北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Data Analytics with R
R语言数据分析

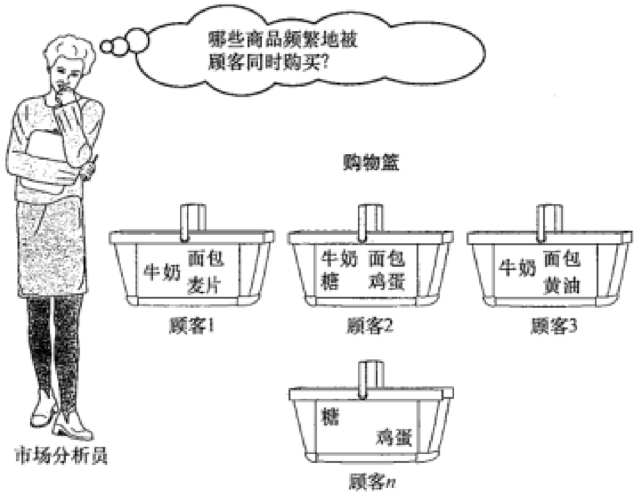
相随相伴、谓之关联

艾新波 / 2018 • 北京

课程体系

- R语言数据分析
  - 上部：论道 
    - 第1章 气象万千、数以等观
    - 第2章 所谓学习、归类而已
    - 第3章 格言联璧话学习
    - 第4章 源于数学、归于工程
  - 中部：执具 
    - 第5章 工欲善其事必先利其器
    - 第6章 基础编程
    - 第7章 数据对象
  -
    - 第8章 人人都爱tidyverse
    - 第9章 最美不过数据框
  - 下部 博术 
    - 第10章 观数以形
    - 第11章 相随相伴、谓之关联
    - 第12章 既是世间法、自当有分别
    - 第13章 方以类聚、物以群分
    - 第14章 庐山烟雨浙江潮

购物篮分析



TID	项集	黄油	鸡蛋	麦片	面包	牛奶	糖
1	{ 面包, 牛奶, 麦片 }	0	0	1	1	1	0
2	{ 牛奶, 面包, 糖, 鸡蛋 }	0	1	0	1	1	1
3	{ 牛奶, 面包, 黄油 }	1	0	0	1	1	0
n	{ 糖, 鸡蛋 }	0	1	0	0	0	1

图片来自韩家炜等《数据挖掘：概念与技术》，机械工业出版社，2016年，pp.158

基本概念：项、项集、事务

TID	项集	黄油	鸡蛋	麦片	面包	牛奶	糖
1	{ 面包, 牛奶,麦片 }	0	0	1	1	1	0
2	{ 牛奶, 面包, 糖, 鸡蛋 }	0	1	0	1	1	1
3	{ 牛奶, 面包, 黄油 }	1	0	0	1	1	0
n	{ 糖, 鸡蛋 }	0	1	0	0	0	1

$I = \{i_1, i_2, \cdots, i_d\}$购物篮数据中所有项的集合

$T = \{t_1, t_2, \cdots, t_N\}$所有事务的集合

每个事务t_i包含的项集都是I的子集

基本概念：关联分析、频繁项集

关联分析（Association Analysis）用于发现隐藏在大型数据集中有意义的联系，所发现的联系可以用频繁项集或关联规则的形式表示

- **项item的集合称为项集itemset**
- **包含k个项的项集称为k-项集**
- **项集出现的频度是包含该项集的事务数，简称为项集的频度、支持度计数**
- **如果项集的支持度满足预定义的最小支持度阈值，称之为频繁项集frequent itemset**

