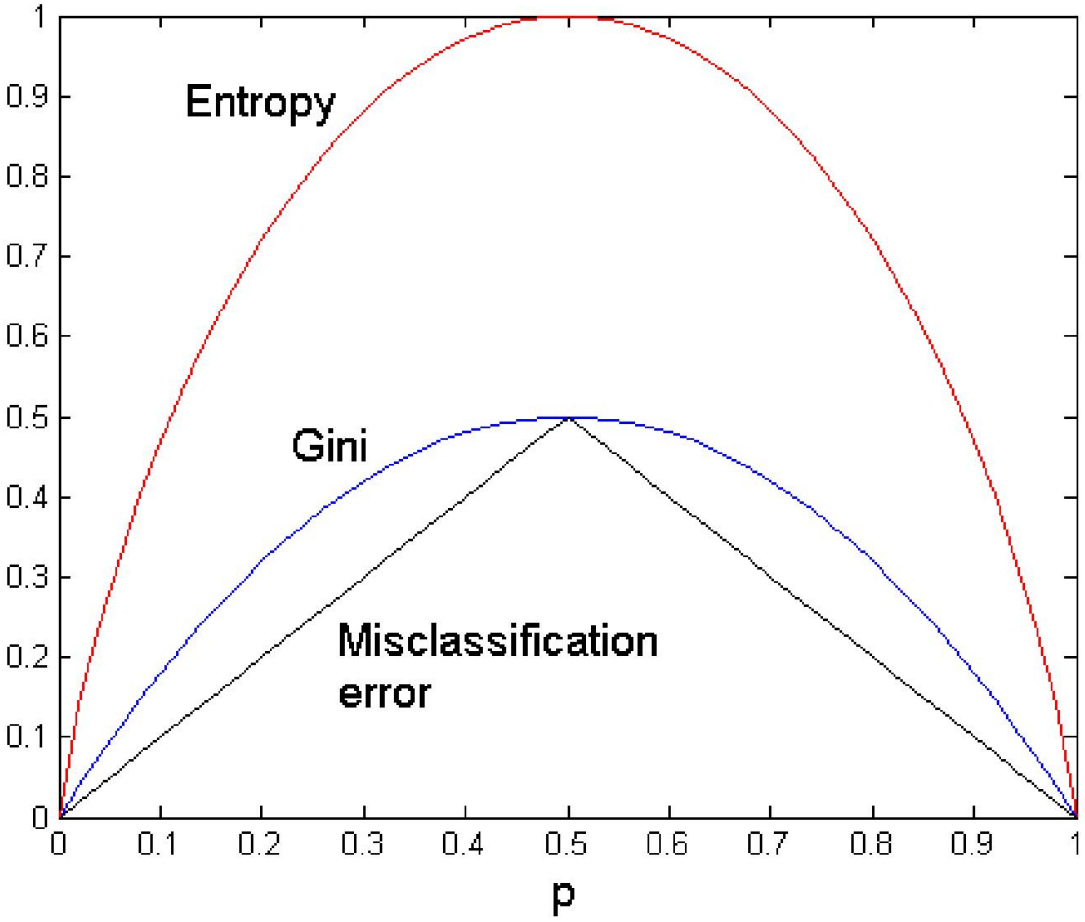
C1	2
C2	4

P(C1) = 2/6 P(C2) = 4/6

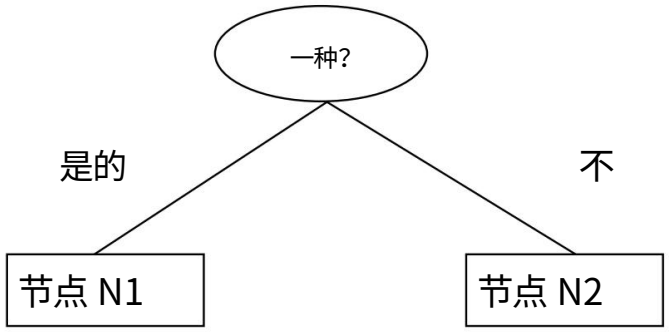
误差 = 1 – 最大值 (2/6, 4/6) = 1 – 4/6 = 1/3

杂质指标比较 among Impurity Measures

对于 2 类问题， p给出 1 类的相对频率



错误分类误差与基尼指数 Error vs Gini Index



	家长
C1	7
C2	3
基尼 = 0.42	

基尼系数
(N1) = 1 – (3/3)2 – (0/3)2
= 0

基尼系数
(N2) = 1 – (4/7) 2 – (3/7) 2
= 0.489

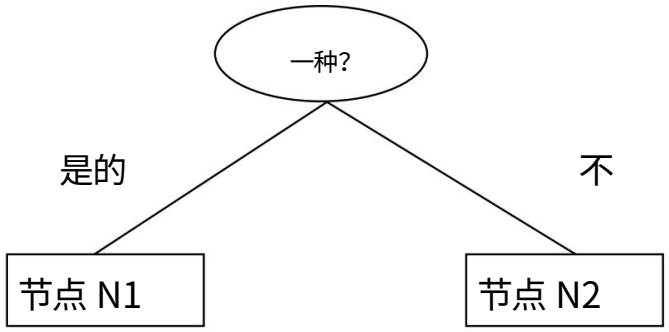
	N1	N2
C1	3	4
C2	0	3
基尼=0.342		

基尼系数（儿童）
= 3/10 * 0 + 7/10 * 0.489

= 0.342

基尼系数有所改善,但错误呢?

