文章编号：2096-1472(2018)-03-20-03
DOI:10.19644/j.cnki.issn2096-1472.2018.03.005

关联规则挖掘在游戏视频销售中的研究与应用

闫东明，陈占芳，姜晓明
(长春理工大学计算机科学技术学院，吉林 长春 130022)

摘　要：数据挖掘是一项热门技术，该技术融合了数据库、统计学等领域知识，关联规则的挖掘则能找出商品销售中商品之间的联系。本文针对Apriori算法，及其改进算法FP-Growth进行了研究，对比了Apriori算法与FP-Growth算法的效率，得出FP-Growth算法由于只需要对数据进行一次扫描即可生成相应的数据集，使其生成数据集的整体效率要高于Apriori算法。

关键词：Apriori算法；数据挖掘；FP-Growth算法；关联规则；游戏销售
中图分类号：TP391　　**文献标识码**：A

Research and Application of Mining Association Rules in Game Sales

YAN Dongming,CHEN Zhanfang,JIANG Xiaoming
(*School of Computer Science and Technology,Changchun University of Science and Technology,Changchun* 130022,*China*)

Abstract: Data mining is a hot technology which comprises database,artificial intelligence,statistics,etc.The mining association rules can find out the relations among the selling commodities.This paper studies Apriori algorithm and its improved algorithm,FP-Growth,and compares the efficiency of them,where it is found that corresponding data set can be generated after only one data scanning based on FP-Growth algorithm,leading to higher overall efficiency of the generated data set than that of Apriori algorithm.

Keywords: Apriori algorithm;data mining;FP-Growth algorithm;association rules;game sale

1 引言(Introduction)

关联规则是近年来数据挖掘领域中最热门的问题之一，它已经被证明对于市场与零售业，以及其他不同领域都有很重要的作用。关联规则问题涉及了诸多技术知识，如数据库、人工智能和统计学等，该问题的研究目的是统计庞杂的数据并提取出有效信息进行分析，得到同一事件出现在不同项的相关性，关联规则发现的主要对象是交易型数据库。

Apriori算法是一种经典的挖掘关联规则的算法，该算法根据用户给出的最小支持度阈值找到交易型数据库中所有频繁项集，再通过计算找出符合最小置信度的强关联规则，从而挖掘出有用的知识。Apriori算法的核心思想有两点：(1)非频繁项的超集是非频繁项集；(2)频繁项的子集是频繁项集。该算法利用了一个层次顺序搜索的循环方法来完成频繁项集的挖掘计算。但是，多次扫描数据库和产生数量巨大的候选集是Apriori算法的两个无法避免的性能瓶颈[1]。很多基于Apriori算法的方法被提出，目的都是为了提搞扫描数据库的效率或是减少候选集的产生，其中AprioriTid算法对Apriori算法的循环扫描方式做出了改进[2]，AprioriTid算法仅在计算第一个频繁项集时扫描一次数据库，然后使用候选集$Ck-1$来计算项集的支持度并得到频繁$k-$项集，从而使扫描候选项集的次数随着频繁项集阶数的增加而逐步减少，提高了扫描数据的效率。在扫描的初始阶段，由于候选项集数量远大于数据项数量，这将导致候选事务的数据量可能远大于原始事务的数据量，所以此时AprioriTid算法的效率要低于Apriori算法[3-5]，而后随着候选项集的减少，AprioriTid只需要扫描比原始数据库小得多的候选事务数据库，使运算效率得以大幅提升。

2 理论基础(Theoretical basis)

关联规则挖掘的问题被定义为：

$I=\{i_1,i_2,\cdots,i_n\}$为一个或多个的n项组，项目是其中的一个字段，一般指一次交易中的一个物品[6]；

$D=\{t_1,t_2,\cdots,t_m\}$为一组被称为交易数据库的事务集，而每

个事务t都是I的非空子集，即每一个交易都与一个唯一的标识符对应。每一条事务中仅包含该事务涉及的项目，并不包含项目中的具体信息；

$X \to Y$：表示规则(Rule)，其中$X,Y \subseteq I$并且$X \cap Y \subseteq \varnothing$，项$X$和$Y$分别是规则的前提和结论，或被称为左手边与右手边；