基本概念：关联规则

蕴涵式
Implication

关联规则是一个蕴涵式：$A \Rightarrow B$，其中A和B是不相交的项集

具体含义是：A出现的时候，B也出现；或者说，B伴随着A出现

关联规则在事务集T中成立，所具有支持度和置信度：

支持度：$support(A \Rightarrow B) = P(A \cup B)$

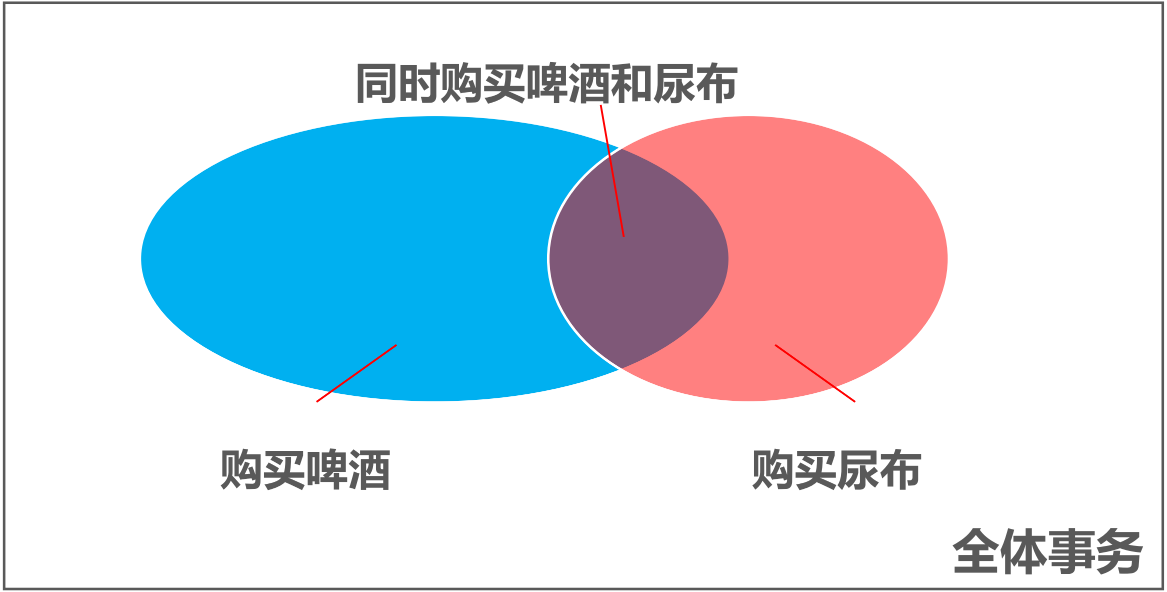
置信度：$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}$

支持度：**减少偶然性**

置信度：**增加推断能力**

满足最小支持度和置信度的规则称为强规则

支持度和置信度



关联规则挖掘

一旦获得A、B和A∪B的支持度

显然则很容易导出对应的关联规则 A⇒B 或 B ⇒ A，并且检查他们是否属于强规则

关联规则实际上是频繁项集分成左右两个子集的故事！！

换言之，关联规则的挖掘可以分为两个步骤：

（1）找出所有频繁项集，满足最小支持度

（2）由频繁项集产生强关联规则，满足最小置信度

Apriori算法原理

youdao

中英 apriori

apriori

英 美 [ˌɑpriˈɔri; ˌɑpriˈori; ˌepraɪˈɔrai; ˌɑpriˈorˌai]

adj. 先验的；推测的

adv. 自原因推及结果地

拉丁语a priori (“from the earlier”)和a posteriori ("from the latter")