错误分类误差与基尼指数 Error vs Gini Index



	家长
C1	7
C2	3

误差 = 0.3

误差(N1) =
1 – 最大值[(3/3), (0/3)]
= 0

误差(N2) =
1 – 最大值[(4/7), (3/7)]
= 3/7

	N1 N2	
C1	3	4
C2 0		3

误差 = 0.3

误差（父） = 1 –
最大值[(7/10), (3/10)] = 0.3

误差（儿童） =
3/10 * 0 + 7/10 *
= 0.3

基尼系数有所改善,但错误仍然相同！！

贝叶斯分类器

▪贝叶斯分类器使用贝叶斯定理，

$$p(cj \mid d) = p(d \mid cj) * p(cj) / p(d)$$

在哪里

p(cj |d) = 实例d在类cj中的概率， p(d|cj) =给定类cj生成实例d的概率， p(cj) = 类cj出现的概率， p(d) = 概率实例d发生和

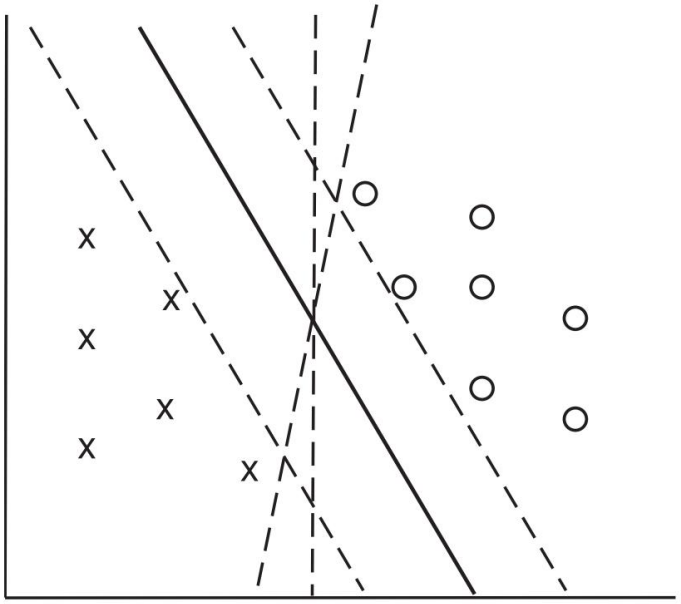
▪例如： d可能代表个人,类别可能是“买电脑”和“不买电脑” d = （年龄=青年,收入=中等,学生=是，信用评级=公平）

