上次课回顾

- 数据挖掘的基本过程
- 基本概念：DM、ML、DL、KDD……
- 深度学习的三大经典模型
- 十大经典算法

- 实验：
  - 手写数字体识别
  - 混淆矩阵、精确率、召回率、F1指数
  - 主成分分析（选做）

第三章 关联规则挖掘理论和算法

内容提要

- 基本概念与解决方法
- 经典的频繁项目集生成算法分析
- Apriori算法的性能瓶颈问题
- Apriori的改进算法
- 对项目集格空间理论的发展
- 基于项目序列集操作的关联规则挖掘算法
- 改善关联规则挖掘质量问题
- 约束数据挖掘问题
- 关联规则挖掘中的一些更深入的问题
- 数量关联规则挖掘方法

关联规则挖掘是数据挖掘研究的基础

- 关联规则：Association Rule Mining，就是发现数据背后存在的某种规则或者联系。
- 最早是由Agrawal等人提出的（1993）。最初提出的动机是针对购物篮分析（Basket Analysis）问题提出的，其目的是为了发现交易数据库（Transaction Database）中不同商品之间的联系规则。
- 关联规则的挖掘工作成果颇丰。例如，关联规则的挖掘理论、算法设计、算法的性能以及应用推广、并行关联规则挖掘（Parallel Association Rule Mining）以及数量关联规则挖掘（Quantitive Association Rule Mining）等。

应用举例

- 通过调研超市顾客购买的东西，可以发现30%的顾客会同时购买床单和枕套，而在购买床单的顾客中有80%的人购买了枕套，这就存在一种隐含的关系：床单→枕套，也就是说购买床单的顾客会有很大可能购买枕套，因此商场可以将床单和枕套放在同一个购物区，方便顾客购买。
- 一般，关联规则可以应用的场景有：
  - 优化货架商品摆放或者优化邮寄商品的目录
  - 交叉销售或者捆绑销售
  - 网络安全领域中的入侵检测技术；
  - 搜索词推荐或者识别异常
  - 在移动通信领域中，指导运营商的业务运营和辅助业务提供商的决策制定。

第三章 关联规则挖掘理论和算法

内容提要

- 基本概念：关联规则的应用领域
- 经典的频繁项目集生成算法分析
- Apriori算法的性能瓶颈问题
- Apriori的改进算法
- 对项目集格空间理论的发展
- 基于项目序列集操作的关联规则挖掘算法
- 改善关联规则挖掘质量问题
- 约束数据挖掘问题
- 关联规则挖掘中的一些更深入的问题
- 数量关联规则挖掘方法

- 商场布局

关联规则的应用

- 文本挖掘
  - 一个句子（段落）——购物篮
  - 每一个词——商品
  - 词语词之间的频繁模式

关联规则的应用

表5 莎拉波娃———最后4拍击球落点关联规则分析结果(对手最后1拍)Table 5 The Result of Association Rule Analysis of Ball Placement of the Last Four Shots from Maria Sharapova(Last Shot is Opponent)

下载原表

序号	规　则	支持度/%	置信度/%	规则提升度
1	莎娃倒数第1拍 = L5 对手最后1拍 = C = = > 莎拉波娃得分	13.79	67.86	1.391
2	对手倒数第2拍 = L7 and 莎娃倒数第1拍 = L4，对手最后1拍 = S = = > 莎拉波娃得分	6.93	70	2.186
3	对手倒数第2拍 = L7 对手最后1拍 = L9 = = > 莎拉波娃失分	29.49	69.57	1.55
4	莎娃倒数第2拍 = L8 and 对手倒数第2拍 = L7，对手最后1拍 = L9 = = > 莎拉波娃失分	7.69	80	1.783
5	莎娃倒数第2拍 = L7 and 莎娃倒数第1拍 = L5，对手最后1拍 = L9 = = > 莎拉波娃失分	7.69	80	1.783
6	莎娃倒数第2拍 = L7 and 对手倒数第2拍 = L7，对手最后1拍 = L9 = = > 莎拉波娃失分	7.69	80	1.783

