商业销售中的利润挖掘及商品选择算法

徐秀娟, 贾立峰, 周春光, 王 喆, 徐笑昂
(吉林大学 计算机科学与技术学院, 长春 130012)

摘要: 给出 ItemRank算法解决带有交叉影响的商品选择问题, 构建了以顾客为导向的购买行为模型, 在此基础上给出算法 ItemRank. 同时从马尔可夫随机链出发提出了 SALSARank算法模拟顾客行为. 实验表明, 这两种算法在选择商品的利润评估中具有较好的效果.
关键词: 利润挖掘; 商品选择算法; 交叉销售; 关联规则
中图分类号: TP183 文献标识码: A 文章编号: 1671-5489(2006)02-0201-06

ProfitMining and Item Selection Algorithms in Commerce

XU Xiu-juan, JIA Li-feng, ZHOU Chun-guang, WANG Zhe, XU Xiao-ang
(*College of Computer Science and Technology, Jilin University, Changchun* 130012, *China*)

Abstract: ItemRank algorithm is presented to solve the problem of item selection with the consideration of cross-selling effect. First a new-devised model—— CustomerBehaviorModel is proposed. And then the novel algorithm ItemRank is implemented on the basis of CustomerBehaviorModel. SALSARank algorithm is based on the theory of Markov chain to simulate customer's behavior. The experiments showed the advantage and effect about the algorithms.
Key words: profit mining; item selection algorithm; cross-selling; association rule

在数据挖掘中有很多不同的关联规则算法, 可是很少有算法应用到具体的商业销售中. 对于商业销售人员, 即使知道了这样的关联规则也不能判断如何在销售时得到最大的利润. 这里最大的障碍是因为每个商品的利润是不同的, 而且商品之间的销售是相互依赖的.

现在零售商已经开始注意到, 商品之间具有交叉影响[1], 即商品的利润不仅与自身相关, 也受相关商品的影响. 如某些商品本身虽不能产生很高的利润, 但它能刺激其他高利润商品的销售. 商品之间的互相依赖性可以用来支持零售商的市场决策. 于是寻找这种相关性成为关联发现中一个重要的问题. 但是, 当前关联规则的发现方法不足以支持这种基于利润的挖掘. 因此如何将发现的规则应用于实际是近年来数据挖掘研究的新方向.

利润挖掘[2,3]是数据挖掘应用的研究热点, 也是数据挖掘的最终方向. 利润挖掘以微观经济观点[4]为基础, 将数据挖掘的知识应用于商业的销售领域中. 利润挖掘使人们注意到, 仅仅找到规则是不够的; 必须能对这些规则做出响应, 能依据规则行动, 最终将数据转换为信息, 将信息转换为行动, 将行动转换为价值[4]. 这样数据挖掘才完成了一个周期.

本文提出一个商品选择算法 ItemRank(ItemRank算法考虑了商品之间的交叉影响因子, 能快速地

收稿日期: 2005-03-14
作者简介: 徐秀娟(1978~), 女, 汉族, 博士研究生, 从事利润挖掘和数据挖掘的研究, E-mail: xuxiujuan666@yahoo.com.cn 联系人: 周春光(1947~), 男, 汉族, 教授, 博士生导师, 从事计算智能、数据挖掘和进化计算的研究, E-mail: cgzhou@jlu.edu.cn
基金项目: 国家自然科学基金(批准号: 60433020)、教育部重点项目基金(批准号: 02090)和教育部"符号计算与知识工程"重点实验室基金.

找到较好的方案达到目的),讨论了商品中马尔可夫链,提出另一个算法 SALSARank,并给出了两个合成数据库进行算法性能比较.

1 商品选择问题中的交叉影响

利润挖掘中的最优产品选择问题[5]是典型的零售商问题,即零售商通过定期更新商品的类型来将损失利润的商品抛弃以引入新商品,因此要找到一个有意义的离散子集以最大化利润.这类问题考虑了影响其他商品销售的交叉因子,这个因子可以通过分析以往的交易获得.

