第32卷 第20期 计 算 机 工 程 2006年10月
Vol.32 № 20 Computer Engineering October 2006
·软件技术与数据库· 文章编号：1000—3428(2006)20—0084—02 文献标识码：A 中图分类号：TP311

一种基于粗糙集的Web用户访问规则获取方法

张 永，杨志勇
（兰州理工大学计算机与通信学院，兰州 730050）

摘 要：用户浏览模式获取是现阶段 Web 日志挖掘的主要目标之一。该文根据用户浏览的重要特征，提出了一种应用粗糙集理论获取规则的方法。选取重要特征作为条件属性并通过算法实现获取最终规则，实例分析效果良好。该方法的特点是只需要简单的数据预处理但可以获得简洁有效的访问模式。

关键词：Web 访问模式；粗糙集；数据挖掘

Rule Acquisition for Web Logs Based on Rough Set

ZHANG Yong, YANG Zhiyong
(School of Computer & Communication, Lanzhou University of Science and Technology, Lanzhou 730050)

【Abstract】 Presently, acquisition of user browsing is one of the major topics on Web Log mining. According to the important character of user browsing, a way of applying rough set acquisition rule is presented. Important character is selected as condition properties, then through arithmetic ultimate rule was acquired, instance analysis effect turns to be well. In this way, just in virtue of simple data pretreatment, simple and effective access pattern can be obtained.

【Key words】 Web access pattern; Rough set; Data mining

面对 Web 丰富的信息内容，巨大的数据量，发现用户的浏览兴趣模式对于 Web 管理员按照用户的爱好优化网站设计及电子商务等具有重要的意义。目前人们已经提出了一些用户浏览兴趣模式的挖掘算法：Chen[1]等首先将数据挖掘技术应用于 Web 服务器日志挖掘，发现用户的浏览模式。提出最大前向引用(maximal forward reference，MFR)系统的概念。将用户会话分割成一系列的事务，然后采用与关联规则的方法挖掘频繁的浏览路径。Myra Spiliopoulou 等人提出了一套类似 SQL 的挖掘语言 MINT，允许用户认为指定感兴趣的频繁路径的特点，然后根据用户的要求挖掘满足条件的结果。Han[2]等人将 Web 服务器日志保存为数据立方体(data cube)，然后在其上执行 OLAP 的各种操作，如提升、钻取等，用于用户发现的访问模式。但是这些挖掘方法都是要处理大量数据且并不一定得到有效的规则。考虑到粗糙集只需要处理少量数据就能取得很好的效果，本文提出了在粗糙集理论的基础上获取 Web 用户访问规则的新方法。

1 粗糙集理论的基本概念

粗糙集(rough set)理论是由波兰科学家 Zdzislaw Pawlak 于 1982 年最早提出的，主要用于对不确定性知识的表示、经验学习、知识分析、近似模型分类、不确定性推理、数据约简等[3]。

定义 1 决策表系统是一类特殊而重要的知识表达系统。多数决策问题都可以用决策表形式来表达。一个决策系统可表示为如下形式：

设 $S=(U,A,V,f)$，其中 U 为论域；$A=C\cup D$，$C\cap D=\phi$，C 称为条件属性集，D 称为决策属性集。$V=\cup V_a$，V_a 为属性 a 的值域；$f:U\times A\to V$ 是一个信息函数。

定义 2 在信息系统 S 中，对于每个属性子集 $B\subseteq A$，可以定义一个不可区分关系 $IND(B)$：

$$IND(B)=\{(x,y)\in U\times U:\forall b\in B, f(x,b)=f(y,b)\}$$

显然 $IND(B)$是一个等价关系，对象 x 在属性集 B 上的等价类$[x]_{IND(B)}$定义为

$$[x]_{IND(B)}=\{y:y\in U, yIND(B)x\}$$

为简便起见，在不产生混淆的情况下用 B 代替 $IND(B)$。

定义 3 在决策表 S 中，对于 $\forall x\in U$，用 dr_x 表示决策规则，即

$$dr_x: des([x]_C)\to des([x]_D)$$

其中 $des([x]_C)$ 表示对等价类$[x]_C$的描述，即等价类$[x]_C$对于各条件属性值的特定取值：$des([x]_D)$表示对等价类$[x]_D$的描述，即等价类$[x]_D$对于各决策属性值的特定取值；而对于 $\forall a\in C\cup D$，$dr_x(a)=a(x)$，$a(x)$为个体 x 关于属性 a 的属性值，且$dr_x|C$，$dr_x|D$ 分别称为 dr_x 的条件和决策。

2 规则生成预处理

运用粗糙集理论获取规则，首先需要建立决策表，然后再对决策表中的特征值用离散化数据表示。互联网上用户可以根据自己的兴趣爱好选择访问网站和网站上的页面，研究表明，多数用户访问具有以下几个特点：