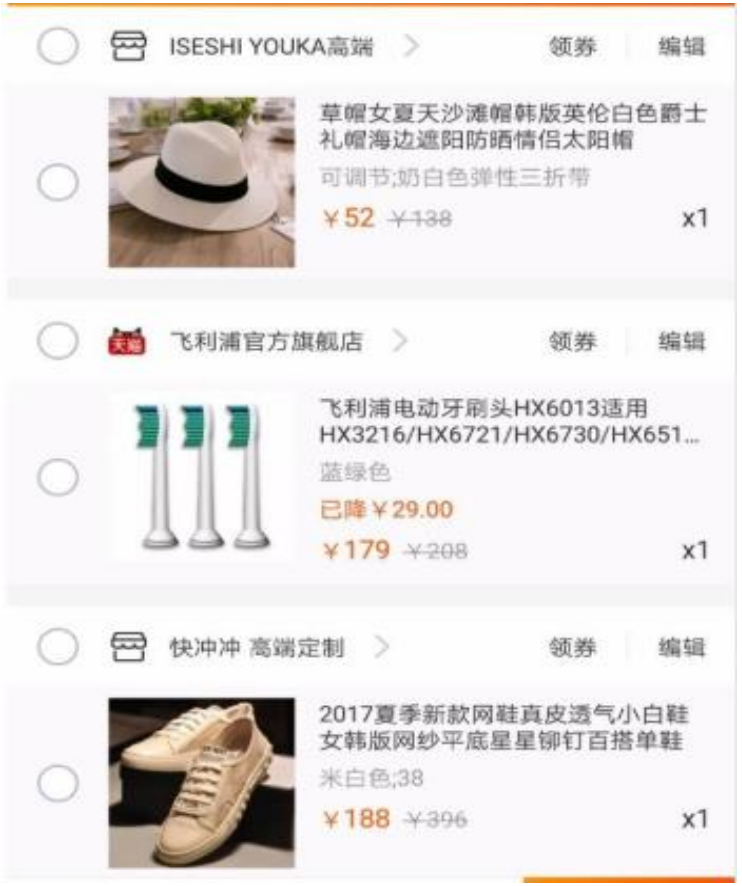
第三章 关联规则挖掘理论和算法

内容提要

- 基本概念： 项集、可信度、支持度、频繁项集
- 经典的频繁项目集生成算法分析
- Apriori算法的性能瓶颈问题
- Apriori的改进算法
- 对项目集格空间理论的发展
- 基于项目序列集操作的关联规则挖掘算法
- 改善关联规则挖掘质量问题
- 约束数据挖掘问题
- 关联规则挖掘中的一些更深入的问题
- 数量关联规则挖掘方法

关联规则的基本概念

- 项：（Item）面包、牛奶、巧克力、饼干……
- 交易：（购物篮，Transcation）由商品所构成的非空的集合



关联规则的基本概念

- 交易数据库：记录顾客每次购买的所有商品信息的数据库

购物篮

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

关联规则的基本概念

- 交易数据库中的每一行对应一次交易

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

关联规则基本概念

- 项（item）：beer、biscuit、nuts……
- 所有项的集合（I）：{所有商品}
- 项集（Itemset）：{部分商品}

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

关联规则（association rule）

- 关联规则（association rule）：是形如 X → Y 的蕴含表达式，其中X和Y是不相交的项集，即：X∩Y=∅。
- 项集：在关联分析中，包含0个或多个项的集合被称为项集（itemset）。如果一个项集包含k个项，则称它为k-项集。例如:{床单，枕套，牛奶，花生}是一个4-项集。空集是指不包含任何项的项集。
- 频数：一个项集x在数据库D中出现的次数，count
- 关联规则的强度可以用它的支持度（support）和可信度（confidence）来度量。