A priori knowledge or justification is independent of experience

Apriori算法原理

一个包含k个项的数据集可能产生2^k-1个频繁项集

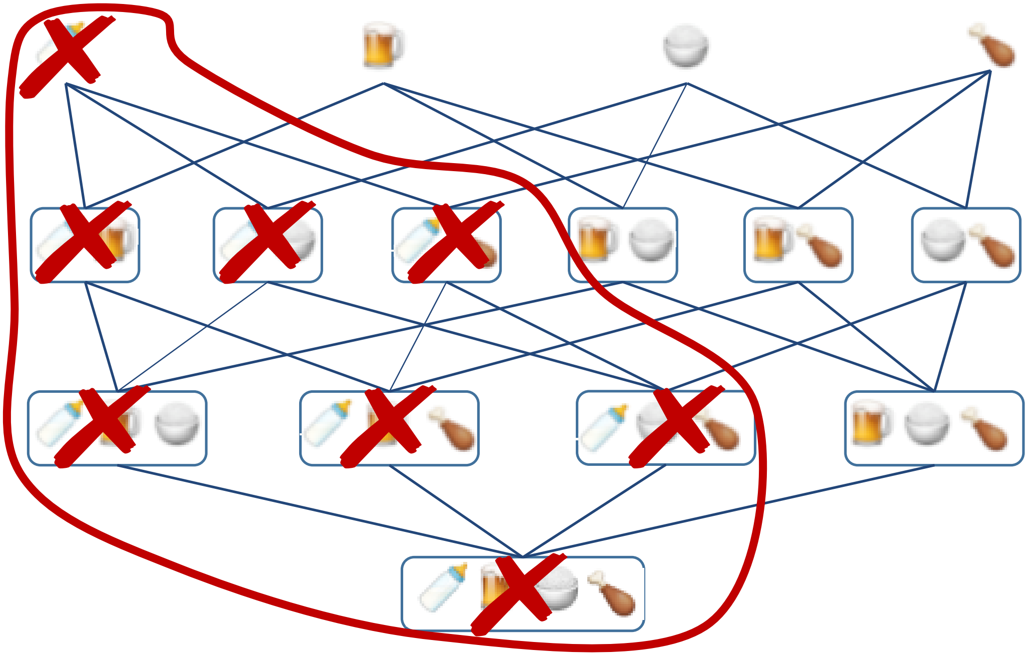
野蛮搜索的话，频繁项集的搜索空间是指数规模的

幸好我们有以下先验性质**Apriori** **Property**：

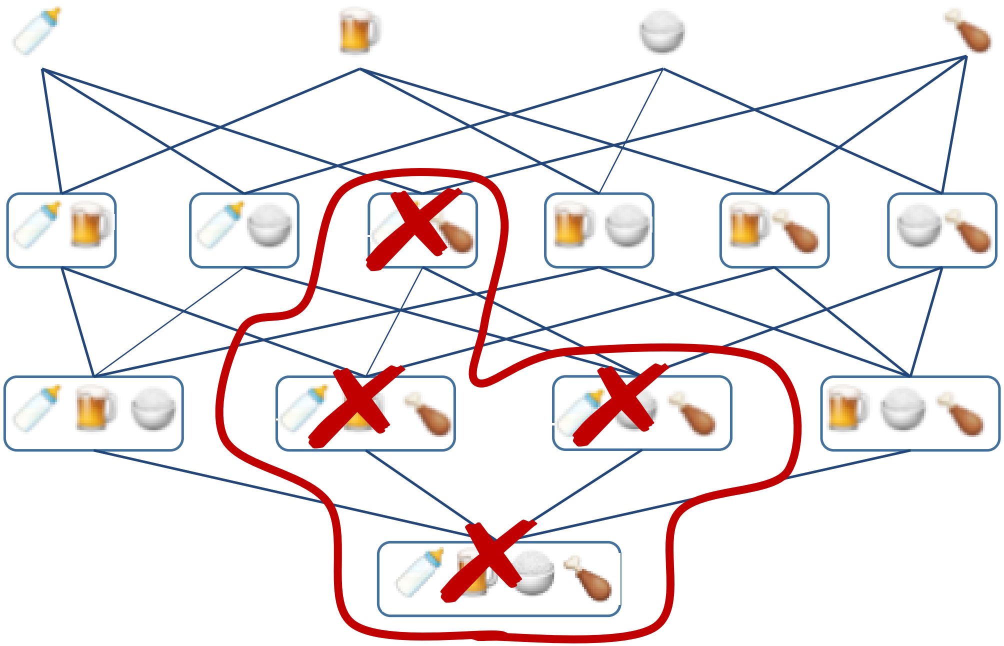
- 频繁项集的所有非空子集也一定是频繁的
- 非频繁项集的超集必定是非频繁的

每一个项集的支持度不会超过它的子集的支持度