(1)每次访问只有一个主题，并且都是从辅助页面开始浏览，最终目的是为了浏览一个内容页面；

(2)用户花在一个页面上的时间与该页面对用户是辅助页面还是内容页有关；

(3)访问过程中只有在改变访问主题时，才会访问前面访问过的页面以跳转到另外的页面；

(4)用户一次访问的时间都不会超过一个最大的限制——时间

基金项目：甘肃省自然科学基金资助项目(3ZS042-B25-014)
作者简介：张 永(1962－)，男，副教授，主研方向：数据库理论，数据挖掘；杨志勇，硕士生
收稿日期：2006-01-19 **E-mail：**zhiyoya@163.com.cn

窗口。

在特征属性的选择上考虑到条件属性重要性对决策规则的影响，我们选择以下 3 个最重要的特点作为条件属性。

(1)特征属性选择页面停留时间。因为时间是连续值，可以把时间分为若干区间，并且对每一个区间赋予不同的特征值。如分为 3 个段，建立如表 1 所示的映射表。

表 1 时间与属性值的映射

页面停留时间(min)	属性值
(0，5)	1
(5，10)	2
(10，∞)	3

从而 V_1={1,2,3}，不同的属性值反映用户的特征度不同。处理后的知识系统将使得一些用户合为一组，即那些具有相似访问模式的用户。

(2)特征属性选择用户访问跳转次数，即用户可能改变兴趣主题的次数。很明显用户在一次访问中有可能不改变主题，也有可能改变一次甚至更多。从而可以取其属性值为 V_2 ={0，1}，它们分别反映用户的访问兴趣。

(3)特征属性选择用户访问路径长度[4]。所谓访问路径就是把每一个页面作为一个节点，而访问的各节点就组成了一条路径。假设当路径节点数少于 5 时很少访问，其属性值为 0；如果大于 5，其属性值设为 1，可以得到 V_3={0, 1}。

对于用户访问网页的结果有可能相同，也有可能不同，如未发生交易行为，发生交易行为。因此选则用户的交易行为作为决策属性集 D={d}，其中 d={0, 1}。

3 算法实现

3.1 相关定义

为简便起见，该文主要讨论只有一个决策属性的决策表的属性值约简，即 D={d} (含多个决策属性的决策表的属性值约简可采用类似方法)，并以 $|X|$ 来表示集合 X 中的元素个数。为了能统一处理一致决策表，先给出如下定义[4]：

定义 4 在决策表 S 中，决策规则 dr_x 关于条件属性集 C 的一致程度 $\mu(dr_x,C)$ 定义为

$$\mu(dr_x,C)=\frac{|[x]_C\cap[x]_{[d]}|}{|[x]_C|}$$

显然，$0<\mu(dr_x,C)\le 1$，且 $\mu(dr_x,C)=1$ 时。dr_x 是一致的，否则是不一致的。

定义 5 在决策表 S 中，对于 $\forall x\in U$，用 $\partial_C(x)$ 表示 x 的广义决策，即

$$\partial_C(x)=\{v\in V_d:\exists y\in U\,(yIND\,C(x)\text{ 且 }d(y)=v)\}$$

由定义 6 可以看出：决策表 S 是一致的，当且仅当对 $\forall x\in U$，$|\partial_C(x)|=1$，否则就是不一致的。

定义 6 在决策表 S 中，若 $B\subseteq C$，如下列两个条件满足：

(1) $\partial_B(x)=\partial_C(x)$ 且 $\mu(dr_x,B)=\mu(dr_x,C)$；

(2)对 $\forall b\in B$，$\partial_B(x)\ne\partial_{B-\{b\}}(x)$ 或 $\mu(dr_x,B)\ne\mu(dr_x,B-\{b\})$。

则称 $dr_x|B$ 为决策规则 dr_x 的属性值约简，记为 $RED(dr_x)$。

定义 7 在决策规则 dr_x 所有属性值的约简中都存在的属性值称为 dr_x 的值核，dr_x 的所有值核构成一个集合：

$$CORE(dr_x)=\{dr_x(c):c\in C,\ \partial_C(x)\ne\partial_{C-\{c\}}(x)\text{ 或 }\mu(dr_x,C)\ne\mu(dr_x,C-\{c\})\}$$

3.2 算法描述

在决策规则中，有的属性不可缺少 ，有的规则可以删除，故利用 Rough 集方法对决策表进行属性值的约简，充分去除决策表中的冗余信息，从而得到更加简洁的决策规则。但每一个决策规则都可能存在多个属性值约简，同属性约简一样，求最小属性值约简和所有的属性值约简也是组合爆炸问题。基于此，采用启发式方法求取次优的属性值约简：以属性值的重要性作为启发式信息，并以值核为初始候选集合，之后选择重要性最高的属性值添加到候选集合中，判断当前候选集合是否为一个值约简，如此反复直到找到一个值约简为止。先给出属性值重要性的定义：