$supp(X,Y)$：表示项X和Y的支持度，支持度的计算公式如公式(1)所示：

$$supp(X,Y)=P(X\&Y) \tag{1}$$

$conf(X \to Y)$：被称为规则$X \to Y$的置信度，置信度的计算公式如公式(2)所示：

$$Conf(X \to Y)=\frac{P(X\&Y)}{P(X)} \tag{2}$$

$minsupp(X)$：表示用户自定义的一个衡量支持度的阈值，同时也表示该项目集在统计意义上的最低阀值，用支持度来衡量规则是非常重要的，因为非常低的支持度只会偶然发生，低支持度的规则从商业的角度出发看起来也是没有意义的，因为推广客户购买非常低可能性同时出售的商品可能是无利可图的，基于上述原因，支持度常被用来消除无意义的规则；

频繁项集：对于一个项目集X，如果$supp(X) \geq minsupp(X)$，则称为频繁项集。

强关联规则：如果$X \to Y$的置信度和支持度不小于用户自定义的$minsupp$和$minconf$，则称$X \to Y$是一个强关联规则，否则为弱关联规则。

关联规则的挖掘是在事务数据库中，找到满足用户定义的最小支持度和最小置信度要求的关联规则，其过程主要有两个阶段：

(1)第一个阶段必须先从所有数据集合中找出所有的频繁项集。

在事务数据库D的所有数据中找出满足条件的全部频繁项的集合，也就是找出所有的$supp(X) \geq minsupp(X)$的项集X。

(2)第二个阶段是在这些频繁项中产生关联规则

利用频繁项集产生关联规则，针对每一频繁项集X，如果$Y \subset X$，Y非空，且$conf(X \to Y) \geq minconf$，则$X$与$Y$构成了关联规则，$X \to Y$满足用户给定的最小支持度和最小置信度。

关联规则的第一个阶段是从原始的数据集合中开始的，需要找出所有频繁项集，这一步骤是关联规则挖掘的一个重点问题，也是能够衡量关联规则算法优良的指标。频繁项集的是指某一个项出现的频率相对于所有记录而言，必须到达某一水平。第二个问题相对容易一些，目前所有的关联规则算法都是针对第一个问题提出的[7]。

3　Apriori算法与AprioriTid算法的研究(Research on Apriori algorithm and AprioriTid algorithm)

Apriori算法是一种基本的挖掘关联规则算法，该算法的第一步为确定初始的一个大项集，记为$L1$。接下来的K步包含两部分，第一部分，项集$Lk-1$做$(k-1)$次Apriori-gen循环以产生候选集Ck。第二部分，扫描数据并计算候选集Ck中每一项的支持度。如果存在候选K-集，其支持度不小于最小支持度，则判定为频繁K-项集，Apriori-gen循环到不再产生候选集结束。Apriori算法输入为销售型数据库D，和最小支持度阈值，输出的是计算过后的频繁项集，该频繁项集包含了所有满足最小支持度的项。

Apriori算法伪代码实现：

```
//找出频繁1-项集，即通过单遍扫描确定每个项的支持度，得到频繁1-项集的集合L1
L1=find_frequent_1-itemsets(D);
For(k=2;Lk-1!=null;k++)
{
Ck=apriori_gen(Lk-1);        //产生候选集，并剪枝
For each 事务t 属于 D        //扫描D并计数
{
Ct=subset(Ck,t);             //得到t的子集
For each 候选子集c属于Ct
c.count++;
}
//返回候选项集中，候选子集不小于最小支持度的的集合，即频繁k-项集Lk
Lk={c属于Ck|c.count>=min_sup}
}
Return L=所有的频繁项集;
第一步：连接步
Procedure apriori_gen(Lk-1:frequent(k-1)-items)
For each 项集l1处于Lk-1
    For each 项集l2属于Lk-1
If((l1[1]=l2[1])&&(l1[2]=l2[2])&&
……&& (l1[k-2]=l2[k-2])&&(l1[k-1]<l2[k-1]))
then
{
    c=l1连接l2
    //连接步：产生候选子集c，若频繁k-1-项集Lk-1中已经存在子集c则进行剪枝
    If has_infrequent_subset(c,Lk-1)then
            Delete c;
```

```
//剪枝步：删除不符合条件的候选子集
        else add c to Ck；
    }
    Return Ck；
```

即通过将Lk−1与自身连接产生候选k−项集的集合Ck。

第二步：剪枝步

```
Procedure has_infrequent_subset(c:candidate k−itemset；
    Lk−1:frequent(k−1)−itemsets)
    For each (k−1)−subset s of c
        If s不属于Lk−1 then
                Return true；
        Return false；
```

如代码中描述的，Apriori算法的工作模式是先生产，再检验。如果数据集非常大，那么不断扫描数据集就会造成运算效率较低这一问题。

FP−Growth算法的思路是首先扫描两次数据集，构建出FP−Tree；然后将数据集中的所有事务映射在FP−Tree上；最后通过FP−Tree找出其中的频繁项集。FP−Growth算法的输入输出与Apriori算法一致，而构建FP−Tree的过程为：

