数据科学导论

Introduction to Data Science

第四章 数据挖掘基础

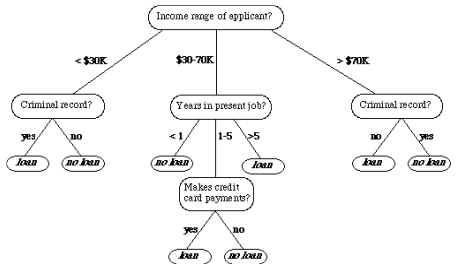
黄振亚，陈恩红
Email: huangzhy@ustc.edu.cn, cheneh@ustc.edu.cn

课程主页：
http://staff.ustc.edu.cn/~huangzhy/Course/DS2023.html

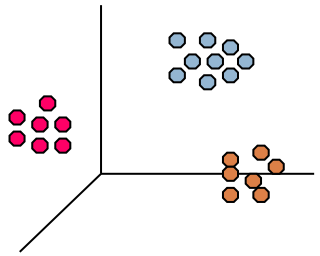
数据挖掘基础

- 数据挖掘——四个任务有哪些常用方法?

分类与预测



聚类



数据

	T		H		P	
	L	H	L	H	L	H
J	-6.0	8.8	60	100	986	1044
F	-2.8	10.9	48	100	973	1025
M	-5.6	17.7	34	100	976	1037
A	-1.2	22.2	27	100	996	1036
M	-0.8	27.8	25	100	1003	1034
J	5.2	29.1	26	100	998	1030
J	9.8	30.6	23	99	997	1027
A	5.6	26.1	31	100	992	1029
S	5.2	24.8	35	100	998	1028
O	-0.4	21.3	42	100	990	1031
N	-7.6	17.3	55	100	963	1023
D	-10.4	9.2	53	100	987	1039

table 17a

2010 monthly weather variation, Cambridge (UK)

关联分析

关联分析

- 数据挖掘任务——关联分析(Association Analysis)
  - 例如： “啤酒与尿布”
  - 在一次圣诞节的顾客消费行为分析中，沃尔玛意外发现跟尿布一起购买最多的商品竟然是啤酒。经过深入分析后，卖场立即对两类商品的空间距离与价格都进行了调整，结果尿布与啤酒销量双双大增。

萨姆·沃尔顿
沃尔玛公司创始人

轰动一时的啤酒与尿布关联规则

关联规则挖掘

- 常用方法 —— 关联规则挖掘 (Association Rule Mining)
  - 给出事务的集合，能够发现一些规则：$A => B$
    - 当事务中某些子项出现时，预测其他子项也出现
  - 例如，从下表中得到一个可能的规则

购买尿布(Diaper)的用户很大可能会购买啤酒(Beer)

——→尿布和啤酒应陈列在一起销售

顾客购物交易数据

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

关联规则挖掘

- 关联规则挖掘的基本概念
  - Itemset（项集）
    - 一个或多个项目(items)的集合
    - k-itemset：大小为k的项集
    - 例：{Milk, Bread, Diaper}是3项集
  - Support （支持度）
    - 一个项集在数据中的出现频率
    - 例：$support$({Milk, Bread, Diaper})= $\frac{2}{5}$
  - **Frequent Itemset （频繁项集）**
    - 用户自行设定最小支持度阈值min $\_sup$，支持度大于min $_sup$的项集称为频繁项集
    - 例：设min $\_sup = 0.3$，则{Milk, Bread, Diaper}为频繁项集

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

关联规则挖掘

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- 关联规则挖掘的基本概念
  - **Association Rule（关联规则）**
    - 形如$X \rightarrow Y$的表达式，X, Y均为项集
    - 例：{Milk, Diaper} →{Beer}
  - Confidence (置信度)
    - 度量包含X的事务中同时出现Y的频率

$$Confidence(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A)}$$

    - 例：对于关联规则{Milk, Diaper} →{Beer}

$$confidence(\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}) = \frac{2}{3}$$