关联规则基本概念

- X={beer}，是几项集，频数是多少？

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

关联规则的基本概念

- X={beer}，是几项集，频数是几？
- X={beer}，1-项集，count（X）=4

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

关联规则的基本概念

- X={beer，diaper}，是几项集，频数是多少？

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

关联规则的基本概念

- X={beer，diaper}，是几项集，频数是多少？
- X={beer，diaper}，2-项集，count（X）=3

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

小结

- 项集的长度：一个项集包含的项的个数
- k项集：包含k个项的项集
- 频数：一个项集x在数据库D中出现的次数，count

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

第三章 关联规则挖掘理论和算法

内容提要

- 基本概念： 项集、可信度、支持度、频繁项集
- 经典的频繁项目集生成算法分析
- Apriori算法的性能瓶颈问题
- Apriori的改进算法
- 对项目集格空间理论的发展
- 基于项目序列集操作的关联规则挖掘算法
- 改善关联规则挖掘质量问题
- 约束数据挖掘问题
- 关联规则挖掘中的一些更深入的问题
- 数量关联规则挖掘方法

支持度(Support)

交易号码	商品
0	豆奶，莴苣
1	莴苣，尿布，葡萄酒，甜菜
2	豆奶，尿布，葡萄酒，橙汁
3	莴苣，豆奶，尿布，葡萄酒
4	莴苣，豆奶，尿布，橙汁

支持度：一个项集或者规则在事物数据库中出现的频率。

{豆奶}的支持度　为4/5
{尿布}的支持度　为4/5
{豆奶、尿布}　　为3/5

支持度

- $$support(X) = \frac{count(X)}{|D|} \times 100\%$$ 项集X的**支持度**，记为support（X），其中D的模代表D中交易的个数。

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

- 求X={beer，biscuit，diaper}的支持度？
- 1/5=20%

频繁项集

- 一个项集X的支持度大于用户给定的一个最小支持度阈值，则X被称为频繁项集（或频繁模式），X是频繁的。

交易号码	商品
0	豆奶，莴苣
1	莴苣，尿布，葡萄酒，甜菜
2	豆奶，尿布，葡萄酒，橙汁
3	莴苣，豆奶，尿布，葡萄酒
4	莴苣，豆奶，尿布，橙汁

support{豆奶}=80%
support{尿布}=80%
support{豆奶、尿布}=60%
令最小支持度为80%

频繁项集：{豆奶}{尿布}
{豆奶、尿布}?

第三章 关联规则挖掘理论和算法

内容提要

- 基本概念：项集、可信度、支持度、置信度、频繁项集、
- 经典的频繁项目集生成算法分析
- Apriori算法的性能瓶颈问题
- Apriori的改进算法
- 对项目集格空间理论的发展
- 基于项目序列集操作的关联规则挖掘算法
- 改善关联规则挖掘质量问题
- 约束数据挖掘问题
- 关联规则挖掘中的一些更深入的问题
- 数量关联规则挖掘方法

置信度（confidence）

$$Confidence\ (X \rightarrow Y) = P(Y/X) = \frac{P(X, Y)}{P(X)} = \frac{\text{同时购买}\ \{X, Y\}\ \text{的订单}}{\text{购买}X\text{的订单}}$$

交易号码	商品
0	豆奶，莴苣
1	莴苣，尿布，葡萄酒，甜菜
2	豆奶，尿布，葡萄酒，橙汁
3	莴苣，豆奶，尿布，葡萄酒
4	莴苣，豆奶，尿布，橙汁

求：{尿布}→{豆奶}的置信度？

其中{尿布，豆奶}的支持度为3/5，{尿布}的支持度为4/5，

所以“尿布→豆奶”的可信度为

3/5 ÷ 4/5 =0.75

最小置信度

- **最小置信度**：一个关联规则的置信度大于某一个预先给定的阈值，这个阈值一般称为minimum confidence

· 如果一个规则X→Y同时满足support(X→Y)≥minsup和confidence(X→Y)≥minconf，则称该规则在数据库D中成立。

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

计算规则{beer、diaper}→nuts的置信度？

confidence({beer, diaper}→nuts)=66.7%。

提升度 (Lift)

