第16卷 第5期
2006年5月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.16 No.5
May 2006

用户行为模式挖掘问题的研究

刘洪辉[1,2],吴岳芬[2]
(1.华中科技大学 计算机科学与技术学院,湖北 武汉 430074;
2.湖南理工学院 计算机系,湖南 岳阳 414006)

摘 要:在软件可用性测试中,分析用户行为模式是一个关键的问题。为解决具有序列长度长、以序列片断为支持度计算依据等特点的用户行为模式挖掘问题,提出了一种有效的基于前缀树的频繁事件序列扩展方法,给出了比特图索引表的构造、事件扩展、事务扩展以及支持度计算的算法。使频繁事件序列能够简单快速地被确定。

关键词:事件序列;序列模式挖掘;用户行为模式挖掘;数据挖掘;软件可用性
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2006)05-0050-03

Research on User's Behavior Pattern Mining

LIU Hong-hui[1,2], WU Yue-fen[2]
(1.School of Computer Sci. and Tech., Huazhong Univ. of Sci. and Tech., Wuhan 430074, China;
2.Dept. of Computer, Hunan Institute of Sci. & Tech., Yueyang 414006, China)

Abstract: Analyzing a user's behavior pattern is a key problem in software usability testing. To solve user's behavior pattern mining problem, which is characterized as long sequence, calculation support according to sequence episode, an effective method based on extension of the frequent event sequence of the prefix tree are proposed. The algorithms for bit map index construction, event extension, transaction extension and support calculation are also presented. Making frequent event sequence be sured quickly and simply.

Key words: event sequence; sequential pattern mining; user's behavior pattern mining; data mining; software usability

0 引 言

在软件系统的使用过程中,对其进行可用性测试与改进能有效地提高软件的功能,用户使用软件系统时的人机交互行为从用户的观点反映了软件系统所应该具有的某种能力。结合序列模式挖掘[1]和软件可用性[2]而提出用户行为模式挖掘,通过对用户使用软件系统的事件序列集提取出最大频繁事件序列,从而从用户的角度来分析和提高软件系统的可用性。

用户行为模式挖掘具有序列长度长、以序列片断为支持度计算依据等特点。针对序列模式挖掘,文献[3]中提出了一种基于 Apriori 性质的算法 GSP;文献[4]中提出了基于投影数据库的算法 Freespan;文献[5]中提出了基于前缀投影的算法 prefixspan 等等,现有序列模式挖掘算法在挖掘用户行为模式时,不能很好地解决问题。从事件序列集中挖掘用户行为模式,快速有效地确定频繁事件序列是其关键。

收稿日期:2006-01-12

作者简介:刘洪辉(1967-),男,湖南岳阳人,讲师,硕士研究生,研究方向为数据挖掘、数据库与信息系统;导师:李国徽,教授,博士生导师,研究方向为现代数据库与信息系统。

1 相关概念和理论

给定一个事件序列集 B(见表1)和挖掘标准 ϕ,且 ϕ 具有反单调性,序列 α 在 B 中的支持度记为 support(α),在序列模式挖掘中,设定一个序列应该满足的最小支持度,记为 min_sup。

表1 事件序列集 B

标识号	事件序列
1	<(1,2),(2),(2)>
2	<(2),(2),(2)>
3	<(1,2),(2),(1)>

定义1 一个事件序列 $\alpha = < s_1, s_2, \cdots, s_n >$ 是另一个事件序列 $\beta = < t_1, t_2, \cdots, t_m >$ 的序列片断当且仅当存在整数 $1 \leqslant j \leqslant m$ 满足 $s_1 = t_j, s_2 = t_{j+1}, \cdots, s_n = t_{j+n-1}$,记为 $\beta \rhd \alpha$。

定义2 如果一个事件序列 $\alpha = < s_1, s_2, \cdots, s_n >$ 是另一个事件序列 $\beta = < t_1, t_2, \cdots, t_m >$ 的子序列,且 $s_1 \subseteq t_1 \wedge s_n \subseteq t_m$,则称 β 支持 α,记为 $\alpha \succ \beta$。如果 β 支持 α,且 β 中不再存在任何其他序列片断支持 α,则称 β 紧支持 α,记为 $\beta \succ_{sup} \alpha$。

