关联规则挖掘在分析型 CRM 中的应用

Application of Association Rule Mining in Analytical CRM

张新香　（中南财经政法大学信息学院 武汉 430064）

摘要：本文在分析传统关联规则挖掘的基本思想和在分析型 CRM 应用的基础上，提出前件固定后件约束和后件固定前件约束两种基于约束的关联规则挖掘算法，并分析了它们在 CRM 中的实际应用。

关键词：CRM　关联规则　约束

1　引言

随着 internet 时代的到来，电子商务得到迅速发展，企业要想赢得竞争份额，就得借助先进的管理技术，记住“以客户为中心”的理念，挖掘客户、把握客户、拓展客户。而分析型的 CRM 技术正是利用数据仓库、数据挖掘技术，对交易操作所积累的大量数据过滤、抽取、分析；从而帮助企业管理人员发现市场规律，有效地改变经营方针，提升企业的市场竞争力。

分析型 CRM 中用到的数据挖掘技术包括关联规则挖掘，借助关联规则能发现交易数据中不同商品之间的关系，找出顾客的购买行为模式，为商家进行商品货架设计、货存安排、个性化页面推荐系统的生成、商品的交叉销售，提供极富意义的科学依据。

2　传统的关联规则及其在 CRM 中的应用

关联规则由美国的 Agrawal 等人提出的，是数据挖掘技术中一种简单而又实用的规则。

2.1　关联规则挖掘的 Apriori 算法

Apriori 算法是关联规则挖掘的核心算法，其过程分为两步：

（1）找出所有支持度大于最小支持度的项集，这些项集称为频集，包含 K 个项的频集称为 K－项集。

（2）对于每个频繁项集，产生满足最小置信度要求的规则。

频繁项集挖掘算法如下：

输入：事务数据库 D；min_support（最小支持度阈值）；

输出：D 中的频繁项集 L；

```
Procedure Apriori(D,min_support)
  L1 = find_frequent_1 – itemsets(D); //产生频繁一项集 L1
    For(k=2;Lk-1≠¢;K++)
      { Ck = Apriori_gen(Lk-1,min_support);
        For each transaction t∈D
        { Ct = Subset(Ck,t);
          For each candidate c∈Ct
          c.count++;
        }
      Lk = {c∈Ck | c.count ≥ min_support};      }
Return L = ⋃k Lk;
```

2.2　传统关联规则挖掘在 CRM 中的应用

以购物篮为例，说明传统关联规则挖掘在 CRM 中的应用。

过滤、分析销售数据库，可以发现人们对面包、牛奶、香肠的购买情况如下：

记录号	面包	牛奶	香肠
1	1	1	0
2	0	1	0
3	1	1	0
4	1	0	1
5	0	1	0

其中1代表购买,0代表没有购买,假设设定最小支持度为20%,Apriori算法步骤如下:

产生第一频集 L_1

项目集X	支持度/%
面包	60(3/5)
牛奶	80(4/5)
香肠	20(1/5)

都满足最小支持度要求,最终 L_1 频集

项目集X	支持度/%
面包	60(3/5)
牛奶	80(4/5)
香肠	20(1/5)

得出 L_2 候选集

项目集X	支持度/%
面包,牛奶	?
面包,香肠	?
牛奶,香肠	?

产生 L2 频集

项目集X	支持度/%
面包,牛奶	40(2/5)
面包,香肠	20(1/5)

同样还可以得出 L_3 候选集(面包,牛奶,香肠),但是其支持度为0,不满足条件。

同时算出:

Conf(面包=>牛奶)=P(面包,牛奶)/P(面包)=40%/60%=67%

Conf(面包=>香肠)=P(面包,香肠)/P(面包)=20%/60%=33%

从中可以发现有意思的规律,买面包,牛奶的顾客较多,而且顾客买面包再买牛奶的时候比买面包再买香肠的时候多,基于这样分析,我们可以将面包和牛奶放在显眼的货架或页面主要位置,同时考虑将面包和牛奶放得比较靠近,或者在它们之间形成超链接,方便客户选择。

3 基于约束的关联规则在分析型CRM中的应用

传统的关联规则思路简单,但是存在诸多缺点,比如:

(1) 缺乏用户的主动参与和控制。用户只能在挖掘的初始阶段设定最小支持度和最小置信度参数,不能知晓挖掘进程的进展情况,也无法调整最小支持度、最小置信度参数对挖掘进程进行控制。