给出一个数据集,有 m 个交易 $t_1, t_2, \cdots, t_m$ 和 n 个项集 $I_1, I_2, \cdots, I_n$. 设 $I=\{I_1, I_2, \cdots, I_n\}$. 交易 t_i 中商品 I_a 的利润称为 I_a 的单独利润,记为 $\mathrm{prof}(I_a, t_i)$. 设 $S\subset I$ 为选择的商品集合 J. 为了计算该选择能获得的利润,在每个交易 t_i 中,定义两个符号 t_i' 和 d_i:

$$t_i' = t_i \cap S, \qquad d_i = t_i - t_i'. \tag{1.1}$$

其中,t_i' 指在交易 t_i 中被 S 选择的商品集合;d_i 指交易 t_i 中未被 S 选择的商品集合.

假定选择商品子集 S 意味着将排除在 $I_1, I_2, \cdots, I_n$ 中的一些商品.如果事先移走了这些商品,也许不能精确地发生交易 $t_1, \cdots, t_m$,因为顾客如果知道不能买到某些商品,他们也许就不会购买另一些商品.因此,当去掉这些商品时,交叉销售因子 $\mathrm{csfactor}(D, I_a)$ 将影响利润 $\mathrm{prof}(I_a, t_i)$,其中 D 是商品集合,$0\leqslant \mathrm{csfactor}(D, I_a)\leqslant 1$. $\mathrm{csfactor}(D, I_a)$ 为不能提供 D 中商品时在交易中损失的 I_a 的利润部分.

定义 1.1[5] 一个商品选择 S 的完全利润为

$$P=\sum_{i=1}^{m}\sum_{I_a\in t_i'}\mathrm{prof}(I_a, t_i)(1-\mathrm{csfactor}(d_i, I_a)). \tag{1.2}$$

定义 1.2(MPIS(Maximal Profit Item Selection)问题)[5] 给出交易集合,并指定每个交易中每个商品的利润和交叉销售因子 $\mathrm{csfactor}(\ )$,从中选择 J 个商品的集合 S,S 能给出 J 个商品中的最大利润.

可以证明,MPIS问题为 NP问题[5].

设从给定的交易集合中评估可能的利润.假定从商品集合选出 J 个商品的集合 S,考虑交易集合中的交易 t_i,假定 t_i 中有一些已选商品如 I_a,$I_a\subset S$,同时有一些未选商品如 d_i,$d_i\subset I-S$. 此时,在商品选择的完全利润中为交叉因子建模,使用 $\mathrm{conf}(I_a\rightarrow \diamondsuit d_i)$ 确定未选商品对已选商品的影响 $\mathrm{csfactor}(d_i, I_a)$,其中定义 $\diamondsuit d_i$ 为:

定义 1.3[5] 设 $d_i=\{Y_1, Y_2, Y_3, \cdots, Y_q\}$,其中 Y_i $(i=1, 2, \cdots, q)$ 指单个商品,则 $\diamondsuit d_i=Y_1\vee Y_2\vee Y_3\vee\cdots\vee Y_q$,规则 $I_a\rightarrow\diamondsuit d_i$ 称为损失规则.

损失规则 $I_a\rightarrow\diamondsuit d_i$ 暗示着购买了商品 I_a 的顾客一定也购买了 d_i 中至少一项商品.此时,给出基于关联规则的商品选择 S 的完全利润如下:

$$P=\sum_{i=1}^{m}\sum_{I_a\in t_i'}\mathrm{prof}(I_a, t_i)(1-\mathrm{conf}(I_a\rightarrow\diamondsuit d_i)). \tag{1.3}$$

损失规则 $I_a\rightarrow\diamondsuit d_i$ 也可以看作一种关联规则.这个规则的信任度 $\mathrm{conf}(I_a\rightarrow\diamondsuit d_i)$ 类似于一般的关联规则,使用下式求解[5]:

$$\frac{\text{包含 } I_a \text{ 且包含 } d_i \text{ 中至少任一元素的交易数量}}{\text{包含 } I_a \text{ 的交易数量}}. \tag{1.4}$$

2 基于顾客购买模型的算法

2.1 顾客购买模型

在微观经济中,考察的是个体行为;对于数据集需要考察单笔交易,即购物篮分析.对于单笔交易 $t_i=\{I_1, I_2, \cdots, I_n\}$,交易中的商品是互相关联的,有些商品之间关系密切.可以假想,一个顾客在购买商品时,首先随机挑选商品 I_k,然后以一定的相关概率 d 随机挑选下一个商品 I_{i+1};或者以 $1-d$ 的

概率购买与 I_i无关的商品.