定义3 如果事件序列 α 的支持度 support(α) = $|\{\gamma \mid \phi(\gamma) \wedge \gamma \succ_{sup} \alpha \wedge \beta \rhd \gamma \wedge \beta \in B\}|$ 大于或等于给定的 min_sup,其中 $|\cdot|$ 表示集合中元素个数,$\phi(\gamma)$ 成立表示 γ 满足挖掘标准 ϕ,则称之为频繁事件序列。

定义 4　用户行为模式的挖掘问题就是:从事件序列集中找出所有最大频繁事件序列。所谓最大频繁事件序列是指该频繁事件序列不是任何其他频繁事件序列的子序列,也称之为用户行为模式。

性质 1(反单调性[3])　频繁事件序列的子序列一定是频繁事件序列。

证明:假定 α 是一个频繁事件序列,δ 是 α 的子序列。由于 α 是频繁事件序列,故由定义 3 有 $\text{support}(\alpha) = |\{\gamma \mid \varnothing(\gamma) \wedge \gamma \succ_{\text{sup}} \alpha \wedge \beta \rhd \gamma \wedge \beta \in B\}|$ 大于或等于 min_sup。又由于 δ 是 α 的子序列,$\varnothing$ 具有反单调性,且 γ 紧支持 α,则不同的 γ 一定存在 γ' 满足 $\varnothing(\gamma') \wedge \gamma' \succ_{\text{sup}} \delta \wedge \beta \rhd \gamma' \wedge \beta \in B$。从而 $\text{support}(\alpha) \leqslant \text{support}(\delta)$。故 $\text{support}(\delta)$ 大于或等于 min_sup,即 δ 是一个频繁事件序列。

对一个事件序列而言,同一事务中的事件不需考虑它们之间的顺序关系,那么,一个事件序列可以表示成惟一的序列形式。一个长度为 k 的事件序列 S 的前 m 个事件 $(m \leqslant k)$ 对应的子序列称为事件序列 S 的前缀。

性质 2　所有长度为 $k+1$ 的频繁事件序列均可由长度为 k 的频繁事件序列通过增加一个事件或由一个事件构成的事务得到。

证明　假定 α 是任意一个长度为 $k+1$ 的频繁事件序列,δ 是 α 的前缀,且 δ 的长度为 k,即 α 由 δ 通过增加一个事件或由一个事件构成的事务得到。由于 α 是 δ 的前缀,故 δ 是 α 的子序列。又由于频繁事件序列具有反单调性,因而 δ 也是频繁事件序列。

序列集 B 对应的频繁事件序列前缀树中的节点满足:树的根节点为空事件序列;父节点对应的频繁事件序列是其所有子孙节点对应的频繁事件序列的前缀;深度为 k 的节点对应的频繁事件序列的长度为 k。

例如,针对表 1 给定的事件序列集合,假定最小支持度为 2,且规定出错事务个数最多为 0,则其对应的深度为 3 的频繁事件序列前缀树如图 1 所示。

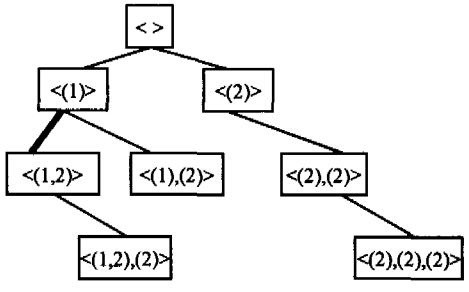

图 1　频繁事件序列前缀树

依据前缀树的定义,父节点与其子节点间的关系有两种:一种关系是父节点通过增加一个事件得到子节点,称子节点是父节点的事件扩展,如图 1 中粗线所示;另一种关系是父节点通过增加一个事务得到子节点,如图 1 中细线所示,称子节点为父节点的事务扩展。如果规定事件扩展和事务扩展的顺序,则事件序列集合对应的前缀树将是惟一的。由此,对于任何一个非空的频繁事件序列,均可由一个空的事件序列通过事件扩展和事务扩展得到,即频繁事件序列可通过事件扩展和事务扩展两种方式进行生长,从图 1 可看出根节点处只进行事务扩展不进行事件扩展。显然,最长的频繁事件序列仅存在于频繁事件序列前缀树叶子节点对应的频繁事件序列构成的集合中。这样,挖掘用户行为模式的问题可通过两阶段完成:先由事件扩展和事务扩展操作得到频繁事件序列前缀树所有叶子节点对应的频繁事件序列,再从中找出最大频繁事件序列。