提升度： 表示先购买A对购买B的概率的提升作用，用来判断规则是否有实际价值，即使用规则后商品在购物车中出现的次数是否高于商品单独出现在购物车中的频率。如果大于1说明规则有效，小于1则无效。

$$Lift(X->Y) = \frac{P(X,Y)}{P(X)\cdot P(Y)} = \frac{P(Y/X)}{P(Y)} = \frac{\text{同时购买}\{X,Y\}\text{的订单}\cdot\text{总订单}}{\text{购买}X\text{的订单}\cdot\text{购买}Y\text{的订单}}$$

- 提升度 >1 且越高表明正相关性越高；
- 提升度 <1 且越低表明负相关性越高；
- 提升度 =1 表明没有相关性。

提升度（Lift）

交易号码	商品
0	豆奶，莴苣
1	莴苣，尿布，葡萄酒，甜菜
2	豆奶，尿布，葡萄酒，橙汁
3	莴苣，豆奶，尿布，葡萄酒
4	莴苣，豆奶，尿布，橙汁

求：{尿布}->{豆奶}的提升度？

其中{尿布，豆奶}的支持度为3/5，{尿布}的支持度为4/5，{豆奶}的支持度为4/5，

所以“尿布->豆奶”的提升度为

3/5 ÷ （4/5× 4/5 ）=0.9375

第三章 关联规则挖掘理论和算法

内容提要

- 基本概念： 项集、可信度、支持度、置信度、频繁项集、
- 经典的频繁项目集生成算法分析
- Apriori算法的性能瓶颈问题
- Apriori的改进算法
- 对项目集格空间理论的发展
- 基于项目序列集操作的关联规则挖掘算法
- 改善关联规则挖掘质量问题
- 约束数据挖掘问题
- 关联规则挖掘中的一些更深入的问题
- 数量关联规则挖掘方法

频繁项集

- 给定最小支持度阈值minsup，一个频繁项集的所有非空子集，都是频繁项集。

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

- 若minsup=40%，则
- 项集{beer、diaper、nuts}是不是频繁项集？
- 项集{beer、nuts}是不是频繁项集？
- 项集{diaper}是不是频繁项集？……

频繁项集

- 如果一个项集是不频繁项集，则其所有的超集都是不频繁项集。

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

- 若minsup=40%，则
- 项集{beer、cheese、nuts}是不是频繁项集？
- 项集{beer、 cheese、nuts、diaper}是不是频繁项集？

强关联规则（Strong Association Rule）

- **置信度**：描述的是一个关联规则到底有多可信

- 关联规则（association rule）：是形如 X → Y 的蕴含表达式，X和Y是不相交的项集，即：X∩Y=∅。其中X为规则的**前件**，Y为规则的**后件**

$$Confidence\ (X->Y) = P(Y/X) = \frac{P(X,Y)}{P(X)} = \frac{\text{同时购买}\ \{X,Y\}\ \text{的订单}}{\text{购买}X\text{的订单}}$$

- **强关联规则**：D在I上满足最小支持度和最小信任度（Minconfidence）的关联规则称为强关联规则（Strong Association Rule）。

最小置信度

- **最小置信度**：一个关联规则的置信度大于某一个预先给定的阈值，这个阈值一般称为minimum confidence

· 如果一个规则X→Y同时满足support(X→Y)≥minsup和confidence(X→Y)≥minconf，则称该规则在数据库D中成立。

交易号（TID）	商品（Items）
1	beer, diaper, nuts
2	beer, biscuit, diaper
3	bread, butter, cheese
4	beer, cheese, diaper, nuts
5	beer, butter, cheese, nuts