从微观经济观点来看，假设某顾客进入零售店随机选择商品；该顾客在选择了某商品后，然后随机选择下一个商品. 这种随机购买一个商品的可能性就是该商品的权重.

以随机冲浪模型[6]为基础，顾客购买模型的核心思想描述如下：如果有许多商品都可以指向某一个商品，那么该商品的权重值应该很高，或者该商品有一些带有高权重值的商品指向它. ItemRank算法处理了上述情况，通过商品之间的关联规则递归地传递该权重.

2.2 ItemRank算法

ItemRank算法将所有的商品看作一个有向图 $G=(V, E)$，V是所有商品的集合，E为边集，当且仅当存在从商品 I_i到商品 I_j的关联时存在从 i到 j的边. 假定一个商品 I_i有指向它的商品 $I_1, \cdots, I_n$，N_i表示节点 I_i的出度. 有向边 $(p, q)\in E$表示商品 p有指向商品 q的链接. 假定频繁商品的数量远远大于将要选择出商品的数量. 链接 $I_i\rightarrow I_j$表示从 I_i到 I_j的交叉销售影响.

首先，使用 $\mathrm{prof}(I_i)$表示商品 I_i的单独利润，$\mathrm{conf}(I_i\rightarrow I_j)$表示关联规则 $I_i\rightarrow I_j$的信任度. 对于商品 I_i和 I_j，链接 $I_i\rightarrow I_j$的交叉销售影响，不仅依赖于信任度 $\mathrm{conf}(I_i\rightarrow I_j)$，也依赖于商品 I_i的单独利润 $\mathrm{prof}(I_i)$. 为了考虑商品的单独利润，在链接中合并单独利润来解决这个问题. $\mathrm{prof}(I_i)\times\mathrm{conf}(I_i\rightarrow I_j)$代表没有选择 I_j所损失的 $\mathrm{prof}(I_i)$部分.

ItemRank算法根据一个商品所关联的商品数量和质量给该商品分配一个绝对的权重，同时也将链接商品的权重等级考虑在内. 指向一个商品的外部链接商品的权重等级越高，则该链接商品传递给该商品的权重等级值也越高. 如果存在一条从 I_i指向 I_j边，则认为 I_i得到了 I_j的认可；如果有许多这样的边，则可以说 I_i是相对重要的. 计算商品 I_i的等级初始 $\mathrm{ItemRank}(I_i)$(简写为 $\mathrm{IR}(I_i)$)可以写成：

$$\mathrm{IR}(I_i)=\frac{\mathrm{IR}(I_1)}{N_1}+\cdots+\frac{\mathrm{IR}(I_n)}{N_n}=\sum_{(i,j)\in E}\frac{\mathrm{IR}(I_j)}{N_j}. \tag{2.1}$$

式(2.1)没有考虑商品之间的信任度 $\mathrm{conf}(I_i\rightarrow I_j)$，也没有考虑商品自身的利润 $\mathrm{prof}(I_i)$. 如果有 $I_i\rightarrow I_j$的链接，则使用 $\mathrm{prof}(I_i)\times\mathrm{conf}(I_i\rightarrow I_j)$评估 I_j的 IR值. 于是式(2.1)变为

$$\mathrm{IR}(I_i)=\sum_{(i,j)\in E}\frac{\mathrm{IR}(I_j)\times\mathrm{prof}(I_j)\times\mathrm{conf}(I_j\rightarrow I_i)}{N_j}. \tag{2.2}$$

式(2.2)想要得到准确的结果，必须依赖的前提就是要求所有商品应该形成一个牢固的强连通图. 但实际商品很难这样理想，此时可能存在两个问题[6]：等级沉没(rank sink)和等级泄漏(rank leak). 在其中加入衰减因子 d(d为(0, 1)之间的衰减系数)避免上述问题. 这样一个商品的等级 IR仅 d部分在它所关联的商品中分配，剩下的部分用在所有的商品中分配，通常设 d为 0.2 即当一个顾客购买商品 I_i时，存在 d的可能性购买与商品 I_i相关联的商品 I_j，同时也存在 $1-d$的可能性购买与商品 I_i无关的其他商品 I_k. 此时商品 I_i的 $\mathrm{IR}(I_i)$变为

$$\mathrm{IR}(I_i)=d\sum_{(i,j)\in E}\frac{\mathrm{IR}(I_j)\times\mathrm{prof}(I_j)\times\mathrm{conf}(I_j\rightarrow I_i)}{N_j}+\frac{1-d}{m}, \tag{2.3}$$

