**数据挖掘**

**--作业二实验报告**

学 院：自动化学院

专 业：系统工程

年 级：2017 级

姓 名：胡笛

学 号：2620170029

提交日期：2018/4/22

1. 实验背景

随着社会进入信息化时代，现代化的企业搜集了大量数据或高维数据，包括市场、客户、供货商、竞争对手以及未来趋势等重要信息，但是信息超载与无结构化，使得企业决策部门无法有效利用现存的信息，甚至使决策行为产生混乱与误用。通过数据挖掘技术，可以从大量的数据中，挖掘出不同的信息与知识来支持决策，必能产生企业的竞争优势。

数据挖掘和知识发现是一个涉及多学科的研究领域。数据库技术、人工智能、机器学习、统计学、粗糙集、模糊集、神经网络、模式识别、知识库系统、高性能计算、数据可视化等均与数据挖掘相关。近年来，与数据库的知识发现研究领域已经成为热点，其中关联规则数据挖掘算法是数据挖掘中的一个很重要的课题，它是从背后发现数据中的关联或联系。

本实验首先对数据集进行处理，转换成适合关联规则挖掘的形式；找出频繁项集；导出关联规则，计算其支持度和置信度对规则进行评价。

2.实验目的

掌握数据挖掘的基础知识，能够深刻理解并熟练运用Apriori算法，能够熟练Matlab软件进行算法的实现。

根据题目给定的原始事务记录和最小支持度，通过迭代的方法求出各项频繁项目集。

根据求得频的繁项目集和给定的最小可信度，求出相关的关联规则。

3.操作环境

系统环境：Windows 10

软件环境：matlab2017a

软件简介：MATLAB是美国[MathWorks](https://baike.baidu.com/item/MathWorks" \t "_blank)公司出品的商业[数学软件](https://baike.baidu.com/item/%E6%95%B0%E5%AD%A6%E8%BD%AF%E4%BB%B6" \t "_blank)，用于算法开发、数据可视化、数据分析以及[数值计算](https://baike.baidu.com/item/%E6%95%B0%E5%80%BC%E8%AE%A1%E7%AE%97)的高级技术计算语言和交互式环境，MATLAB可以进行[矩阵](https://baike.baidu.com/item/%E7%9F%A9%E9%98%B5" \t "_blank)运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等，主要应用于工程计算、控制设计、[信号处理](https://baike.baidu.com/item/%E4%BF%A1%E5%8F%B7%E5%A4%84%E7%90%86)与通讯、[图像处理](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86)、[信号检测](https://baike.baidu.com/item/%E4%BF%A1%E5%8F%B7%E6%A3%80%E6%B5%8B)、[金融建模](https://baike.baidu.com/item/%E9%87%91%E8%9E%8D%E5%BB%BA%E6%A8%A1)设计与分析等领域。

4.实验数据

本实验选用的的是实验一中的数据集1，对其数据的标称属性进行关联规则挖掘。

5.关联规则挖掘

5.1 Apriori算法原理

Apriori算法是一种最有影响的挖掘布尔关联规则频繁项集的算法。很多的的挖掘算法是在Apriori算法的基础上进行改进的，比如基于散列（Hash）的方法，基于数据分割（Partition）的方法以及不产生候选项集的FP-GROWTH方法等。因此要了解关联规则算法不得不先要了解Apriori算法。

Apriori算法使用一种称作逐层迭代的候选产生测试（candidate generation and test）的方法，k-项目集用于探索（k+1）-项目集。首先，找出频繁1-项目集的集合，该集合记作L 。L 用于找频繁2-向募集到集合L ，而L 用于找L ，如此下去，直到不能找到频繁k-项目集。找每一个L 均需要一次数据库的扫描。

Apriori性质：频繁项集的所有非空子集必须也是频繁的。Apriori性质基于如下观察：根据定义，如果项集I不满足最小支持度阈值，则I不是频繁的，即support(I)＜min-sup。如果项A添加到I，则结果项集（即IUA）不可能比I更频繁出现。因此，IUA也不是频繁的，即support(IUA)＜min-sup。

算法应用Apriori性质以LK-1来找LK，这一过程由连接和剪枝组成。

    C ：Candidate itemset of size k，即k-候选项目集。

    L ：frequent itemset of size k，即k-频繁项目集。

连接步：为找L ，通过L 与自己连接产生候选k-项集的集合。该候选项集记作C 。设 和 是L 中的项集。记 [j]表示 的第j项（例如， [k-2]表示的倒数第3项）。为方便计，假定事务或项集中的项按字典次序排列。执行LK-1与自身连接，其中，LK-1的元素是可连接的，如果（ [1]= [1]）∧（ [2]= [2]）∧…∧（ [k-2]= [k-2]）∧（ [k-1]＜ [k-1]）。条件（ [k-1]＜ [k-1]）是简单地保证不产生重复。连接 和 产生的结果项集是 [1] [2] … [k-1] [k-1]。