▪为了简化任务,朴素贝叶斯分类器假设属性具有独立分布,从而估计

$$p(d \mid cj) = p(d1 \mid cj) * p(d2 \mid cj) * \ldots * p(dn \mid cj)$$

支持向量机分类器 Support Vector Machine Classifiers

- 简单的二维示例：
  - 分数分为两类
  - Find a line (maximum margin line) st line 分为两类,与任一类中最近点的距离最大



支持向量机 Support Vector Machine

- 在n 维中,点被平面而不是线分割
- SVM 可以用作曲线的分隔符,不一定是线性的,由
  分类前的变换点
  - 变换函数可能是非线性的,称为核函数
    职能
  - 分隔符是变换空间中的平面,但映射到原始空间中的曲线
- 给定的一组点可能没有精确的平面分隔符
  - 选择最能分离点的平面

关联规则iation Rules

- 给定一组交易,找到将根据交易中其他项目的出现来预测项目出现的规则

- 零售商店通常对人们购买的不同商品之间的关联感兴趣。
  - 买面包的人很可能也买牛奶
  - 购买了《数据库系统概念》一书的人很可能还购买了《操作系统概念》一书

- 关联信息可以多种方式使用。
  - 例如,当客户购买特定书籍时,在线商店可能会推荐相关书籍

- 关联规则:

  面包→牛奶　　DB-Concepts, OS-Concepts →网络

  - 左侧:前件,右侧:后件
  - 关联规则必须有关联的总体;这人口由一组实例组成
    - 例如,商店的每笔交易（销售)都是一个实例,所有交易的集合就是人口

频繁项集

物品集

- 一个或多个项目的集合

  示例:{牛奶}、{牛奶、面包、尿布}

- k 项集

  包含k个项目的项目集

支持计数( )或绝对支持

- 项集的出现频率
- 例如 ({牛奶、面包、尿布}) = 2

支持或相关支持

- 包含项目集的事务的一部分
- s = / |T|,其中|T|是交易数量
- 例如 s({面包、牛奶、尿布}) = 2/5

频繁（或大)项集

- 支持度大于或
  等于minsup阈值,其中minsup是给定的最小支持度

时间项目	
1	面包、牛奶
2	面包、尿布、啤酒、鸡蛋
3	牛奶、尿布、啤酒、可乐
4	面包、牛奶、尿布、啤酒
5	面包、牛奶、尿布、可乐

时间项目	
1	面包、牛奶
2	面包、尿布、啤酒、鸡蛋
3	牛奶、尿布、啤酒、可乐
4	面包、牛奶、尿布、啤酒
5	面包、牛奶、尿布、可乐

关联规则iation Rules

关联规则

– X 形式的隐含表达式 →
Y,其中 X 和 Y 是项集

–箭头表示同时发生,而不是因果关系

–示例:
{牛奶、尿布} → {啤酒}

时间项目	
1	面包、牛奶
2	面包、尿布、啤酒、鸡蛋
3	牛奶、尿布、啤酒、可乐
4	面包、牛奶、尿布、啤酒
5	面包、牛奶、尿布、可乐

规则评估指标

–关联规则的支持
·交易中同时包含 X 的部分
和 Y

–关联规则的置信度 (c)
·衡量 Y 中的项目在交易中出现的频
率
包含 X

例子:

{牛奶、尿布 →}{啤酒}

矿业协会规则

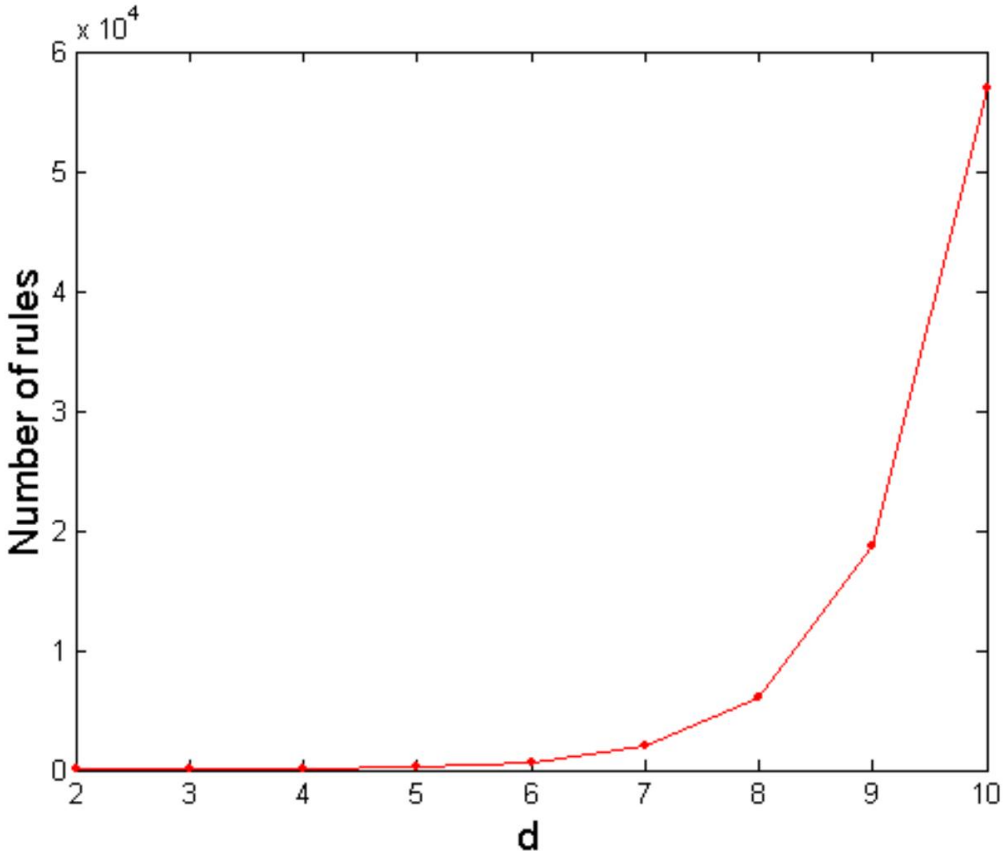
- 给定一组交易T,关联规则挖掘的目标是找到所有有规则
  - Support (s) ⩾ minsup阈值
  - 置信度(c) ⩾ minconf阈值

- 蛮力方法：
  - 列出所有可能的关联规则
  - 计算每个规则的支持度和置信度
  - 修剪未达到minsup和minconf阈值的规则

  计算上令人望而却步！

计算复杂度

- 给定d个唯一项：
  - 项集总数 = 2d
  - 可能的关联规则总数：



$$R = \sum_{k=1}^{d-1} \left[\binom{d}{k} \times \binom{d-k}{j} \right]$$

$$= 3^d - 2^{d+1} + 1$$

如果d=6， R = 602 规则

计算复杂度（证明）

Suppose there are d items. We first choose k of the items to form the left-hand side of the rule. There are $\binom{d}{k}$ ways for doing this. After selecting the items for the left-hand side, there are $\binom{d-k}{i}$ ways to choose the remaining items to form the right hand side of the rule, where $1 \leq i \leq d-k$. Therefore the total number of rules (R) is:

$$
\begin{aligned}
R &= \sum_{k=1}^{d} \binom{d}{k} \sum_{i=1}^{d-k} \binom{d-k}{i} \\
&= \sum_{k=1}^{d} \binom{d}{k} \left(2^{d-k} - 1\right) \\
&= \sum_{k=1}^{d} \binom{d}{k} 2^{d-k} - \sum_{k=1}^{d} \binom{d}{k} \\
&= \sum_{k=1}^{d} \binom{d}{k} 2^{d-k} - \left[2^{d} + 1\right],
\end{aligned}
$$