  - 强关联规则
    - 用户自行设定最小置信度阈值min _conf，置信度大于min _conf的规则称为强关联规则
    - 例：设$\text{min_conf} = 0.5$，则{Milk, Diaper} →{Beer}为强关联规则

练习

- 请依据下表计算出关于早餐的关联规则 {面包}->{豆浆} 的置信度

	买豆浆	不买豆浆	
买面包 不买面包	90 390	30 90	120 480
	480	120	600

买面包的次数=120,
买面包的同时买豆浆的次数=90

置信度=$\frac{90}{120}=\frac{3}{4}$

关联规则挖掘

- 关联规则挖掘的一般步骤
  - 根据支持度，**寻找所有的频繁项集**（频繁k项集）
  - 根据频繁项集，生成频繁规则（长度大于2的频繁k项集）
  - 根据置信度，过滤筛选规则
- 关联规则挖掘的第一步：如何寻找所有的频繁项集？
  - 暴力解法:
    - 穷举所有可能的项集，删除小于min _*sup*的项集

⇒ 计算效率低!

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

频繁项集挖掘

- 频繁项集生成的经典算法
  - APriori算法
  - DHP算法(课后学习)
  - FP-Growth算法(课后学习)

APriori算法

Rakesh Agrawal

Technical Fellow, Microsoft Research

在 microsoft.com 的电子邮件经过验证

Data Mining Web Search Education Privacy

- 频繁项集挖掘——APriori算法
  - 1994年，IBM研究员Agrawal提出，VLDB
  - 核心思想：广度优先搜索，自底而上遍历，逐步生成候选集与频繁项集
  - 反单调性原理：如果一个项集是频繁的，则它的所有子集一定也是频繁
    - 成立原因：

$$\forall X, Y: X \subseteq Y \quad \rightarrow \quad support(X) \geq support(Y)$$

    - 依据该性质，对于某k+1项集，只要存在一个k项子集不是频繁项集， 则可以**直接**判定该项集不是频繁项集
  - 算法步骤
    - 连接步：从频繁$K-1$项集生成候选K项集
    - 剪枝步：从候选K项集筛选出频繁K项集

Fast algorithms for mining association rules — 34008 * — 1994
R Agrawal, R Srikant
Proc. 20th int. conf. very large data bases, VLDB 1215, 487-499

Mining association rules between sets of items in large databases — 23560 — 1993
R Agrawal, T Imieliński, A Swami
Proceedings of the 1993 ACM SIGMOD international conference on Management of …

APriori算法：连接步

- A-Priori算法步骤1：连接步
  - 输入：所有频繁$K-1$项集$\boldsymbol{L_{k-1}}$
  - 输出：候选K项集$\boldsymbol{C_k}$
  - 过程：执行自连接$\boldsymbol{L_{k-1} \bowtie L_{k-1}}$，其中$\boldsymbol{L_{k-1}}$的两个项集是可连接的，当且仅当它们前$(k-2)$个项相同
    - 设l_1和l_2是$\boldsymbol{L_{k-1}}$中的项集，且(记号$l_i[j]$表示l_i的第j项)

$$l_1[1] = l_2[1] \wedge l_1[2] = l_2[2] \wedge \ldots \wedge l_1[k-2] = l_2[k-2] \wedge \boldsymbol{l_1[k-1] < l_2[k-1]}$$

    - 为方便计算，假定事务或项集中的项按字典次序排序，即条件$l_1[k-1] < l_2[k-1]$可确保不产生重复的项集
    - 生成$\{l_1[1], l_1[2], \ldots., \boldsymbol{l_1[k-1], l_2[k-1]}\}$放入$\boldsymbol{C_k}$中

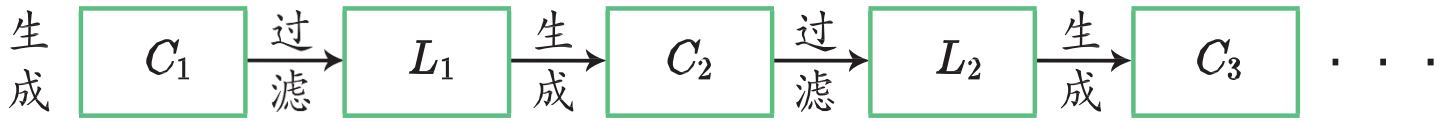
$$L_3 = \{abc, abd, acd, ace, bcd\} \xrightarrow{L_3 \bowtie L_3} \{abc\}\,\{\text{abd}\}\ \{acd\}\,\{ace\} \longrightarrow C_4 = \{abcd, \text{acde}\}$$