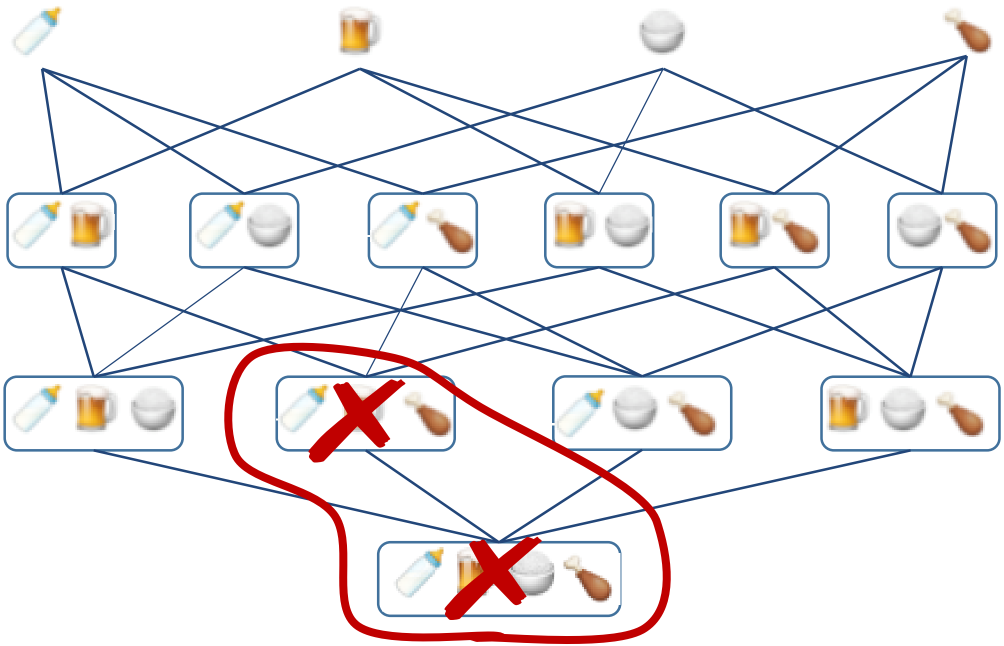
不同于蛮力搜索：基于支持度的候选集剪枝



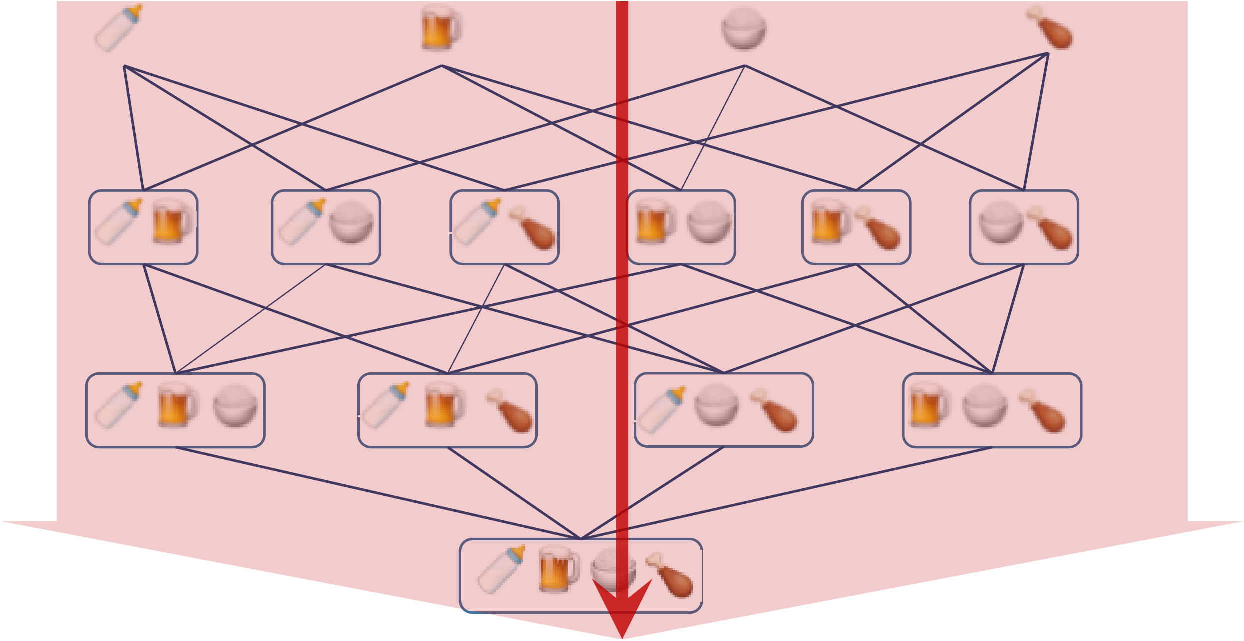
不同于蛮力搜索：基于支持度的候选集剪枝



不同于蛮力搜索：基于支持度的候选集剪枝



不同于蛮力搜索：基于支持度的候选集剪枝



Apriori算法：频繁项集的产生

Apriori算法：	频繁项集的产生

$k = 1$

$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N \times minsup\}$ #发现所有的频繁1-项集