(2) 产生大量规则,使用户难以理解,且其中有很多规则用户并不需要。

(3) 规则的冗余度较高,冗余的规则不能提高任何新的信息,反而降低处理速度。

鉴于此,很多学者开始改进传统的关联规则挖掘算法,基于约束的关联规则可以很好的克服上述传统关联规则的不足,由于要解决问题的侧重点不同,约束条件大致有项目约束、长度约束、模型约束、集合函数约束、项目属性约束等5类。本文所探讨的前件固定,后件受约束的约束方式是项目属性约束(通过对项目的某些属性作限制)的一种,其中前件即规则的左部A,后件即规则的右部B,后件固定,前件受约束的的约束方式是长度约束(限制规则中项目数量)和项目属性约束的综合,他们的提出给分析性CRM提出了新的解决思路。

3.1 前件固定,后件受约束的关联规则

3.1.1 基本思路和算法改进

该约束问题的约束条件如下:

$Sum\_price(B_1,B_2,\cdots,Bn) <= max\_sumprice$

其中,$Sum\_price(B_1,B_2,\cdots,Bn)$ 为规则后件所有产品的价格之和,而 max_sumprice 为规则后件价格之和的最大可能取值,该值的选取根据前件的价格来定。对于前加固定,后件受约束的关联规则有如下特点:

(1) 约束是反单调的。如果 $Sum\_price(B_1,B_2,\cdots,B_k) > max\_sumprice$,那么,$B_1,B_2,\cdots,B_{k+1}$,即 $B_1,B_2,\cdots,B_k$ 的任何超集都不可能小于等于 max_sumprice,这样可以裁掉K-项集中的$\{B_1,B_2,\cdots,B_k\}$,,减少候选项的数目,提高算法的执行效率。

(2) 对于规则 $A=>B_1$ 和 $A=>B_1,,B_2$:因为有

$conf(A=>B_1)=P(AB_1)/P(A)$

$conf(A=>B_1,B_2)=P(B_1,B_2/A)=P(A,B_1,B_2)/P(A)$

并且

$P(B_1/A)>=P(B_1,B_2/A)\ P(A)=P(A)$

所以

$conf(A=>) >= conf(A=>B_1,B_2)$

即如果规则 $A=>B_1$ 不满足最小置信度的要求,则以规则后件 B_1 的超集为后件的规则(如 $A=>B_1,B_2$),也不可能满足最小置信度的要求,这样,可以裁掉2-项集$\{A,B_1\}$裁掉,以利于下一项集的快速产生。

对于前件固定的关联规则挖掘,我们只需找出用

户制定的规则前件的各个频繁项，不包含用户制定的规则前件的事务可完全不必考虑，方法是，我们可以将包含规则前件的所有事务选入一个临时表中 D'，以后的处理针对该临时表，这样可以减少数据库的扫描时间，同时，由于临时表中每个事务肯定包含规则前件项，每个频繁项中都包含该项，所以完全不必考虑该项，这样也可以减少扫描数据库的时间，提高算法的处理效率。

鉴于此，可以改进 Apriori 算法如下：

输入：事务数据库 D；min_supt（最小支持度阀值）；min_conf（最小置信度阀值）；

max_sumprice（规则后件价格之和最大值）；antecedent（固定的规则后件）；

输出：满足；min_sup，min_conf，max_sumprice，antecedent 约束的规则；

Procedure Apriori_1（D，min_sup，min_conf，max_sumprice，antecedent）

If antecedent is unfrequent then

Return；

Select T from D where contain antecedent into D'

Delete from D' where 商品编号 = antecedent；

L_1 = find_frequent_1 – itemsets（D'）；

Delete T where not contain L_1

Get – rules（1，L1）//产生以频繁 1 项集为后件的规则

For（k = 2；$L_{k-1} \neq \Phi$；K = K + +）

{ C_k = Apriori_gen（L_{k-1}，min_sup，max_sumprice）；产生 K'项集

L_k = Subset（C_k，D'）；//产生频繁 K 项集

Get – rules（k，L_k）；//产生频繁 K 项集为后件的规则

}

Return；

3.1.2 在分析型 CRM 中的应用

运用前件固定，后件受约束的关联规则挖掘，我们可以解决如此问题。对于产品销售较好的产品（比如产品 A），我们可以分析出哪些产品（如 $B_1, B_2, \cdots, B_k$）经常和该产品 A 一起被购买，从而给出促销的措施，一旦购买产品 A，即前件固定，将同时赠送或打折销售产品（$B_1, B_2, \cdots, B_k$），当然要求产品（$B_1, B_2, \cdots, B_k$）的价值较低，即后件约束。这样既显得商家产品丰富，又不会出现部分产品滞销，同时还可以刺激客户的购买欲望，有效提高商家的市场占有率，更好的提高企业利润。