APriori算法：剪枝步

- A-Priori算法步骤2：剪枝步
  - 输入：候选K项集C_k
  - 输出：所有频繁K项集L_k
  - 过程：计算C_k中每个候选项集的支持度，从而确定L_k
    - 然而C_k可能很大，这样所涉及的计算量就很大
    - 为了提高效率，可利用反单调性：**任何非频繁的(k-1)项集都不可能是频繁k项集的子集**。
    - 因此，若一个候选k项集的(k-1)项子集不在L_{k-1}中，则该候选也不可能是频繁的，从而可以从C_k中删除

$$\forall X, Y: X \subseteq Y \rightarrow support(X) \geq support(Y)$$

$L_3 = \{abc, abd, acd, ace, bcd\}$　　$C_4 = \{abcd, acde\}$　　$L_4 = \{abcd\}$



APriori算法实例

- A-Priori算法实例

【例】右图为某商店的用户购买记录，共有9个事务，A-Priori假定事务中的项按字典次序存放。

ID	事务
T100	l_1，l_2，l_5
T200	l_2，l_4
T300	l_2，l_3
T400	l_1，l_2，l_4
T500	l_1，l_3
T600	l_2，l_3
T700	l_1，l_3
T800	l_1，l_2，l_3，l_5
T900	l_1，l_2，l_3

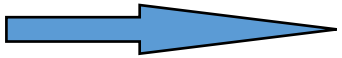
APriori算法实例

- A-Priori算法实例

(1) 在算法的第一次迭代，每个项都是候选1项集的集合C_1的成员。算法简单地扫描所有的事务，对每个项的出现次数计数

ID	事务
T100	l_1，l_2，l_5
T200	l_2，l_4
T300	l_2，l_3
T400	l_1，l_2，l_4
T500	l_1，l_3
T600	l_2，l_3
T700	l_1，l_3
T800	l_1，l_2，l_3，l_5
T900	l_1，l_2，l_3

扫描数据集,对每个候选1项集计算支持度



C_1	支持度
$\{l_1\}$	6
$\{l_2\}$	7
$\{l_3\}$	6
$\{l_4\}$	2
$\{l_5\}$	2

APriori算法实例

- A-Priori算法实例

  (2) 设最小支持度计数=**2**，可以确定频繁1项集的集合L_1

C_1	支持度
$\{l_1\}$	6
$\{l_2\}$	7
$\{l_3\}$	6
$\{l_4\}$	2
$\{l_5\}$	2

比较候选项集支持度与最小支持度阈值

L_1	支持度
$\{l_1\}$	6
$\{l_2\}$	7
$\{l_3\}$	6
$\{l_4\}$	2
$\{l_5\}$	2

APriori算法实例

- A-Priori算法实例

  (3) 使用$L_1 \bowtie L_1$产生候选2项集的集合C_2

L_1	支持度
$\{l_1\}$	6
$\{l_2\}$	7
$\{l_3\}$	6
$\{l_4\}$	2
$\{l_5\}$	2

由L_1产生候选2项集

C_2
$\{l_1, l_2\}$
$\{l_1, l_3\}$
$\{l_1, l_4\}$
$\{l_1, l_5\}$
$\{l_2, l_3\}$
$\{l_2, l_4\}$
$\{l_2, l_5\}$
$\{l_3, l_4\}$
$\{l_3, l_5\}$
$\{l_4, l_5\}$

APriori算法实例

- A-Priori算法实例

  (4) 扫描数据集，计算C_2中每个候选项集的支持度

ID	事务
T100	l_1, l_2, l_5
T200	l_2, l_4
T300	l_2, l_3
T400	l_1, l_2, l_4
T500	l_1, l_3
T600	l_2, l_3
T700	l_1, l_3
T800	l_1, l_2, l_3, l_5
T900	l_1, l_2, l_3

