数据库系统概论新技术篇

数据挖掘

李翠平

中国人民大学信息学院

概　览

- 什么是关联规则？
- 基本概念
- 关联规则挖掘的步骤
- 一种典型的关联规则挖掘算法：Apriori方法

什么是关联规则

- 关联规则是发现交易数据库中不同商品（项）之间的联系
- 这些规则找出顾客购买行为的模式

  如购买了某一商品对购买其他商品的影响
- 发现这样的规则可以应用于商品货架设计、存货安排以及根据购买模式对用户进行分类。

“尿布和啤酒”的例子

观察发现很多顾客在买尿布的时候同时也购买啤酒，这样超市把尿布和啤酒放在一起就可以提高销售额，如果把土豆片摆在它们中间，则会同时提高这三者的销售额。

基本概念

- 规则：{x1，x2，…,xn}→Y
  - 如果顾客把商品x1，x2，…, xn放入购物篮中的话，则很可能也会把商品Y放入其中。
  - 这个可能性（购买x1，x2，…,xn的前提下购买Y）称作规则的可信度，人们通常只对那些可信度大于某个值的规则感兴趣，希望把这些规则找出来。
  - 当然，有可能出现这样的情况，某些商品是被随机放入购物篮中的，例如：类似这样的规则，{牛奶，黄油}—〉面包，它的可信度非常大，可能是因为很多人都购买面包，象这样的规则是无用的。

基本概念

- 可信度和最小可信度

  购买x1，x2，…, xn 的情况下购买Y的可能性, 条件概率

- 支持度和最小支持度

  同时购买x1，x2，…,xn　和Y的可能性

- 频繁项目集

  满足最小支持度的项目集

基本概念

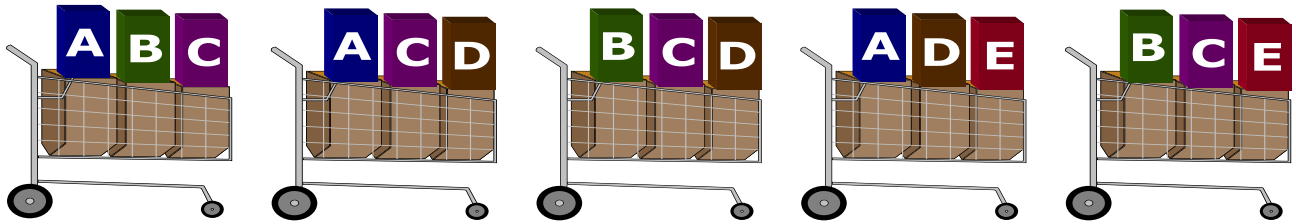
支持度：

$support\ (A \Rightarrow B) = P(A \cup B)$

可信度：

$confidence\ (A \Rightarrow B) = P(B|A)$

Association Rules



Rule	Support	Confidence
A ⇒ D	2/5	2/3
C ⇒ A	2/5	2/4
A ⇒ C	2/5	2/3
B & C ⇒ D	1/5	1/3

因果关系

- 理想情况下，人们都希望从关联规则中得出这样的结论，因为购买了商品x1，x2，…,xn，所以一定会购买商品Y。
- 但是因果关系很不好确定，下面举一个例子来说明对购物篮数据来说，因果关系意味着什么。
  - 如果商店降低尿布的价格而升高啤酒的价格，则会吸引尿布购买者，而这些购买尿布的人同时很可能会购买啤酒，这样商店在尿布上遭受的损失就会从啤酒中得以弥补，这是因为“尿布是因，啤酒是果”。相反，如果降低啤酒的价格而升高尿布的价格，买啤酒的人多了但买尿布的人并不会增多，商店就赔本了。