令最小置信度为60%，规则{beer、diaper}→nuts是强关联规则吗？

confidence({beer, diaper}→nuts)=66.7%。

关联规则挖掘基本过程

- 关联规则挖掘问题可以划分成两个子问题：
  - 1. 发现所有的频繁项集:通过用户给定Minsupport，寻找所有频繁项目集或者最大频繁项目集。
  - 2、从频繁项集中发现关联规则:通过用户给定Minconfidence，在频繁项目集中，寻找强关联规则。

第三章 关联规则挖掘理论和算法

内容提要

- 基本概念与解决方法
- 经典的频繁项目集生成算法分析: Apriori算法
- Apriori算法的性能瓶颈问题
- Apriori的改进算法
- 对项目集格空间理论的发展
- 基于项目序列集操作的关联规则挖掘算法
- 改善关联规则挖掘质量问题
- 约束数据挖掘问题
- 关联规则挖掘中的一些更深入的问题
- 数量关联规则挖掘方法

关联规则-Apriori算法

- Apriori算法是一种挖掘关联规则的频繁项集算法，其核心思想是通过候选集生成和情节的向下封闭检测两个阶段来挖掘频繁项集。

- 关联规则算法的主要应用是购物篮分析，是为了从大量的订单中发现商品潜在的关联。其中常用的一个算法叫Apriori（先验）算法。

- Apriori算法是一种用于挖掘数据集内部关联规则的算法，“apriori”的意思为“先验的”，这个算法是用先验知识来预测数据的关联规则的

令最小支持度50%，最小置信度80%

Database TDB

Tid	Items
1	A, C, D
2	B, C, E
3	A, B, C, E
4	B, E

1st scan

C_1

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}

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

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

C_3

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

3rd scan

L_3

Itemset	sup
{B, C, E}	2

关联规则的生成问题

- 关联规则挖掘问题可以划分成两个子问题：

- 1、发现所有的频繁项集：
  - Apriori算法：逐层发现算法，按照项集的长度由小到大逐级进行，并最后发现频繁N项集。

- 2、在得到了频繁项目集后，可以按照下面的步骤生成强关联规则：
  - 对于每一个频繁项目集L，生成其所有的非空子集；
  - 对于L的每一个非空子集x，计算Confidence（x），如果Confidence（x）≥minconfidence，那么“x⇨（*l*-x）”成立。

关联规则的生成问题

- 1、找出频繁项集：开始数据库里有4条交易，min_support=50%

Database TDB

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

Itemset	sup
{B, C, E}	2

- 2、在频繁项目集中找强关联规则

· 如果一个规则X→Y同时满足support(X→Y)≥minsup和confidence(X→Y) ≥minconf, 则称该规则在数据库D中成立。

关联规则的生成问题

- 2、min_confidence=80%作为可信度阈值，生成关联规则

Itemset	sup
{B, C, E}	2

频繁项集（支持度）	前项（支持度）	后项（支持度）	规则	可信度	是否关联规则
B、C、E（50%）					

关联规则的生成问题

- 2、min_confidence=80%作为可信度阈值，生成关联规则

Itemset	sup
{B, C, E}	2

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

频繁项集（支持度）	前项（支持度）	后项	规则	可信度	是否关联规则
B、C、E（50%）	B、C（50%）				

关联规则的生成问题

- 2、min_confidence=80%作为可信度阈值，生成关联规则

Itemset	sup
{B, C, E}	2

频繁项集（支持度）	前项（支持度）	后项	规则	可信度	是否关联规则
B、C、E（50%）	B、C（50%）	E	BC→E	100%	

关联规则的生成问题

- 2、min_confidence=80%作为可信度阈值，生成关联规则

Itemset	sup
{B, C, E}	2

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

频繁项集（支持度）	前项（支持度）	后项（支持度）	规则	可信度	是否关联规则
B、C、E（50%）	B、C（50%）	E	BC→E	100%	是
	B、E（75%）				