对每个候选2项集计算支持度

C_2	支持度
$\{l_1, l_2\}$	4
$\{l_1, l_3\}$	4
$\{l_1, l_4\}$	1
$\{l_1, l_5\}$	2
$\{l_2, l_3\}$	4
$\{l_2, l_4\}$	2
$\{l_2, l_5\}$	2
$\{l_3, l_4\}$	0
$\{l_3, l_5\}$	1
$\{l_4, l_5\}$	0

APriori算法实例

- A-Priori算法实例

  (5)最小支持度计数=**2**，确定频繁2项集的集合L_2

比较候选项集支持度
与最小支持度阈值

L_2	支持度
$\{l_1, l_2\}$	4
$\{l_1, l_3\}$	4
$\{l_1, l_5\}$	2
$\{l_2, l_3\}$	4
$\{l_2, l_4\}$	2
$\{l_2, l_5\}$	2

APriori算法实例

L_2	支持度
$\{l_1, l_2\}$	4
$\{l_1, l_3\}$	4
$\{l_1, l_5\}$	2
$\{l_2, l_3\}$	4
$\{l_2, l_4\}$	2
$\{l_2, l_5\}$	2

- A-Priori算法实例

(6) 使用$L_2 \bowtie L_2$产生候选3项集的集合C_3

①连接步： $C_3 = L_2 \bowtie L_2$

$= \{\{l_1, l_2\}, \{l_1, l_3\}, \{l_1, l_5\}, \{l_2, l_3\}, \{l_2, l_4\}, \{l_2, l_5\}\}$

$\bowtie$

$\{\{l_1, l_2\}, \{l_1, l_3\}, \{l_1, l_5\}, \{l_2, l_3\}, \{l_2, l_4\}, \{l_2, l_5\}\}$

$= \{\{l_1, l_2, l_3\}, \{l_1, l_2, l_5\}, \{l_1, l_3, l_5\},$
$\{l_2, l_3, l_4\}, \{l_2, l_3, l_5\}, \{l_2, l_4, l_5\}\}$

APriori算法实例

L_2	支持度
$\{l_1, l_2\}$	4
$\{l_1, l_3\}$	4
$\{l_1, l_5\}$	2
$\{l_2, l_3\}$	4
$\{l_2, l_4\}$	2
$\{l_2, l_5\}$	2

- A-Priori算法实例

(6) 使用$L_2 \bowtie L_2$产生候选3项集的集合C_3

②剪枝步：反单调性：频繁项集的所有子集必须是频繁的

$\{\{l_1, l_2, l_3\}, \{l_1, l_2, l_5\}, \{l_1, l_3, l_5\}, \{l_2, l_3, l_4\}, \{l_2, l_3, l_5\}, \{l_2, l_4, l_5\}\}$

- $\{l_1, l_2, l_3\}$的2项子集是$\{l_1, l_2\}, \{l_1, l_3\}$和$\{l_2, l_3\}$
  它们都是L_2的元素。因此保留$\{l_1, l_2, l_3\}$在C_3中
- $\{l_1, l_3, l_5\}$的2项子集是$\{l_1, l_3\}, \{l_1, l_5\}$和$\{l_3, l_5\}$

  $\{l_3, l_5\}$不是L_2的元素，因而不是频繁的，由C_3中删除$\{l_1, l_3, l_5\}$
- 以此类推筛选得到C_3

C_3
$\{l_1, l_2, l_3\}$
$\{l_1, l_2, l_5\}$

APriori算法实例

- A-Priori算法实例

  (7) 扫描数据集，计算C_3中每个候选项集的支持度

ID	事务
T100	l_1，l_2，l_5
T200	l_2，l_4
T300	l_2，l_3
T400	l_1，l_2，l_4
T500	l_1，l_3
T600	l_2，l_3
T700	l_1，l_3
T800	l_1，l_2，l_3，l_5
T900	l_1，l_2，l_3