2　基本数据结构

2.1　比特图索引表及其建立

从用户行为模式挖掘来看,频繁事件序列支持度的计算是由事件序列所包含的紧支持事件序列片断的个数来确定,而且要求事件序列片断必须满足一定的挖掘标准。故需要确定事件序列片断在序列中的位置,也即事件序列片断在事件序列中的起始位置和结束位置。一个事件序列中事件的位置可用下述方式来表示:事务的位置按出现先后用从 1 开始的连续自然数表示,同一事务中所有事件的位置则用事务的位置表示。事件序列片断的位置可由其中包含的事件在事件序列中的位置推导得到。为减少空间开销,将事件序列集中所有事件在事件序列中的位置采用比特图索引表的方式来描述。该索引表中的每一项保存的是一个 64 位的比特图(一般来说,一个序列的事务个数要小于 64),当某个事件出现在某序列的第 n 个事务中时,就将该比特图的第 n 位置 1。

算法 1:generateIndexList(B)/* 输入:B 为事件序列集;输出:indexList 为索引表 */

1) 初始化索引表 indexList;

2) $j = 1$;

3) 针对 B 中第 j 个事件序列的每个事件 k,将其位置信息送至索引表 indexList 的第 k 行第 j 列中的比特图中,并将其对应位置 1;

4) $j = j + 1$;

5) 重复 3) 至 4) 直到 B 中所有事件序列均处理完毕;

6) 返回 indexList;

该算法的空间复杂度为 $64 \times n \times m$,其中 n 为事件的个数,m 为序列数据集中序列的个数。

2.2　序列片断位置信息三元组

利用一个类来保存序列片断位置信息的内容,显然,序列片断的位置信息主要与 3 个量有关:序列号、序列起始位置、序列结束位置。类定于如下:

```
class location
{ int Id; //序列号
  int start; //序列开始位置
  int end;}; //序列结束位置
```

2.3　频繁事件序列队列

用一个队列来保存频繁事件序列,当访问到频繁事件序列的叶子节点的时候,说明该叶子节点的序列就是最大频繁事件序列,将其保存在一个一维队列 leaf 中。

3 序列扩展及支持度的计算

由性质 2 可知,任何频繁事件序列都可以看作由长度为 1 的序列经过序列扩展即事件扩展和事务扩展而得到。当一个频繁事件序列进行了事件扩展或事务扩展之后,就要判断该扩展序列是否为频繁事件序列。由性质 1 可知,如果扩展后的序列不是频繁事件序列,那么就没有必要在序列扩展后的序列后面进行事件扩展和事务扩展。要判断序列扩展后的序列是否为频繁事件序列,就要计算扩展后事件序列的支持度。

由上文的讨论可知,事件序列支持度的计算主要与两个因素有关:挖掘标准和紧支持。可以先将不符合挖掘标准的事件序列的片断信息删除,然后再计算其支持度。经过了序列扩展后,就要计算当前序列的支持度,如果大于或等于 min_sup,那么经过序列扩展后的序列为频繁事件序列。

3.1 事件扩展

算法 2: eventExtends(index,event)/* 输入:index 为保存序列片段位置信息的一维数组; event 为要增加的事件。输出:eindex 保存经事件扩展后对应事件序列的支持事件序列片断位置信息 */

1)初始化一维数组 eindex 为空;

2)从序列片段位置信息数组 index 中取出第一个序列片段位置信息三元组至 S;

3) 从 S 中取得序列编号 Id 和最后位置 end;

4)构造比特图 Map1,使 Map1 的第 end 位为 1,其它位为 0,取得 event 在序列编号为 Id 中的位置信息比特图 Map2;

5) Map1 和 Map2 做与运算得到的结果为比特图 Map,如果 Map 不为 0,则将 S 添加到 eindex 中,取出下一个序列片段位置信息三元组至 S,如果 S 不为空,转 3),否则返回 eindex。

