# Apriori

一个挖掘关联规则的频繁项集算法

1. 一些术语/定义：（以购物篮数据为例）

关联规则（假设）：牛奶→鸡蛋【支持度=2%，置信度=60%】

支持度：所有购物篮中既购买牛奶又购买鸡蛋的购物篮占比

置信度：购买了牛奶的人中60%也购买了鸡蛋

最小支持度阈值：太低则认为数据不够支持这个关联规则

最小置信度阈值：太低则认为关联规则不能令人信服

p.s.：最小支持度阈值和最小置信度阈值由数据挖掘者/专家设定

项集：商品集

k-项集：k个商品组成的集合

频繁项集：满足最小支持度的项集，频繁k-项集一般记为L\_k

强关联规则：满足最小支持度阈值和最小置信度阈值的关联规则

1. 基础算法：（python/apriori.py）

①产生频繁项集：

设定最小支持度→1-项集→扫描数据库计算支持度得到频繁1-项集→

连接频繁1-项集得到2-项集并剪枝→扫描数据库计算支持度得到频繁2-项集→......

（连接步）在频繁(k-1)-项集中选2个进行连接得到k-项集

p.s.：连接方式：按照项集顺序遍历，只连接前(k-2)-项相等且最后一个位置

上不等的

（剪枝步）利用先验性质：频繁项集的所有子集必须是频繁的

扫描是否产生的k-项集所有(k-1)-子项集在频繁(k-1)项集中

p.s.：如{1,2,3}要在频繁3-项集中必须要求{1,2}{1,3}{2,3}均在频繁2-项集中，

否则无需考虑，因为其支持度一定小于最小支持度

p.s.：扫描数据库是必须的，因为即使一个k-项集的所有(k-1)-子项集都是频繁的，这个

k-项集本身也不一定是频繁的，如最小支持度为2，数据为{1,2}{1,3}{2,3}{1,2,3}， {1,2,3}的所有2-子项集都是频繁的，但是{1,2,3}本身不是频繁的

* 详细例子见韩家炜书p161例6.3

②由频繁项集产生强关联规则：

* 如{1,2,3}是一个频繁项集，则有{1,2}→{3}，{1,3}→{2}，{2,3}→{1}，{1}→{2,3}，{2}→{1,3}，{3}→{1,2}这几种可能，然后验证最小置信阈值是否满足即可

p.s.：由于规则由频繁项集产生，因此每个规则都自动满足最小支持度

1. R内置函数：

install.packages("arules")

library(arules)#这个包中有内置数据集：Groceries，购物篮

data(Groceries)#载入数据文件

inspect(Groceries)#观看数据集里的数据

rules=apriori(Groceries,parameter=list(support=最小支持度,confidence=最小置信度))

inspect(rules)#可以观察产生的规则，其中lift=P(L,R)/(P(L)P(R))，当L,R独立时=1，越大越能说明他们在同一个购物篮中不是一个偶然现象

1. 应用：

推荐宝贝/视频

1. 优点：

简单，易理解，数据要求低

1. 缺点

I/O负载大，会产生过多候选项目集，每次计算项集支持度时都对数据库中的全部记录进行了一遍扫描比较

**改进1：FP-growth**

**http://www.cnblogs.com/qwertWZ/p/4510857.html#\_label4**

**改进2：Hash技术**

**韩家炜pdf189**