Repeat

    $k = k + 1$

    $C_k = apriori - gen(F_{k-1})$ #产生候选集

    for每个事务$t \in T$ do

        $C_t = subset(C_k, t)$ #识别属于t的所有候选集

        for每个候选项集$c \in C_t$ do

            $\sigma(c) = \sigma(c) + 1$ #支持度计数增加

        end for

    end for

    $F_k = \{c | c \in C_k \wedge \sigma(c) \geq N \times minsup\}$

Until $F_k = \emptyset$

Result = $\cup F_k$

Apriori算法：候选集的产生

方法名称	方法描述
蛮力方法	把所有的k-项集都看做可能的候选集，然后再剪枝
$F_{k-1} \times F_1$方法	用其他频繁项来扩展每个频繁(k-1)-项集
$F_{k-1} \times F_{k-1}$方法	合并一对(k-1)-项集，当且仅当他们的前k-2项都相同

Apriori算法：候选集的产生

Apriori算法：apriori_gen(F_{k-1}) #产生候选集

For each 项集 $l_1 \in F_{k-1}$

  For each 项集 $l_2 \in F_{k-1}$

    if$((l_1[1] = l_2[1]) \wedge \cdots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1]))$

      then$c = l_1 \bowtie l_2$

      if has_infrequent_subset(c, F_{k-1})

        delete c #删除非频繁的候选

      else add c to C_k

Result = C_k

Apriori算法：has_infrequent_subset(c, F_{k-1})

For each $(k-1)$ subset s of c

  if $s \notin F_{k-1}$

    return TRUE

End for

return FALSE

Apriori算法：候选集产生过程示意图

设有如下事务数据：

TID	商品列表	TID	商品列表
T1	I1, I2, I5	T6	I2, I3
T2	I2, I4	T7	I1, I3
T3	I2, I3	T8	I1, I2, I3, I5
T4	I1, I2, I4	T9	I1, I2, I3
T5	I1, I3		

求最小支持度计数为 2 的频繁项集！

Apriori算法：候选集产生过程示意图

TID	商品列表	TID	商品列表
T1	I1, I2, I5	T6	I2, I3
T2	I2, I4	T7	I1, I3
T3	I2, I3	T8	I1, I2, I3, I5
T4	I1, I2, I4	T9	I1, I2, I3
T5	I1, I3		

$k = 1$

项集	计数	是否频繁
{I1}	6	√
{I2}	7	√
{I3}	6	√
{I4}	2	√
{I5}	2	√

求最小支持度计数为 2 的频繁项集！

Apriori算法：候选集产生过程示意图

TID	商品列表	TID	商品列表
T1	I1, I2, I5	T6	I2, I3
T2	I2, I4	T7	I1, I3
T3	I2, I3	T8	I1, I2, I3, I5
T4	I1, I2, I4	T9	I1, I2, I3
T5	I1, I3		

求最小支持度计数为 2 的频繁项集！

$k = 1 \rightarrow k = 2$

频繁K项集	候选K+1项集
{I1} {I2} {I3} {I4} {I5}	{I1, I2} {I1, I3} {I1, I4} {I1, I5} {I2, I3} {I2, I4} {I2, I5} {I3, I4} {I3, I5} {I4, I5}

Apriori算法：候选集产生过程示意图

TID	商品列表	TID	商品列表
T1	I1, I2, I5	T6	I2, I3
T2	I2, I4	T7	I1, I3
T3	I2, I3	T8	I1, I2, I3, I5
T4	I1, I2, I4	T9	I1, I2, I3
T5	I1, I3		

$k = 1$

项集	计数	是否频繁
{I1, I2}	4	√
{I1, I3}	4	√
{I1, I4}	1	×
{I1, I5}	2	√
{I2, I3}	4	√
{I2, I4}	2	√
{I2, I5}	2	√
{I3, I4}	0	×
{I3, I5}	1	×
{I4, I5}	0	×

求最小支持度计数为 2 的频繁项集！

Apriori算法：候选集产生过程示意图

TID	商品列表	TID	商品列表
T1	I1, I2, I5	T6	I2, I3
T2	I2, I4	T7	I1, I3
T3	I2, I3	T8	I1, I2, I3, I5
T4	I1, I2, I4	T9	I1, I2, I3
T5	I1, I3		