其中，m是图中节点总数. 设 $\boldsymbol{x}=\{\mathrm{IR}(I_1), \cdots, \mathrm{IR}(I_n)\}$，$\boldsymbol{e}$是单位行向量 $\boldsymbol{e}=\{1, \cdots, 1\}$，注意到商品购买的可能性分布，所有商品的 IR和为 1，则有 $\boldsymbol{ex}=1$. 设 $\boldsymbol{I}$为单位矩阵. 对于矩阵 $\boldsymbol{A}$，其中任一元素 a_{ij}，如果存在从 I_i到 I_j的边，则 $a_{ij}=\mathrm{prof}(I_i)\times\mathrm{conf}(I_i\rightarrow I_j)/N_j$，否则 $a_{ij}=0$. 这样式(2.3)可以表示为

$$\begin{aligned}\boldsymbol{x}&=\begin{pmatrix}\mathrm{IR}(I_1)\\ \vdots\\ \mathrm{IR}(I_n)\end{pmatrix}=\frac{1-d}{m}\times\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}+d\times\boldsymbol{A}\times\begin{pmatrix}\mathrm{IR}(I_1)\\ \vdots\\ \mathrm{IR}(I_n)\end{pmatrix}=\frac{1-d}{m}\times\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}\times 1+d\times\boldsymbol{A}\times\boldsymbol{x}=\\ &\frac{1-d}{m}\times\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}\times\boldsymbol{e}\times\boldsymbol{x}+d\times\boldsymbol{A}\times\boldsymbol{x}=\left(\frac{1-d}{m}\times\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}\times\boldsymbol{e}+d\times\boldsymbol{A}\right)\times\boldsymbol{x}=\\ &\left(\frac{1-d}{m}\times\boldsymbol{I}+d\times\boldsymbol{A}\right)\times\boldsymbol{x}=\boldsymbol{Bx},\end{aligned} \tag{2.4}$$

其中 $\boldsymbol{B}=(1-d)/n\times\boldsymbol{I}+d\times\boldsymbol{A}$. 通过求解式(2.4)中矩阵 $\boldsymbol{B}$ 的主特征向量得到商品的排序结果.

3 基于顾客购买随机性的算法

马尔可夫链[7]是事件的一种概率模型，其中一个事件的概率完全取决于紧接它前面的事件，用 X_i 表示某事件第 i 次发生的概率，假设已知第0次到第 n 次的试验结果，则第 $n+1$ 次发生的概率仅和第 n 次有关，而与之前的 $n-1$ 次结果无关，此随机过程 $(X_1, X_2, X_3, \cdots, X_{n+1})$ 称为马尔可夫链.

在HAP算法[8]中，商品能分成 hub 和 authority 两类. 如果商品 I_j 对许多商品的购买是必要的，即有很多 $\mathrm{conf}(I_i\rightarrow I_j)$，则商品 I_j 是好的 authority 商品. 如果商品 I_i 对许多商品的购买是必要的，则商品 I_i 是好的 hub 商品. HAP算法考虑的是 authority 商品和 hub 商品之间的加强关系. ItemRank 算法是基于用户随机的向前购买商品的直觉知识. 实际上顾客大多数情况下是向前购买商品，但是很多时候也会回退购买商品. 基于上述直觉知识，修改 SALSA 算法[7]来计算商品选择问题.

从微观经济看，SALSARank 算法依赖于单次交易的随机特性. SALSARank 算法将商品看成两个不同的马尔可夫随机链，即 hub 链和 authority 链. Authority 为具有较高价值的商品，依赖于指向它的商品；而 hub 为指向较多 authority 的商品，依赖于它所指向的商品. 考虑核心商品 t，与 t 相关的商品有好的 hub 商品 $t\text{-}h$ 和好的 authority 商品 $t\text{-}a$ 两个集合；商品 t 有两个权重，即中心权重(hub weight)和权威权重(authority weight). 对于单次交易，可以看作是在两条马尔可夫链上进行随机行走的结果. 这两个马尔可夫链集合以随机方法替代了互相加强的关系. 同时，SALSARank 算法考虑用户回退购买商品的情况，更强调顾客购买的随机性以及向前购买商品的直觉知识，借鉴了随机购买思想，同时摒弃了HAP算法描述的两类商品相互增强的方法.