3.2 事务扩展

算法 3: transactionExtends(index,event)/* 输入:index (同上);(event)为要增加的事务。输出:tindex 保存经事务扩展后对应事件序列的支持事件序列片断位置信息 */

1)初试化一维数组 tindex 为空;

2)如果 index 不为空,转 4);

3)从索引表 indexList 中分别提取出 event 在序列编号为 Id 中的位置信息比特图 Map,如果 Map 中有不为 0 的比特位 n,则分别将序列片段位置信息三元组(Id,n,n)加入到 tindex 中,返回 tindex;

4)从 index 中取出第一个序列片段位置信息三元组至 S;

5) 从 S 中取得序列编号 Id,开始位置 start 和最后位置 end;

6)构造比特图 Map1,使 Map1 的第 end + 1 位以前的比特位为 0,其它位都为 1,从 indexList 中取得 event 在序列编号为 Id 中的位置信息比特图 Map2;

7)Map1 和 Map2 做与运算得到的结果为 Map,如果 Map 不为 0,设 Map 中不为 0 的比特位为 n,则将序列片段位置信息三元组(Id,start,n)加入到 tindex 中,取出下一个序列片段位置信息三元组至 S,如果 S 不为空,转 5),否则返回 tindex。

从算法 2 和算法 3 可以看出,序列扩展的主要时间开销在于移位和与运算,经过了两次移位和一次与运算。

3.3 支持度的计算

算法 4:delete(index,size)/* 输入:index(同上);挖掘标准 ø 为:size + maxError≥序列片段的长度,为全局变量;输出:index; */

1)number = size;

2)从 index 中取出第一个序列片段位置信息至 Infor;

3)从 Infor 中取到序列片段的开始位置 start 和结束位置 end,如果 number + maxErrorend≥end − start + 1,转 4),否则将 Infor 从 index 中删除;

4)取出下一个序列片段位置信息至 Infor,如果 Infor 不为空,转 3),否则返回。

算法 4 的时间复杂度为序列片断位置信息的个数。

算法 5:audict(index)/* 输入:index(同上); 输出:number 支持度大小 */

1)令 number 为 0, i 为 1;

2) 取出 index 中的第 i 个序列片段位置信息的序列编号 Id1,开始位置 start1 和结束位置 end1, j = 1;

3)取出 index 中第 j 个序列片段位置信息的序列编号 Id2,开始位置 start2 和结束位置 end2;

4)如果 Id1 等于 Id2 且 start1 小于或等于 start2 且 end1 大于或等于 end2,则转 7);

5) $j = j + 1$,如果 j 小于或等于 index 的大小,转 3);

6)number + +;

7) $i = i + 1$,如果 i 小于或等于 index 的大小,转 2);

8)返回 number。

算法 5 的时间复杂度为: $O((\text{number}(\text{index}))^2)$。

4 结束语

文中提出了一种有效的频繁事件序列扩展方法,用户行为模式挖掘的整个时间开销主要在于支持度的计算。从如何减少时间复杂度和空间复杂度出发,引入比特图索引表,极大地方便了事件扩展和事务扩展。在处理序列片段位置方面,用序列片段位置信息三元组来表示,能快速地判断事件序列片段是否满足挖掘标准。

在上文讨论的基础上,随后可设计出完整的用户行为模式挖掘算法。

参考文献:

[1] Agrawal R,Srikant R. Mining Sequential Patterns[A]. In:

(下转第 55 页)

	相关	不相关
检出	R1/A	N1/B
未检出	R0/C	N0/D

检出相关文本和未检出不相关文本都是过滤正确的情况。而未检出相关文本意味着遗漏,检出不相关文本意味着错检。线性 utility 函数对这4种情况赋相应的权重:

Utility = A * R1 + B * N1 + C * R0 + D * N0

这里的 R1/R0/N1/N0 指的是每个主题4种文本的数量,A,B,C,D 决定了每种情况的代价。Utility 值越大,系统的过滤性能就越好。TREC10 中选择 A=2,B=-1,这样得到的指标称为 T10U[6]。