对每个候选3项集
计算支持度

C_3	支持度
$\{l_1, l_2, l_3\}$	2
$\{l_1, l_2, l_5\}$	2

APriori算法实例

- A-Priori算法实例

  (8)最小支持度计数=**2**，确定频繁3项集的集合L_3

比较候选项集支持度
与最小支持度阈值

L_3	支持度
$\{l_1, l_2, l_3\}$	2
$\{l_1, l_2, l_5\}$	2

APriori算法实例

ID	事务
T100	l_1, l_2, l_5
T200	l_2, l_4
T300	l_2, l_3
T400	l_1, l_2, l_4
T500	l_1, l_3
T600	l_2, l_3
T700	l_1, l_3
T800	l_1, l_2, l_3, l_5
T900	l_1, l_2, l_3

- A-Priori算法实例

(9) 使用$L_3 \bowtie L_3$产生候选4项集的集合C_4，尽管{ }，这个项集被剪去，因为它的子集$\{l_2, l_3, l_5\}$不是频 ，算法终止，找出了所有的频繁项集如下

L_1	支持度
$\{l_1\}$	6
$\{l_2\}$	7
$\{l_3\}$	6
$\{l_4\}$	2
$\{l_5\}$	2

L_2	支持度
$\{l_1, l_2\}$	4
$\{l_1, l_3\}$	4
$\{l_1, l_5\}$	2
$\{l_2, l_3\}$	4
$\{l_2, l_4\}$	2
$\{l_2, l_5\}$	2

L_3	支持度
$\{l_1, l_2, l_3\}$	2
$\{l_1, l_2, l_5\}$	2

APriori算法

- APriori算法
  - **总结：**APriori算法适合用在数据集稀疏，频繁模式较短，支持度较高的场景中
  - **不足：**难以适用于稠密数据和长频繁模式
    - 可能产生大量的候选集
    - 可能需要重复扫描数据集多次
  - 改进方法**（课后学习）**
    - DHP算法
    - Partition算法
    - Sample算法
    - DIC算法

关联规则挖掘

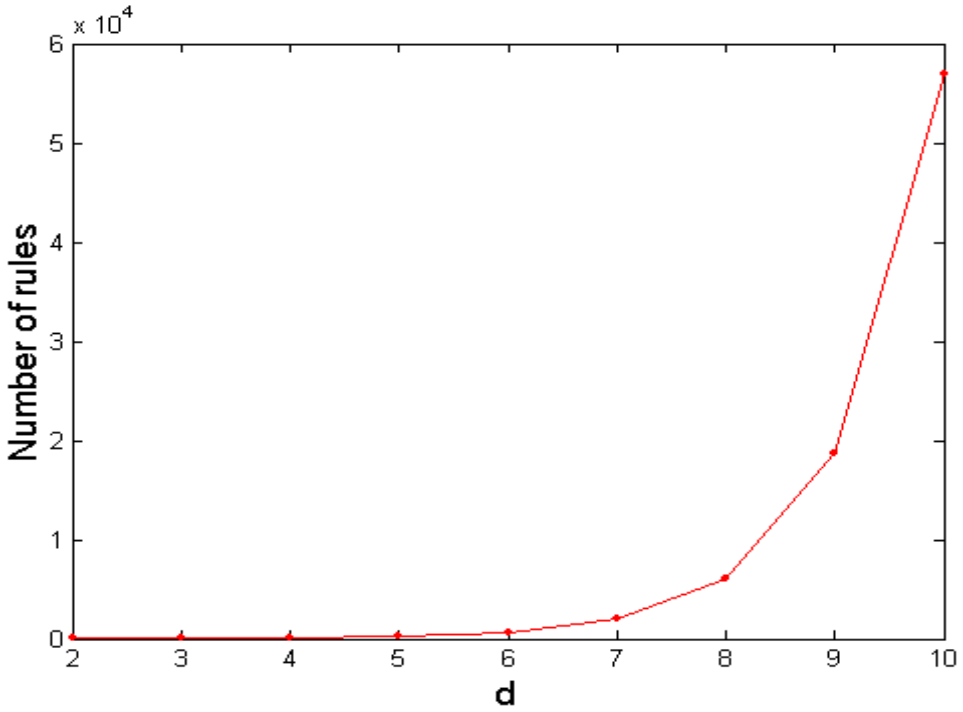
- 关联规则挖掘的第二步：如何从频繁项集中生成规则?
  - 任务：给定一个频繁项集L, 寻找所有非空子集$f \subset L$ 使得$f \rightarrow L - f$ 满足置信度要求
    - 若{A,B,C,D}是频繁项集, 候选规则有14种:

      ABC →D, ABD →C, ACD →B, BCD →A,
      A →BCD, B →ACD, C →ABD, D →ABC
      AB →CD, AC → BD, AD → BC, BC →AD,
      BD →AC, CD →AB,

  - **若$|L| = k$, 则有$2^k - 2$ 种候选的关联规则**(忽略$L \rightarrow \emptyset$和$\emptyset \rightarrow L$)

关联规则生成

- 关联规则生成(Rule Generation)——计算复杂度
  - 对于d个项目:
    - 候选项集数= 2^d
    - 可能规则数R = $3^d - 2^{d+1} + 1$



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

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

If d=6, R = 602 rules

关联规则生成

- 关联规则生成(Rule Generation)
  - 如何高效地从频繁项集中生成规则?
  - 一般而言，置信度不满足反单调性

    $confidence$(ABC $\rightarrow$ D) 可能大于或小于$confidence$(AB $\rightarrow$ D)

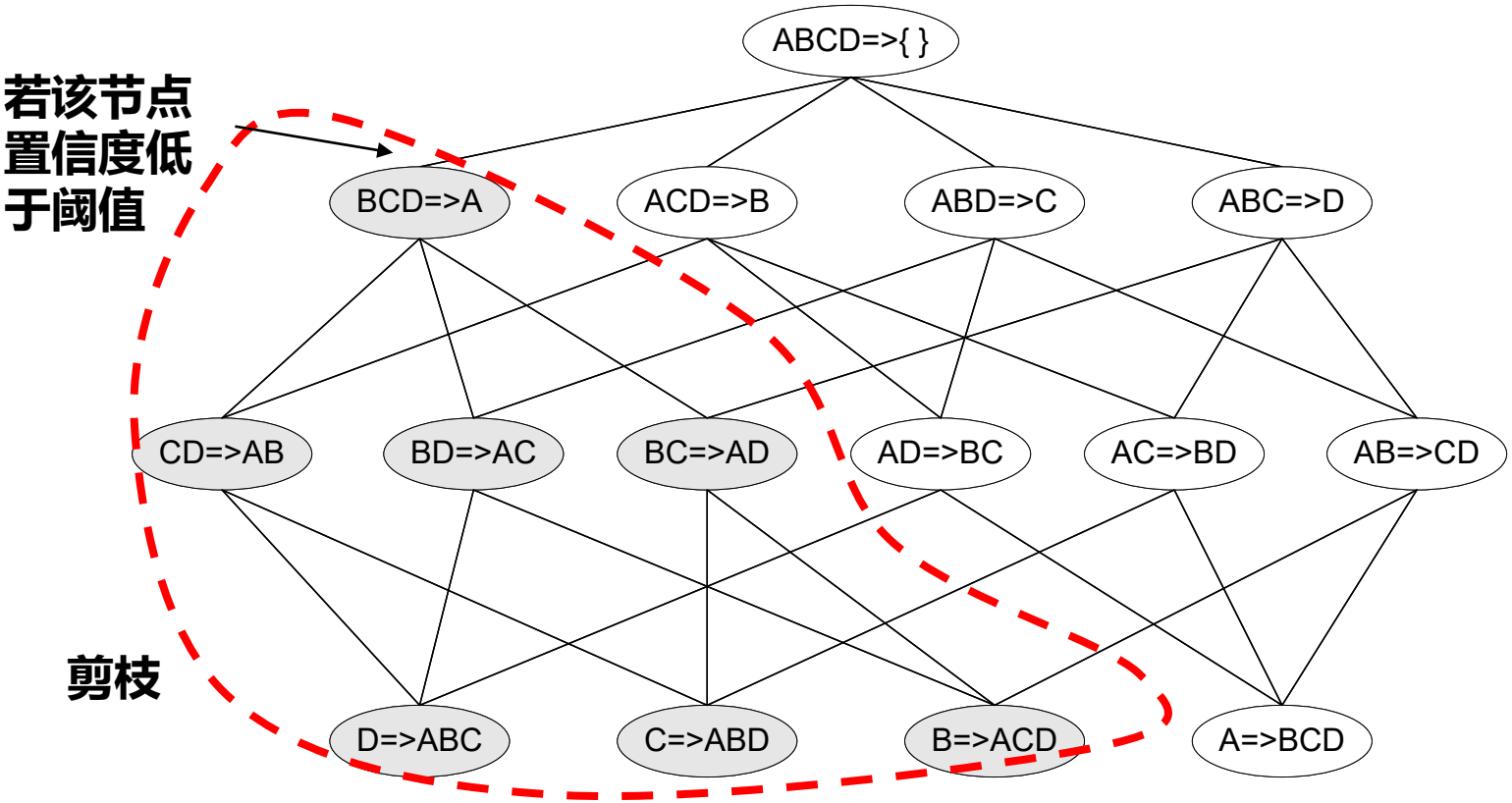
  - 但从同一项集生成的规则满足反单调性
    - 例：L = {A,B,C,D}