关联规则的生成问题

- 2、min_confidence=80%作为可信度阈值，生成关联规则

Itemset	sup
{B, C, E}	2

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

频繁项集（支持度）	前项（支持度）	后项（支持度）	规则	可信度	是否关联规则
B、C、E（50%）	B、C（50%）	E	BC→E	100%	是
	B、E（75%）	C	BE→C	67%	否
	C、E（50%）	B	CE→B	100%	是
	B（75%）	CE	B→CE	67%	否
	C（75%）	BE	C→BE	67%	否
	E（75%）	BD	E→BD	67%	否

小结-Apriori算法

- 发现所有的频繁项集
- 从频繁项集中发现关联规则

Apriori算法

Database TDB

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

C_1

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

1st scan

L_1

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

C_2

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

2nd scan

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

在频繁项目集中，寻找强关联规则。

- 开始数据库里有4条交易，min_confidence=80%作为可信度阈值，生成强关联规则

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

Itemset	sup
{B, C, E}	2

最大频繁项集（支持度）	子集（支持度）	可信度	规则	是否强规则
B、C、E（50%）	B、C（50%）	100%	BC→E	是
B、C、E（50%）	B、E（75%）	67%	BE→C	否
B、C、E（50%）	C、E（50%）	100%	CE→B	是
B、C、E（50%）	B（75%）	67%	B→CE	否
B、C、E（50%）	C（75%）	67%	C→BE	否
B、C、E（50%）	E（75%）	67%	E→BD	否

· 算法：Apriori

· 输入：交易数据库D，最小支持度阈值minsup

· 输出：D中的所有频繁项集的集合F

· 主要步骤：

(1) Find all of frequent items from D and save them in F_1, k=l;

(2) if F_k is not empty then begin

(3) C_{k+1}=gen_candidate(F_k);

(4) for each transaction t in D begin

(5) for any candidate c 属于 C_{k+1}

(6) if itemset c occurs in t then

(7) c.count++;

(8) end for

(9) F_{k+1}=(c 属于 C_{k+1} | support(c) ≥minsup);

(10) k=k+1

利用已经发现的频繁k项集生成候选（k+1）项集

扫描数据库D，对每个候选（k+）项集统计其出现次数，计算支持度，得到频繁的（k+1）项集集合

Apriori算法的主要步骤

- 1、扫描全部数据，产生候选1-项集的集合C1.
- 2、根据最小支持度，由候选1-项集的集合C1产生频繁1-项集的集合L1.
- 3、对k>1,重复执行步骤4,5,6
- 4、由Lk执行连接和减枝操作，产生候选k+1-项集的集合Ck+1
- 5、根据最小支持度，由候选（k+1）-项集的集合Ck+1，产生频繁（k+1）-项集的集合Lk+1
- 6、若L不等于∅，则k=k+1，步骤跳4，否则结
- 7、束根据最小置信度，由频繁项集产生强关联规则，结束

第三章 关联规则挖掘理论和算法

内容提要

- 基本概念与解决方法
- 经典的频繁项目集生成算法分析
- Apriori算法的性能瓶颈问题
- Apriori的改进算法
- 对项目集格空间理论的发展
- 基于项目序列集操作的关联规则挖掘算法
- 改善关联规则挖掘质量问题
- 约束数据挖掘问题
- 关联规则挖掘中的一些更深入的问题
- 数量关联规则挖掘方法

Apriori算法的性能瓶颈

- Apriori作为经典的频繁项目集生成算法，在数据挖掘中具有里程碑的作用。

- Apriori算法有两个致命的性能瓶颈：
  - 1. 多次扫描事务数据库，需要很大的I/O负载
    - 对每次k循环，侯选集C_k中的每个元素都必须通过扫描数据库一次来验证其是否加入L_k。假如有一个最大频繁项目集包含10个项的话，那么就至少需要扫描事务数据库10遍。
  - 2. 可能产生庞大的侯选集
    - 由L_{k-1}产生k-侯选集C_k是指数增长的，例如10^4个1-频繁项目集就有可能产生接近10^7个元素的2-侯选集。如此大的侯选集对时间和主存空间都是一种挑战。

Thank you !!!