实验测试结果如表1所示。

表1　概念学习获取过滤模板测试结果

	相关文本数(法轮功/台独)	不相关文本数(法轮功/台独)
检出	54/37	25/18
未检出	12/10	39/65

在表1结果上的三项评测指标结果如表2所示。

表2　概念学习获取过滤模板评测指标结果

Precision(法轮功/台独)	Recall(法轮功/台独)	T10U(法轮功/台独)
0.6835/0.6727	0.8181/0.7872	110/111

将给出的主题描述作为用户模板,并将训练文本中对应特征项的平均评估值作为其各个主题词的权值,以此方法进行实验得测试结果如表3所示。

表3　主题描述作为过滤模板测试结果

	相关文本数(法轮功/台独)	不相关文本数(法轮功/台独)
检出	46/33	29/19
未检出	20/14	35/64

在表3结果上的三项评测指标结果如表4所示。

表4　主题描述作为过滤模板评测指标结果

Precision(法轮功/台独)	Recall(法轮功/台独)	T10U(法轮功/台独)
0.6133/0.6346	0.6969/0.7021	78/97

以上两种方法的评测指标结果对比如表5所示。

表5　两种方法的评测指标结果对比

	平均准确率	平均查全率	平均T10U
概念学习用户模板	0.6781	0.8027	111
主题描述用户模板	0.6239	0.6995	88

实验结果分析:在没有模板学习即系统不具有反馈功能的条件下进行试验并获得了上述结果,从结果来看,由概念学习一次性获取的过滤模板用于文本过滤系统,与直接使用主题描述作为用户模板的方法相比,过滤效果已有明显提高;另外,其相对于不同主题的平均准确率已接近70%,平均查全率已超过80%,过滤效果已比较令人满意,但距理想结果仍差距很大,如果能够在自适应过滤系统上进行实验,就有可能更大幅度地提高过滤性能。

2　结束语

基于内容的文本过滤包括3个基本环节[7]:确定用户的信息需求,即建立过滤模板;确定文本与过滤模板的匹配机制以及利用用户评注作为相关反馈动态改进过滤模板。其中如何建立过滤模板是一个关键问题。文中提出了一种基于概念学习的过滤模板获取方法,即通过对给定的少量训练文本进行概念学习来获取用户过滤模板。实验部分验证了此法具有一定的可行性。

参考文献:

[1] Mitchell T M. 机器学习[M]. 曾华军,张银奎译. 北京:机械工业出版社,2003.

[2] 李　凡,鲁明羽,陆玉昌. 关于文本特征抽取新方法的研究[J]. 清华大学学报(自然科学版),2001(7):98-101.

[3] 张鹏飞,李　贤,刘建毅,等. 基于相对词频的文本特征抽取方法[J]. 计算机应用研究,2005(4):23-26.

[4] 王　斌. TREC之文本过滤技术[R]. 北京:中科院计算所软件室,2001.

[5] 夏迎炬. 文本过滤关键技术研究[D]. 上海:复旦大学,2003.

[6] 赵　林,胡　恬,黄萱菁,等. 基于知网的概念特征抽取方法[J]. 通信学报,2004(7):46-54.

[7] 黄钢石,张亚非,陆建江,等. 基于NMF的用户模板构造方法[J]. 情报学报,2004(8):394-398.

(上接第52页)

ICDE 1995[C]. Taipei:[s.n.],1995.3-14.

[2] Nielsen J. 可用性工程[M]. 刘正捷译. 北京:机械工业出版社,2004.

[3] Srikant R, Agrawal R. Mining Sequential Patterns: Generalizations and Performance Improvements[A]. In: Proc. 5th Int. Conf. Extending Database Technology[C]. Avignon, France:[s.n.],1996.3-17.

[4] Han J, Pei J, Mortazavi-Asl B, et al. FreeSpan: Frequent pattern projected sequential pattern mining[A]. In: Proc. 2000 Int. Conf. Knowledge Discovery and Data Mining[C]. Boston, MA:[s.n.],2000.355-359.

[5] Pei J, Han J, Mortazavi-Asl B, et al. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth[A]. In: ICDE 2001[C]. Heidelberg, Germany:[s.n.],2001.215-224.