$$confidence(ABC \rightarrow D) \geq confidence(AB \rightarrow CD) \geq confidence(A \rightarrow BCD)$$

为什么？课后验证一下

关联规则生成

- 关联规则生成
  - 对某个频繁项集，自顶向下生成候选规则
  - 若某个父节点置信度较低，其所有子节点无需再判断



FP-Growth

- 频繁项集挖掘——FP-Growth算法
  - 关联规则挖掘的经典算法之一，于2000年由韩家炜等提出
  - 核心思想：能够压缩原始数据的频繁模式树(Frequent Pattern Tree，FP-tree)

Mining frequent patterns without candidate generation

J Han, J Pei, Y Yin - ACM sigmod record, 2000 - dl.acm.org

Mining frequent patterns in transaction databases, time-series databases, and many other kinds of databases has been studied popularly in data mining research. Most of the previous studies adopt an Apriori-like candidate set generation-and-test approach. However, candidate set generation is still costly, especially when there exist prolific patterns and/or long patterns. In this study, we propose a novel frequent pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about …

☆ 保存 99 引用 被引用次数：10079 相关文章 所有 64 个版本

https://hanj.cs.illinois.edu/

关联规则挖掘前沿：课后学习

- 多维关联规则挖掘
  - 多维的关联规则，如{购买: 电脑}∧ {年龄∈ [20,30]} → {购买:手机}
- 多层关联规则挖掘
  - {光明牛奶，全麦面包} 的支持度低，抽象化为{牛奶，面包}等高层概念
- 稀有模式挖掘
  - 金融安全领域：普通的交易行为，非正常交易（欺诈交易）
- 负模式挖掘
  - {可口可乐，百事可乐} 支持度高，但 {可口可乐}→ ¬{百事可乐}
- 序列模式挖掘算法
  - 加入事务发生的时间：用户购买顺序的关联分析

Lei Zhang, Ping Luo, Linpeng Tang, Enhong Chen, Qi Liu, Min Wang, Hui Xiong, Occupancy-based Frequent Pattern Mining, ACM Transactions on Knowledge Discovery from Data.

数据挖掘基础

- 数据挖掘定义、四类任务及其应用场景
- 聚类任务
  - 无监督：K-Means、DBSCAN、评估方法
- 分类任务
  - 有监督：决策树、K近邻、感知机/SVM、集成分类器、评估方法
- 关联分析
  - 支持度和置信度、Apriori算法
- 异常检测

Tips: 结合实际场景，分析问题，设计模型，评估结果。