where

$$
\sum_{i=1}^{n} \binom{n}{i} = 2^{n} - 1.
$$

计算复杂度（证明继续）

Since

$$(1+x)^d = \sum_{i=1}^{d} \binom{d}{i} x^{d-i} + x^d,$$

substituting $x = 2$ leads to:

$$3^d = \sum_{i=1}^{d} \binom{d}{i} 2^{d-i} + 2^d.$$

Therefore, the total number of rules is:

$$R = 3^d - 2^d - \left[2^d + 1\right] = 3^d - 2^{d+1} + 1.$$

计算复杂度（直接组合论证）

设d项。对于给定的项目，它可能被放置在规则的 LHS、RHS 或完全被排除在外，这说明了给定项目的 3 种可能性，这为所有d项目提供了 3 d种可能性。

但以上包括带有空白 LHS 或空白 RHS 的规则，它们不是有效的规则。
RHS 为空白的规则数为2d，其中还包括一个两者都有空白的规则侧面；排除这条规则会得到2d – 1，它表示具有非空白 LHS 但空白 RHS 的规则。

类似地考虑非空白 RHS 但空白 LHS 占另一个2d – 1 规则。

结合以上给出 2×(2d – 1) = 2d+1 – 2 规则，其中一个空白 LHS 或一个空白 RHS，但两边都不是空白。

将一条带有空白 LHS 和空白 RHS 的规则添加到2d+1 – 2 +1 = 2d+1 – 1，这表示要排除的规则总数。这些是左侧空白、右侧空白或两侧空白的（无效）规则。

从3d的无限可能性中减去这些规则，我们得到总数有效规则数为3d – 2 d+1 + 1，与上面的分析证明一致。

矿业协会规则

时间项目	
1	面包、牛奶
2	面包、尿布、啤酒、鸡蛋
3	牛奶、尿布、啤酒、可乐
4	面包、牛奶、尿布、啤酒
5	面包、牛奶、尿布、可乐

规则示例：

{牛奶、尿布} → {啤酒} (s=0.4, c=0.67)
{牛奶、啤酒} → {尿布} (s=0.4, c=1.0)
{尿布、啤酒} → {牛奶} (s=0.4, c=0.67)
{啤酒} → {牛奶、尿布} (s=0.4, c=0.67)
{尿布} → {牛奶、啤酒} (s=0.4, c=0.5)
{牛奶} → {尿布、啤酒} (s=0.4, c=0.5)

- 上述所有规则都是同一项目集的二元分区：
  {牛奶、尿布、啤酒}
- 源自相同项集的规则具有相同的支持，但可以有不一样的自信

减少搜索空间

- 支持度超过阈值最小值的项集称为大（或频繁)项集,这里的大表示支持度大
- 如果项目集的基数非常高,那么发现所有大型项目集以及支持值是一个主要问题
- 一个典型的超市有数千种商品
  - 不同（非空)项集的数量为 2 m项,并且对所有可能项集的计数支持变, 其中m是数量得计算密集型。
- 为了减少组合搜索空间,寻找算法关联规则利用两个属性