根据以上描述，可以将商品看成二分无向图 $G=(V_h, V_a, E)$，V 是所有商品的集合，E 为边集.

$$V_h=\{s_h \mid s\in C \text{ and } \text{out-degree}(s)>0\}\text{表示 } G\text{ 的 hub 顶点集合；}$$

$$V_a=\{s_a \mid s\in C \text{ and } \text{in-degree}(s)>0\}\text{表示 } G\text{ 的 authority 顶点集合；}$$

$$E=\{(s_h, r_a) \mid s\rightarrow r \text{ in } T\}\text{表示 } G\text{ 的边.}$$

每个非孤立商品 $s\in G$ 在图 G 中表示两个节点 s_h 和 s_a，每条边 $s_h\rightarrow r_a$ 表示从 s_h 到 r_a 的无向边.

在图 G 中将执行两种随机步. 如果从一条马尔可夫链走入另一条马尔可夫链，则称为前进随机步；否则，称为后退随机步. 每种随机步将从图的一边开始访问节点. 每一步形成了图 G 中的一个边. 每一步要么前进要么后退. 与核心商品 t 相关的 hub 商品 $t\text{-}h$，应该是以 V_h 这样的随机步更频繁地出现在其后访问的 hub 商品集合中. 同理，与商品 t 相关的 authority 商品 $t-a$，应该是以 V_a 这样的随机步更频繁地出现在其后访问的 authority 商品集合中.

以两条不同的随机链模拟这样的随机购买行为，即图 G 中的 authority 链和 hub 链. 定义这两条马尔可夫链:

$$\begin{aligned}H_{i,j}&=\sum_{k\mid(i_h,k_a),(j_h,k_a)\in G}\frac{\mathrm{profit}(i_h)\times\mathrm{conf}(i_h\rightarrow k_a)}{\deg(i_h)}\times\frac{\mathrm{profit}(j_h)\times\mathrm{conf}(j_h\rightarrow k_a)}{\deg(k_a)},\\ A_{i,j}&=\sum_{k\mid(k_h,i_a),(k_h,j_a)\in G}\frac{\mathrm{profit}(i_a)\times\mathrm{conf}(k_h\rightarrow i_a)}{\deg(i_a)}\times\frac{\mathrm{profit}(k_h)\times\mathrm{conf}(k_h\rightarrow j_a)}{\deg(k_h)},\end{aligned}\tag{3.1}$$

其中 $A_{i,j}$ 表示某一商品 k_h 同时指向 i_a 和 j_a；即有 $k_h\rightarrow i_a$，$k_h\rightarrow j_a$. 因此商品 j_a 可由 i_a 通过两步到达：$k_h\rightarrow i_a$ 的逆向链接和 $k_h\rightarrow j_a$ 的正向链接. 对于 $H_{i,j}$ 类似. 求出矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{H}$ 的主特征向量，就是对应的马尔可夫静态链；$\boldsymbol{A}$ 中值大的商品就是较重要商品.

SALSARank 算法没有 HAP 算法中相互加强的迭代过程，同时，只考虑直接相邻的商品对自身中心权重和权威权重的影响，而 HAP 算法是计算整个商品集合对自身中心权重和权威权重的影响. 因此，SALSARank 算法计算量远小于 HAP 算法.

4 实验结果分析与比较

为了评估上述算法，用 Java 实现了 HAP 算法、MPIS 算法和本文的两个算法. 在 2.7 GHz、256 M

机器上进行了实验.

4.1 合成数据库 **1**设置及实验结果

使用 IBM的标准合成数据生成器[9]产生带有强交叉因子的数据库 **1**. 该数据库的相关参数如下：商品数量为 100, 交易数量为 1 000, 平均每个交易长度为 10, 平均每个模式长度为 4. 频繁模式的个数为 1 000. 生成的合成数据库在最小支持度为 1%时，频繁项为 90个，频繁对数量为 3 340个.

同时为数据库 **1**中的每个商品随机产生与文献[8]相同的随机利润，即 10%的商品利润从 0.1元到 1元，80%的商品有 1元到 5元的中等利润，10%的商品利润为 5元到 10元的高利润. 商品的利润大致呈正态分布[8]，这里使用相对简单的随机平均分布. 在实验中使用合成数据库的全部交易利润为 26 470.