剪枝步：C 是L 的超集，即是，它的成员可以是频繁的也可以是不频繁的，但所有的频繁k-项集都包含在C 中。扫描数据库，确定C 中每个候选的计数，从而确定L （即根据定义，计数值不小于最小支持度计数的所有候选是频繁的，从而属于L ）。然而C 可能很大，这样所涉及的计算量就会很大。为压缩C ，可以用一下办法使用Apriori性质：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，如果一个候选k-项集的（k-1）子集不在L 中，则该候选也不可能是频繁的，从而可以由C 中删除。这种子集测试可以使用所有频繁项集的散列树快速完成。

算法首先扫描数据库，各数据项每出现一次，计数一次。由每个数据项的计数值可以算出其支持度。支持度超过阈值的1-itemset即为频繁1-itemset。产生所有单项频繁数据项集的集合L 。从L 出发，可以循环计算二项频繁项集L 、三项频繁项集L 、…L 、…。直到有某个r值使得L 为空，这时算法停止。这里在第k次循环中，即：从L 计算L 时，先用apriori\_gen()函数从L 生成候选的k项数据项集的集合C ，C 中的每一个项集是对两个只有一个项不同的属于L 的频繁项集做一个（k-2）-连接来产生的。C 中的项集是用来产生和频集的候选集，最后的频集L 必须是C 的一个子集。C 中的每个元素需在交易数据库中进行验证来决定其是否加入L 。

5.2查找频繁项目集

首先通过下面代码将数据文件导入matlab。

%% 主程序

min\_sup=**0.2**;

confidence=**30;**

L1=find\_frequent\_1\_itemset(stdata,min\_sup);

Ck=aproiri\_gen(L1,min\_sup);

把最小支持度设为0.2,进行频繁项集的筛选。

function [L1]=find\_frequent\_1\_itemset(stdata,min\_sup)

L1=stdata;

L1(1,:)=[];

L1(:,1)=[];

C1=L1;

for j=1:size(L1,2)

x=tabulate(L1(:,j));

x=sortrows(x,3,'descend'); %按从大到小排列

%找出最小支持度小于0.2，在x中的那一行 m

for i=1:size(x,1)

if x{i,3}<=min\_sup

m=i;

break

end

end

for xi=m:size(x,1)

V=[];

vj=1;

for ii=1:size(L1,1)

if strcmp(x{xi,1},L1{ii,j})

V(1,vj)=ii;

vj=vj+1;

end

end

L1(V,:)=[];

end

end

end

5.3连接和剪枝

function [Ck]=aproiri\_gen(L1,min\_sup)

%% 创建Ck

Ck={};

for i=1:size(L1,1)

t=1;

for j=1:size(L1,2)

for m=j+1:size(L1,2)

C2{i,t}={L1{i,j},L1{i,m}};

t=t+1;

end

end

end

%% 支持度计数 v

v=[];

for j=1:size(Ck,2)

vi=1;

for i=1:1:size(Ck,1)

support=0;

for t=1:size(Ck,1)

if strcmp(Ck{i,j},Ck{t,j})

support=support+1;

end

end

v(vi,j)=support;

vi=vi+1;

end

end

%计算支持度概率,放入support/ b的列是待删除的行

support=[];

support=100\*v./size(Ck,1);

b=[];

for supportj=1:size(support,2)

bi=1;

for supporti=1:size(support,1)

if support(supporti,supportj)<=min\_sup

b(bi,supportj)=supporti;

bi=bi+1;

end

end

end

c=b;

%删除

L2=Ck;

for j=1:size(b,2)

for i=1:size(b,1)

if b(i,j)>0

L2(b(i,j),:)=[];

b(:,j)=b(:,j)-1;

end

end

end

end

5.4挖掘结果

最终得到的结果如下图所示，我们可以清楚的看到在支持度为0.2%，可信度为30%的前提下，共有11项是符合关联规则的：

'otc alterations permit' ；'09/15/2017'；'Addison' ；'St'； 'soft story retrofit per sfebc chapter 4d engineering criteria 2016 cebc appendix a-4' ；'filed'； '09/15/2017' '09/15/2017' 'wood frame (5)' 'woodframe (5)' 'Glen Park'