减少搜索空间

先验原则：

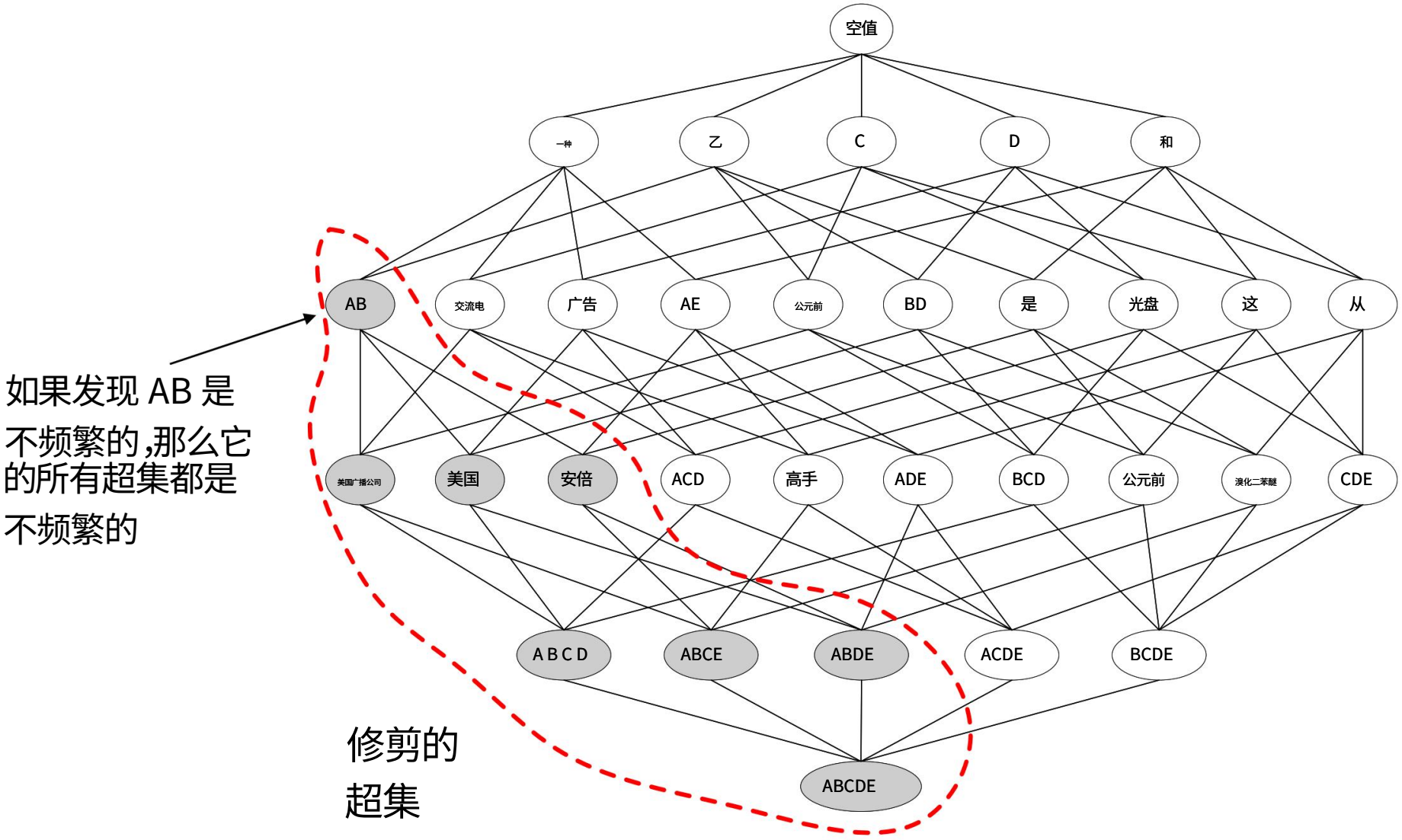
- 如果一个项集是频繁的，那么它的所有子集也一定是频繁的
- 一个项集的支持度永远不会超过其子集的支持度

$$X, Y : (X \quad Y) \quad s(X) \quad s(Y)$$

- 向下关闭
  - 大项集的子集也必须很大（即，大项集的每个子集都超过了所需的最小支持度）。
- 反单调性
  - 相反，小项集的超集也很小（暗示它没有足够的支持）。
- 因此，一旦发现一个项目集很小（不是一个大项目集），那么通过将一个或多个项目添加到该集合而形成的对该项目集的任何扩展也将产生一个小项目集