使用定义 1.1所述的评估利润方法. 合成数据库 **1**的结果如图 1和图 2所示. 从合成数据库 **1**的参数中看到，在实验设定的支持度(最小支持度为 1%)条件下，能找到 90%以上的频繁商品，即此时商品之间的交叉影响因子很强.

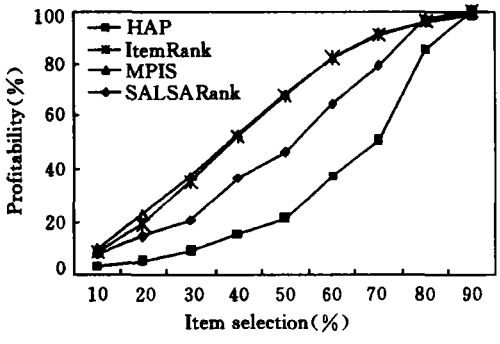

Fig. 1　Results of the synthetic dataset 1

Minsup = 1%.

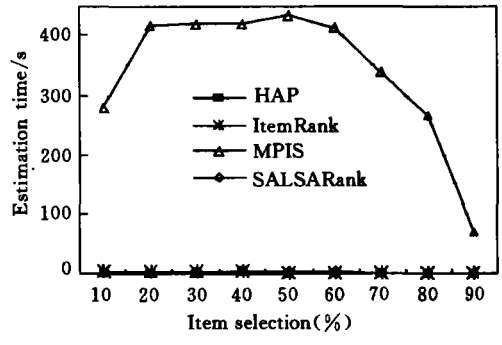

Fig. 2　The curve of estimation time *vs* item selection of the synthetic dataset 1

Minsup = 1%.

由图 1可以看出，对于交叉影响很强的数据库，MPIS算法能够取得最优解. ItemRank算法和 SALSARank算法都能取得比 HAP算法更好的解. 图 2表明，即使使用了各种方法和新的数据结构来提高效率，MPIS算法仍需要运行较长时间，而且运行的时间首先增加，然后逐渐减少. 但总体上 MPIS算法运行的时间较长. 对于更大的合成数据库，各种算法同样表现出各自的特性. 随着数据的增加，MPIS耗费的时间很长.

4.2 差数据库实验结果

为了测试上述算法的性能，给出差数据库 **2**[10]. 其中商品链接方式如图 3所示，称此时的数据库为差数据库 BadDB. 图 4为差数据库运行结果，BadDB将商品分成上下两层，每层商品与另一层商品没有任何交叉影响. 将两个商品组成商品对，每层一个. 例如，$\{I_a, I_b\}\{I_c, I_d\}$. 在每对商品中，前一个可能包含后一个，但后一个不能包含前一个. 即 $I_a \rightarrow I_b$ 的关联规则有很高的支持度，但 $I_b \rightarrow I_a$ 的支持

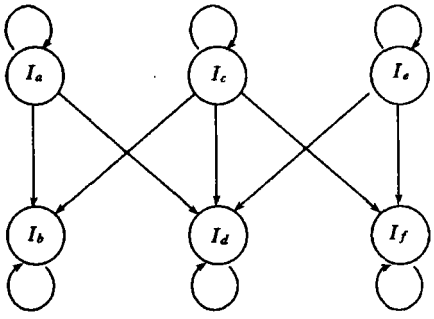

Fig. 3　Illustration of HAP worst-case dataset

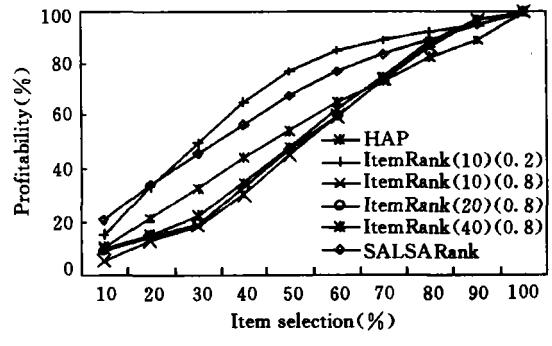

Fig. 4　Results of HAP worst-case dataset

Minsup = 0.05%.

度很低.