重要的公理

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。

关联规则挖掘

通常被分解为两个子问题：

1. 根据用户输入的最小支持度，寻找频繁项目集

2. 根据用户输入的最小可信度，产生关联规则

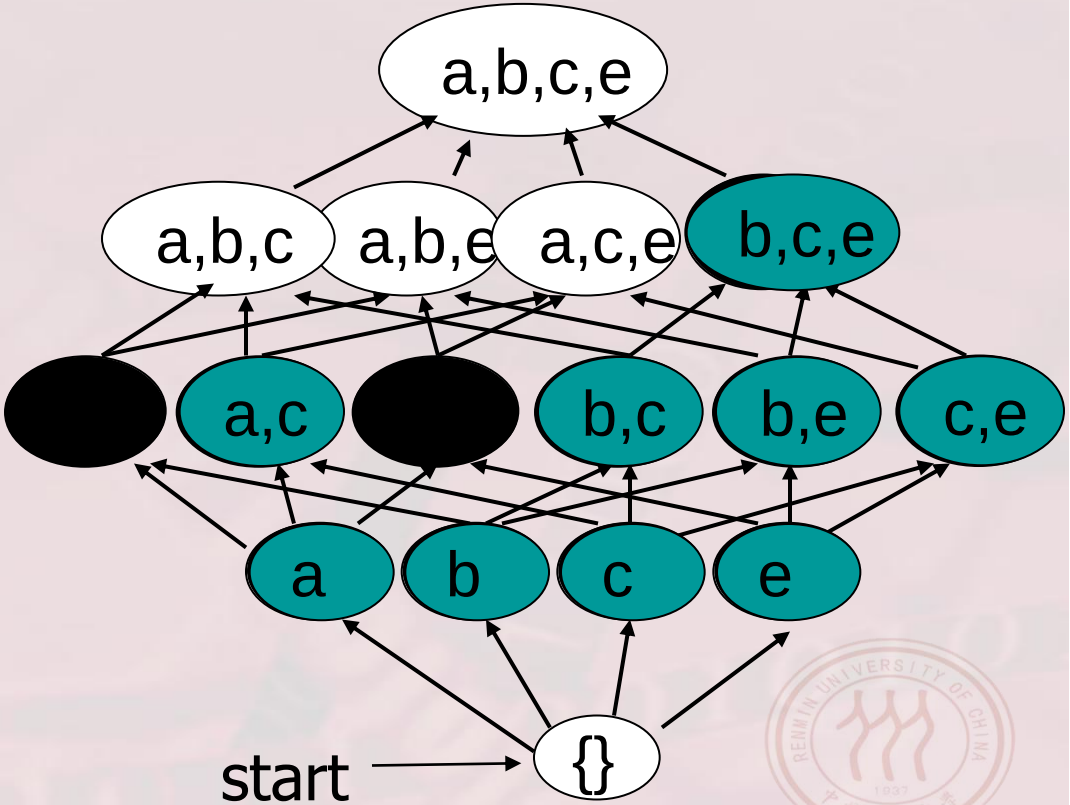
第二个问题比较简单，关键是解决第一个问题

Apriori算法第一步：寻找频繁项目集

1. 给出minisupp，第一次扫描数据的时候，找出频度大于minisupp的项目，称这集合为L1。一般来说，一个商店卖的商品品种不会超过10万个，所以我们假定有足够的内存来计算每个项目的出现频度。
2. 第二次扫描数据时，L1中的项目对成为大小为2的候选项目集C2，我们希望C2不要太大，这样的话就可以有足够的空间为每个候选对分配一个计数器来计算它的出现频度，计数值大于minisupp的候选对构成大小为2的频繁项目集L2。
3. 第三次扫描数据时，由L2生成大小为3的候选项目集C3，C3是这样的集合{A，B，C}，并且{A，B}，{B，C}，{A，C}都在L2中。计算C3中每个三元组的出现频度，大于minisupp构成L3。
4. 一直进行到集合为空时为止。Li是大小为i的频繁项目集，Ci+1是大小为i+1的集合，这些集合的每个大小为i 的子集都在Li中。

Apriori算法第一步：寻找频繁项目集

- 由底至上，宽度优先搜索
- 对数据只进行读取操作
- 递归进行如下两步：
  - 生成候选集
  - 计数，获得真正的频繁项目集



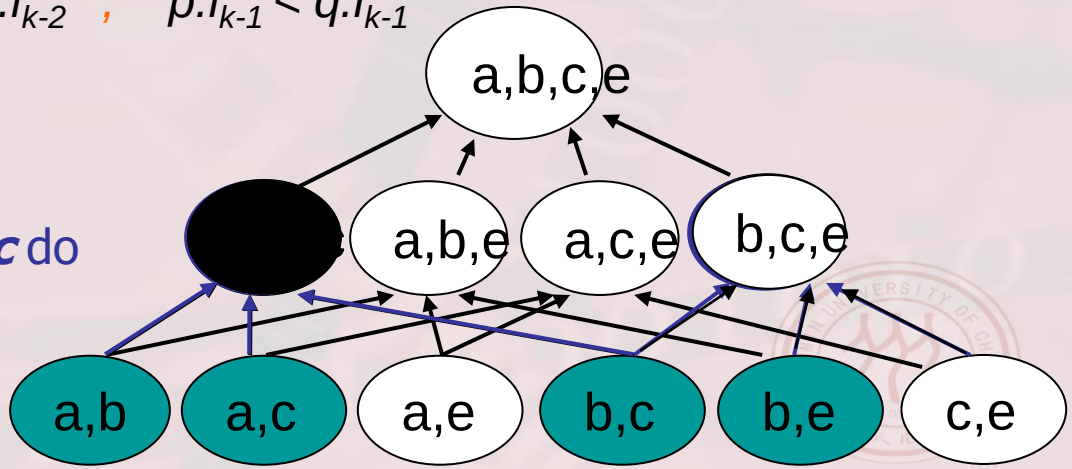
候选集产生和修剪

- Suppose all frequent (k-1) items are in L_{k-1}
- **Step 1: Self-joining L_{k-1}**

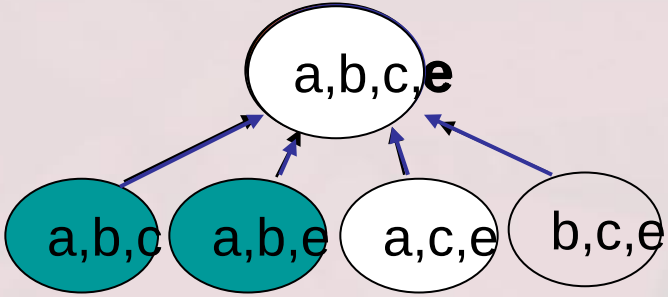
  insert into C_k
  select $p.i_1, p.i_2, ..., p.i_{k-1}, q.i_{k-1}$
  from $L_{k-1}\ p,\ L_{k-1}\ q$
  where $p.i_1=q.i_1\ ,\ ...\ ,\ p.i_{k-2}=q.i_{k-2}\ ,\ p.i_{k-1} < q.i_{k-1}$

- Step 2: pruning

  forall ***itemsets c in*** C_k do
  forall ***(k-1)-subsets s of c*** do
  if (*s is not in* L_{k-1}) **then**
  delete *c* **from** C_k



候选集产生和修剪



Apriori算法例子

$Sup_{min} = 2$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

C_1

1st scan

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

2nd scan

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

Apriori 算法步骤2: 产生关联规则

- 对于每个频繁项目集L，产生它的所有非空子集S
- 对L的每个非空子集S，如果满足如下条件：

$$\frac{\sup port\_count(L)}{\sup port\_count(S)} \geq \min\_conf$$

则输出关联规则 S→(L – S)

谢 谢！