减少搜索空间

Reduce Search Space



候选生成：Fk-1 x Fk-1方法

- 如果它们的第一个 (k-2) 项相同，则合并两个频繁 (k-1) 项集

- F3 = {ABC,ABD,ABE,ACD,BCD,BDE,CDE} 合并
  (ABC, ABD) = ABCD 合并(ABC, ABE) = ABCE
  合并(ABD, ABE) = ABDE

  - 不要合并(ABD,ACD)，因为它们只共享长度为 1 而不是长度为 2 的前缀

说明先验原理

物品	数数
面包	4
可乐	2
牛奶	4
啤酒	3
尿布	4
蛋	1

项目（1-项目集）

物品集	数数
{面包,牛奶}	3
{面包、啤酒}	2
{面包,尿布}	3
{牛奶、啤酒}	2
{牛奶,尿布}	3
{啤酒,尿布}	3

对（2项集）

（无需生成
涉及可口可乐的候选人
或鸡蛋）

最低支持 = 3

三胞胎（3项集）

物品集	数数
{面包、尿布、牛奶}	2

如果考虑每个子集，

6C1 + 6C2 + 6C3
6 + 15 + 20 = 41

使用基于支持的修剪，

6 + 6 + 1 = 13

使用Fk-1xFk-1方法生成候选结果
只有一个 3 项集。这在支持计数步骤之后被消除。

多维关联

- 考虑一个包含三个客户交易的文件
  方面
  - 交易 ID
  - 时间
  - 购买的物品
- 以下规则是我们包含单个维度标签的示例：

  Items-Bought(milk) Items-Bought(juice) ▪有

时，查找涉及多个维度的关联规则可能会很有趣，例如Time(6:30...8:00)
Items-Bought(milk))

- 像这样的规则称为多维关联
  规则

聚类stering

- 在没有训练样本的情况下对数据进行分区通常很有用
  - 这是一个无监督学习的例子
- 在业务中,确定以下群体可能很重要
  对于具有相似购买模式或在医学领域的客户,确定对处方药表现出相似反应的患者组可能很重要。
- 聚类的目标是将记录分组,使得一个组中的记录彼此相似,而与其他组中的记录不同,并且组通常是不相交的。

聚类 Clustering

- 聚类的一个重要方面是使用的相似度函数。
- 欧几里得距离通常用于衡量相似性：

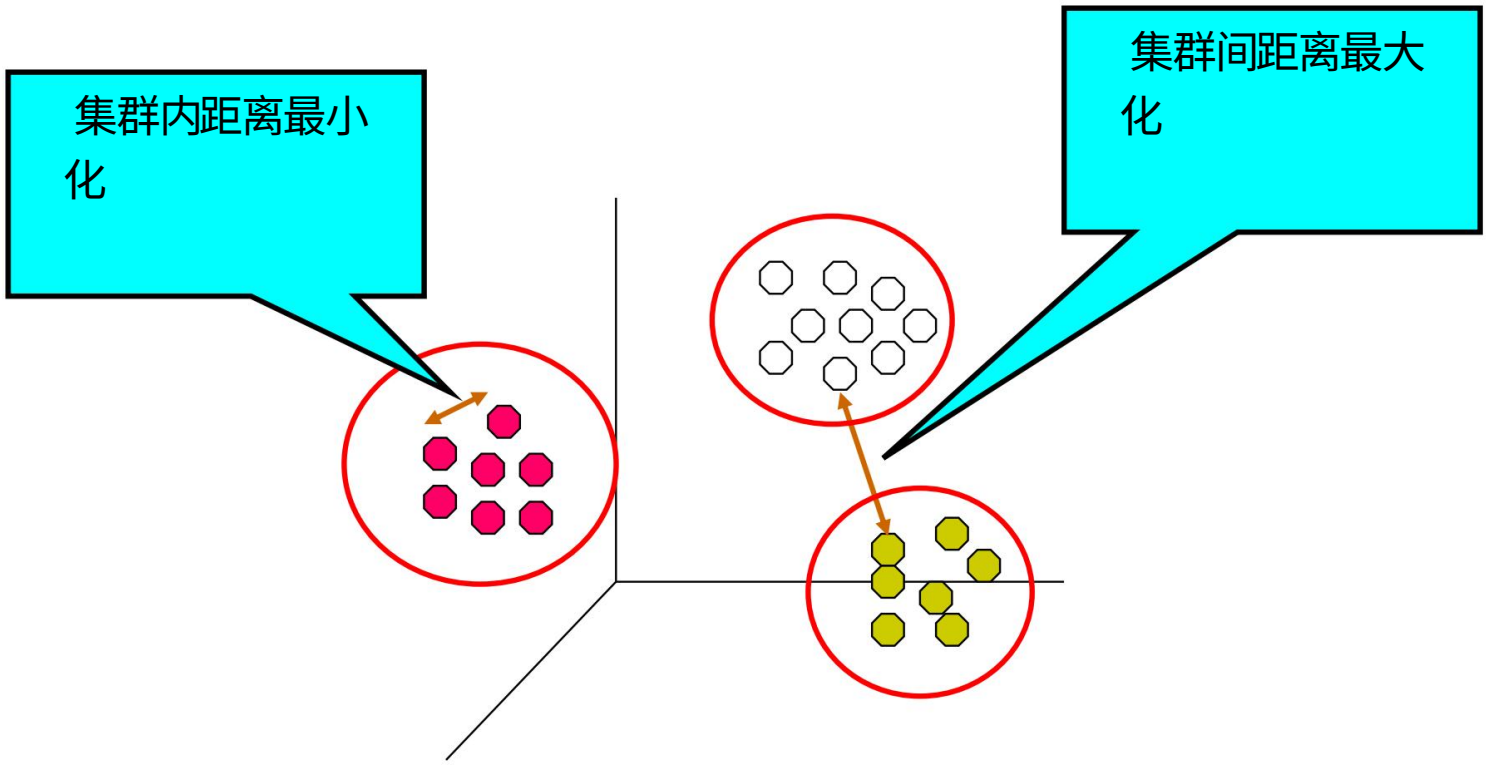
  将两个n 维数据记录视为n维空间中的点x和y 。我们可以将第 i 个维度的值视为两条记录的xi和yi 。 n 维空间中点x = (x1 ,⋯,xn) 和y = (y1 ,⋯,yn)之间的欧几里得距离为