在数据库中,平均分配每个商品对.假设有1 000个商品和10 000个交易,那么有500个相邻链接对.对于每个{I_a, I_b}这样的商品对有20个交易集合.考虑第一个商品对{I_a, I_b},前10个交易包含I_a和I_b,后10个交易只包含I_b而不包含I_a.对于剩余的499个商品对也是如此,指定每20个交易的前后两部分.将所有交易对的前半部分称为前半组.随机地将I_b插入到前半组的80个交易中.使用这样的插入操作,从上层商品到下层商品之间有了弱链接.下层其他元素也使用同样的插入方式插入.

对于数据库中的商品,利润分布同上.当最小支持度为0.05%时,所有的商品都是频繁的;此时有2 194个频繁对.

由图3可以看出,下层商品出现了聚集现象[10].因此,HAP算法将选出下层的大部分商品.在选择结果中没有修正,使得选择的利润很小.由图4可见,HAP算法由于出现了聚集而得到了很差的结果.

图4中,ItemRank算法的名称后面给出了两个参数:首先是迭代次数,然后是衰减因子d的取值.ItemRank算法避免了这样的情况而得到了比HAP算法更好的结果;同时,ItemRank在迭代了10次并且衰减因子为0.2时取得了最好的结果.当衰减因子为0.8且选择商品数量少于70%时,则取得了比HAP差的结果.SALSARank算法也获得了比较好的结果.

综上可见,ItemRank算法和SALSARank算法均采用了随机购买模型;但是,ItemRank算法中商品只有一条链,SALSARank算法有两条马尔可夫链.ItemRank算法将商品权重从一个权威商品传递给另一个权威商品,而SALSARank算法通过两条随机链传递商品权重.上述算法只能处理静态数据,如何将上述算法应用到动态数据[11]中是今后需要研究的问题.

参 考 文 献

[1] Brijs T, Swinnen G, Vanhoof K, et al. The Use of Association Rules for Product Assortment Decisions: a Case Study [C]//Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining. San Diego: ACM Press, 1999: 254-260.

[2] WANG Ke, ZHOU Sen-qiang, HAN Jia-wei. Profit Mining: from Pattern to Actions [C]//Proc 2002 Int Conf on Extending Data Base Technology (EDBT'02). Prague: Czech Republic: Springer-Verlag GmbH, 2002: 70-87.

[3] ZHOU Sen-qiang, WANG Ke. The Encyclopedia of Data Warehousing and Mining: Profit Mining [M]. Hershey: Idea Group Reference, 2004.

[4] Kleinberg J, Papadimitriou C, Raghavan P. A Microeconomic View of Data Mining [J]. Knowledge Discovery and Data Mining, 1998, 2(4): 254-260.

[5] Wong Raymod Chi-wing, Fu Ada Wai-chee, WANG Ke. MPIS: Maximal-profit Item Selection with Cross-selling Considerations [C]//ICDM. Melbourne: IEEE Computer Society, 2003: 371-378.

[6] Brins S, Page L. The Anatomy of a Large-scale Hypertextual Web Search Engine [J]. Computer Networks and ISDN Systems, 1998, 30(1/7): 107-117.

[7] Lempel R, Moran S. The Stochastic Approach for Link-structure Analysis (SALSA) and the TKC Effect [J]. Computer Networks, 2000, 33(1/6): 387-401.

[8] WANG Ke, Su Ming-yen Thomas. Item Selection by "Hub-Authority" Profit Ranking [C]//SIGKDD. Edmonton, Canada: ACM Press, 2002: 652-657.

[9] Agrwwal R. IBM Synthetic Data Generator [EB/OL]. 2004-11-01. http://www.almaden.ibm.com/cs/quest/syndata.html

[10] Wong Raymod Chi-wing, Fu Ada Wai-chee, WANG Ke. Data Mining for Inventory Item Selection with Cross-selling Considerations [J]. Journal of Data Mining and Knowledge Discovery, 2005, 11(1): 81-112.

[11] WANG Zhe, ZHOU Chun-guang, ZHOU Dong-bin, et al. Clustering Data Streams of Two-tier Structure [J]. Journal of Jilin University (Science Edition), 2005, 43(3): 303-307. (王 喆, 周春光, 周东滨, 等. 双层结构的流数据聚类算法 [J]. 吉林大学学报(理学版), 2005, 43(3): 303-307.)

(责任编辑:赵立芹)