$k = 2 \rightarrow k = 3$

频繁K项集	候选K+1项集
{I1, I2} {I1, I3} {I1, I5} {I2, I3} {I2, I4} {I2, I5}	{I1, I2, I3} {I1, I2, I5} ~~{I1, I3, I5}~~ ~~{I2, I3, I4}~~ ~~{I2, I3, I5}~~ ~~{I2, I4, I5}~~

求最小支持度计数为 2 的频繁项集！

Apriori算法：候选集产生过程示意图

TID	商品列表	TID	商品列表
T1	I1, I2, I5	T6	I2, I3
T2	I2, I4	T7	I1, I3
T3	I2, I3	T8	I1, I2, I3, I5
T4	I1, I2, I4	T9	I1, I2, I3
T5	I1, I3		

$k = 3$

项集	计数	是否频繁
{I1, I2, I3}	2	√
{I1, I2, I5}	2	√

至此，算法终止

找出了所有的频繁项集

求最小支持度计数为 2 的频繁项集！

Apriori算法：由候选集产生关联规则

关联规则的挖掘可以分为两个步骤：

（1）找出所有频繁项集，满足最小支持度

（2）由频繁项集产生强关联规则，满足最小置信度

避免野蛮搜索：

（1）基于支持度对候选集进行剪枝

（2）基于置信度对规则进行剪枝

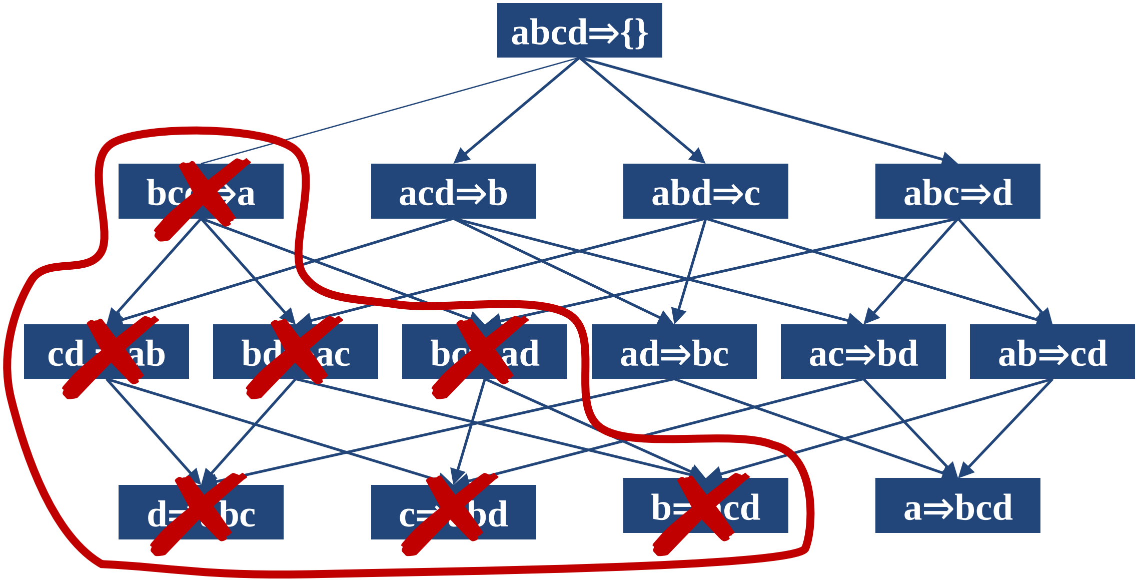
Apriori算法：由候选集产生关联规则

如果规则$X \Rightarrow (Y-X)$不满足置信度阈值，则对于$X' \subset X$，$X' \Rightarrow (Y-X')$的规则也一定不满足置信度阈值

$$confidence(X' \Rightarrow Y - X') = \frac{support(Y)}{support(X')}$$

$$\leq \frac{support(Y)}{support(X)} < min\_conf$$

Apriori算法：基于置信度的规则剪枝



一言以蔽之

Apriori算法是平凡的：搜索、匹配、计数、求比例、…

Apriori算法是美好的：支持度剪枝、置信度剪枝、……

谢谢聆听
Thank you

教师个人联系方式

艾新波

手机：13641159546

QQ：23127789

微信：13641159546

E-mail: 13641159546@126.com

axb@bupt.edu.cn

地址：北京邮电大学科研楼917室

课程网址：https://github.com/byaxb/RDataAnalytics