3.2 后件固定，前件受约束的关联规则

3.2.1 基本思路和算法改进

诸如这样问题：（$B_1, B_2, \cdots, B_n$）=>C 要求后件固定，前件受约束，约束条件如下：

Price（B_i）>= min_price

其中，Price（B_i）为规则前件中的任何一项价格；min_price 为规则前件价格的最小值，当然该值由用户根据后件的价格来决定。同时可以发现，当规则前件的数目太多时，规则将会毫无意义，因此还要将规则前件加约束，即规则前件长度不能太长，表示如下：

Count（antecedent）<= max_antecedent_count

其中 Count（antecedent）为规则前件的数目，max_antecedent_count 为规则前件数目最大值，由用户根据实际情况和经验来决定。

这两个约束条件能极大的减少项集，促使频繁项集的快速生成，从而减少扫描数据库的次数，节约时间，提高算法的效率。但是仔细分析，我们可以看到这样的规则：

Conf（B_1, B_2 => C）=80%（规则 1）

Conf（B_1 => C）=85%（规则 2）

很容易可以看出，规则 1 相对规则 2 是没有多大意义的。鉴于此，我们给出规则的改进度概念，规则的改进度即在规则前件固定时，一个规则的置信度与其子规则置信度差的最小值。描述如下：

对于规则 C => A，其改进度为：

imp（C => A）= min（任何 C' $\cup$ C，conf（C => A）– conf（C' => A））；依据稠密数据库自身的特点，一般要求改进度大于 0，但在实际工作中，当改进度小于 0 时，但是购买 C 的收入肯定大于购买 C'，给商家带来的利润要高，所以确定改进度的值，要依据实际情况来定，不一定非要大于 0。

综上分析，改进后的算法如下：

输入:事务数据库 D; min_sup(最小支持度阀值);min_conf(最小置信度阀值);max_antecedent_count(规则前件的最大数目); min_price (规则前件价格最小值);consequent(固定的规则后件);min_imp(最小改进度);

输出:满足 min_sup,min_conf,min_price,max_atecedent_count,min_imp,consequent 约束的所有规则;

```
Procedure Apriori_2 ( D, min_sup, min_conf, max_antecedent_count, min_price , consequent, min_imp)
    Select T from D into D';
    If consequent is frequent then
      Add consequent to L1;
    else
      return;
    L1 = L1 + find_frequent_1 – itemsets( D' ); //产生频繁 1 项集
    Delete T which not contain L1;
    while( Lk-1 ≠ ¢ and k < = AntecedentCount + 1)
    {
    Ck = apripri – gen( Lk-1, min – sup); //产生 K 项集
    Lk = subset( Ck, D' ); //产生频繁 K 项集
    gen – rules( k, Lk,, Lk-1)) //产生具有固定后件的规则
    }
  return;
procedure gen – rules( k, LK, LK – 1)
  for each c in LK
    if consequent ∈ c
  {
  conf = sup( c) /sup( c – consequent);
  if conf > = min – conf then
  {
   max – subconf = 0;
  for all( k – 1) – subset s in ( c – consequent)
   if conf( s) > max – subconf then max – subconf = conf( s);
    if ( conf – max – subconf) > = min – imp then
    输出 the rule c – consequent = > consequent;
      }
  }
  return;
```

3.2.2 在分析型 CRM 中的应用

当商家要推出新产品时(如 D),我们可以分析其同类产品或该产品的前身(如 C),利用挖掘算法,分析出哪些产品(B1,B2,…,Bn)和产品 C 一起被购买,从而在销售这些产品时,打折销售或赠送产品 D,进而实现产品 D 的有效推广。这样就要求产品 Bi 的价格不能太低,并且产品的总类不能太多,也就是满足上述约束规则中的后件约束要求。

4 结束语

经过试验,在传统关联规则挖掘算法基础上提出的约束关联规则挖掘算法效率较高,而且更能为分析型 CRM 提供有力的科学依据,可以有效的为商家提供产品交叉销售决策支持。当然事务数据库中除了顾客购买的商品信息,还有顾客个人信息和购买的地域时间等多方面的信息,对这些信息运用关联规则进行挖掘分析,也能很好的为企业提供管理决策,提升企业的市场竞争力。

参考文献

1 陈京民,数据仓库原理、设计与应用,北京,中国水利水电出版社,2004。

2 Jiawei Han, Micheline Kamber, DATA MINING Concepts and Techniques, 北京 高等教育出版社, 2001。

3 秋林、力士奇,客户关系管理[M],北京 清华大学出版社,2002。