1)第一次扫描数据库，计算每个项的支持度并得到频繁1−项集

2)把项按支持度递减排序

3)第二次扫描数据库，建立FP−Tree

挖掘频繁项集的过程：

1)根据D和minsupp，调用FP−Tree；

2)倘若FP−Tree是一条简单路径

组合路径上所有支持度大于minsupp的节点，得到频繁项集

else

初始化最大频繁集

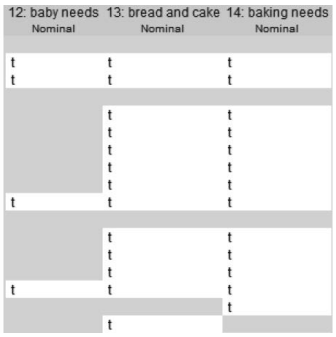
3)根据支持度从大到小排序，以每个1−频繁项为后缀，调用挖掘算法挖掘最大频繁项集

4)输出所有频繁项集。

根据以上两个过程，我们就可以完成FP−Growth的频繁项集挖掘。FP−Growth算法可以避免产生大量的候选集，但由于该伏安法要递归生成条件数据库和条件FP−Tree，所以内存开销较大，且只能用于挖掘单维的布尔型关联规则。

4 对比试验(Contrast test)

本实验通过使用Weka软件，对Apriori算法与FP−Growth算法进行比对，输入数据为超市商品销售数据，其格式如图1所示。



12: baby needs	13: bread and cake	14: baking needs
Nominal	Nominal	Nominal
t	t	t
t	t	t
	t	t
	t	t
	t	t
	t	t
	t	t
t	t	t
	t	t
	t	t
	t	t
t	t	
	t	t
	t	

图1 输入的超市销售数据

Fig.1 Input of supermarket sales data

其横坐标代表了每一次顾客的购买记录，若购买了婴幼儿必须品，则在该类别中以t为记录，纵坐标表示了商品的种类。输入最小支持度与置信度，当计算出最小支持度后，与项的概率相除即置信度，既可以找出相应支持度与置信度的关联规则，首先应用Apriori算法对书籍进行处理，当最小支持度设置为0.15，置信度设置为0.9时，得到的关联规则如图2所示。

```
1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <
2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <
3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705
4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <co
5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(
6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725
7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701
8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> li
9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)
11. baking needs=t fruit=t vegetables=t total=high 831 ==> bread and cake=t 752
12. biscuits=t milk-cream=t total=high 907 ==> bread and cake=t 820    <conf:(0.9)
13. biscuits=t vegetables=t total=high 950 ==> bread and cake=t 858    <conf:(0.9)
14. baking needs=t fruit=t total=high 963 ==> bread and cake=t 869    <conf:(0.9)>
15. tissues-paper prd=t fruit=t total=high 878 ==> bread and cake=t 792    <conf:(
16. margarine=t fruit=t total=high 818 ==> bread and cake=t 737    <conf:(0.9)> li
```

图2 支持度=0.15，置信度=0.9时，生成的关联规则

Fig.2 Generated association rules when the support coefficient is 0.15 and the confidence coefficient is 0.9

由图可知当支持度为0.15，置信度为0.9时，购买饼干、冷冻食品、水果，则很有可能买面包和蛋糕，购买炊事用品、饼干和水果时、很有可能买了面包和蛋糕等。

再利用FP−Growth算法，对同样的数据进行挖掘计算，得到的结果如图3所示。

```
1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <
2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <
3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705
4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <co
5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(
6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725
7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701
8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> li
9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)
11. baking needs=t fruit=t vegetables=t total=high 831 ==> bread and cake=t 752
12. biscuits=t milk-cream=t total=high 907 ==> bread and cake=t 820    <conf:(0.9)
13. biscuits=t vegetables=t total=high 950 ==> bread and cake=t 858    <conf:(0.9)
14. baking needs=t fruit=t total=high 963 ==> bread and cake=t 869    <conf:(0.9)>
15. tissues-paper prd=t fruit=t total=high 878 ==> bread and cake=t 792    <conf:(
16. margarine=t fruit=t total=high 818 ==> bread and cake=t 737    <conf:(0.9)> li
```

图3 置信度=0.15，置信度=0.9时，生成的关联规则

Fig.3 Generated association rules when the support coefficient is 0.15 and the confidence coefficient is 0.9

（下转第19页）