$$\sqrt{\sum_{i=1}^{n} |\quad|^2}$$

  两点之间的距离越小,相似度越大

聚类分析

- 查找对象组,使组中的对象彼此相似但与其他组中的对象不同



K-Means 聚类

- 每个集群都与一个质心（中心点）相关联
- 每个点都分配给具有最近质心的集群
- 必须指定簇数K
- 基本算法非常简单

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

K-Means 聚类

考虑下表

记录	年龄	服务年限
1	30	5
2	50	25
3	50	15
4	25	5
5	30	10
6	55	25

假设所需聚类的数量K为 2。让算法选择聚类C1的记录 3 和聚类C2的记录 6作为初始聚类质心。剩余的记录将在重复循环的第一次迭代期间分配给这些集群中的一个。

K-Means 聚类

记录 1 与C1的距离为 $\sqrt{(202 + 102)} = 22.4$,与C2的距离
32.0,所以它加入集群C1 。记录 2 与C1的距离为 10.0,与C2的距离为 5.0,因此它加入了集群C2 ,依此类推。

创纪录的时代		服务年限	距离 3	距离 6
1	30	5	22.4	32.0
2	50	25	10.0	5.0
3	50	15	0	-
4	25	5	25.5	36.6
5	30	10	20.6	29.2
6	55	25	-	0

因此我们有集群

C1 = {记录 1、记录3、记录 4、记录 5}

C2 = {记录 2,记录6}。

K-Means 聚类

接下来,计算新的质心:

C1的新质心为((30+50+25+30)/4, (5+15+5+10)/4)

= (33.75, 8.75)

C2的新质心为((50+55)/2, (25+25)/2) = (52.5, 25)

在第二次迭代中,六个记录被放置到两个集群中,如下所示:

C 1 = {记录 1、记录 4、记录 5}

C 2 = {记录 2、记录 3、记录 6}。

C 1和 C 2的平均值被重新计算为 (28.3, 6.7) 和 (51.7, 21.7)。

在下一次迭代中,所有记录都保留在它们之前的集群中,算法终止。