定义 8 在决策规则 dr_x 中，设 $a\in C-R\ (R\subset C)$，其对应的属性值 $dr_x(a)$ 的重要性 $SIG(dr_x(a),R)$ 定义为[5]

$$SIG(dr_x(a),R)=\mu(dr_x,R\cup\{a\})-\mu(dr_x,R)$$

其中，若 $R=\phi$，则令 $\mu(dr_x,R)$=1。

下面给出该算法[6]的描述：

输入 决策表 S =<U,$C\cup\{d\}$,V, f >。

输出 约简后的决策规则。

步骤 1 根据定义 7 计算 dr_x 的所有值核，即 $CORE(dr_x)$，设对应的属性集为 P；

步骤 2 $RED(dr_x)=CORE(dr_x)$，R=P；

步骤 3 若 $\partial_Q(x)\ne\partial_R(x)$ 或 $\mu(dr_x,Q)\ne\mu(dr_x,R)$，则反复执行：

(1)在 $Q\sim R$ 中找出使 $SIG(dr_x(a),R)$ 取最大值的属性 a；

(2)将 a 加入到 R 的尾部，$RED(dr_x)=RED(dr_x)\cup\{dr_x(a)\}$；

步骤 4 从 R 的尾部开始，往前对每个属性 a 进行判断：若 $a\in P$，则从 a 开始往前的属性对应的值都是核，$RED(dr_x)$ 就是 dr_x 的属性值约简，跳出步骤 4；否则若 $\partial_Q(x)=\partial_{R-\{a\}}(x)$ 且 $\mu(dr_x,Q)=\mu(dr_x,R-\{a\})$，则说明属性值 $dr_x(a)$ 是可以除的，从 $RED(dr_x)$ 中把 $dr_x(a)$ 删除；

步骤 5 根据各决策规则的属性值约简输出对应的简化后的决策规则。

该算法以值核为起点，步骤 3 确保一定能找到 Q 的某个子集 R，满足 $\partial_Q(x)=\partial_R(x)$ 且 $\mu(dr_x,Q)=\mu(dr_x,R)$；步骤 4 通过一个后向删除的过程将 $RED(dr_x)$ 中可以去除的属性值删除，重要性越低的属性值，越早被处理，从而确保处理后的 $RED(dr_x)$ 中每个属性值都是不可删除的，最后得到的 $RED(dr_x)$ 一定时属性值约简。所以该算法对于计算决策规则的某个属性值约简来说是完备的。

4 实例分析

算法实现选择的决策表的条件属性 C={a,b,c}，a 表示用户在页面上停留的时间，b 表示用户访问跳转次数，c 表示用户访问路径长度；决策属性为 D={d}，d 为页面访问结果。表 2 为用户访问数据经过预处理所建立起来的决策表。

表 2 预处理后的决策

U	a	b	c	d
1	1	0	0	0
2	1	0	1	0
3	2	0	1	1
4	2	1	0	0
5	2	1	0	1
6	3	0	0	0
7	3	1	0	0
8	3	1	0	1
9	3	1	1	1

容易看出，表 2 中决策规则 4、规则 5、规则 7、规则 8 是不一致的，而其它是一致的。然后利用算法对表 2 进行属性值约简，可以获得决策规则的值约简如表 3 所示。最后可得决策规则集：

(下转第 146 页)

为该序列异常。显然，LFC 阈值 τ 的选取非常关键，本文采用 Warrender 中的办法[2]，阈值 τ 取实报率达到 95%以上时的最大 LFC 值。表 3 数据来自新墨西哥大学和 DARPA 数据，Forrest[1]的算法采用的滑动窗口长度为 6，实验结果如表 4。

表 3 实验数据

进程	攻击序列	正常数据 所有数据		正常数据 训练数据		正常数据 测试数据	
	进程数	进程数	调用号数	进程数	调用号数	进程数	调用号数
MIT lpr	1001	2 703	2 926 304	415	568 733	1 645	1 553 768
UNM lpr	1001	1231	434 303	250	86 652	981	347 651
login	8	9	8 887	9	8 887	–	–
ps	21	24	6 144	24	6 144	–	–
stide	105	13 726	15 618 237	200	246 750	13 526	15 185 927
samba	4	12	40 834	5	15 396	7	25 438
httpd	33	18	20 502	6	7 001	12	13501

表 4 实验结果

进程	本文算法 模式集规模	误报率	实报率	阈值	Forrest[1] 模式集规模	误报率	实报率	阈值
MIT lpr	109	2.5%	100%	2	452	0.43%	95.2%	6
UNM lpr	60	0%	100%	X	280	0%	100%	X
login	65	–	75%	X	375	–	75%	14
ps	34	–	100%	9	173	–	100%	4
stide	30	0.08%	95.24%	12	150	0.25%	95.24%	4
samba	81	0%	100%	19	448	0%	100%	5
httpd	10	0%	100%	X	40	0%	100%	X

表 4 的阈值一列中，数字表示实报率达到 95%以上的最大阈值，X 表示可取任意阈值。可以看出，对多数进程，至少有一个阈值可以使得实报率高于 95%，而误报率相比是很低的。另外，本文算法的分析速度也是很快的。在训练过程中，每秒平均可分析 23 000 多个调用号（Forrest 的训练方法只有 1 250 个/s）。有时候，尽管模式集规模比较大，但由于模式之间的转移关系比较简单，即 N 中没有很长的数组，算法分析速度仍然是很快的。

4 总结

基于程序的结构特征，本文提出了一种快速的不定长序列模式寻找算法，并首次将模式间次序关系引入到了匹配算法中。实验结果表明，本文算法相比其它基于系统调用短序列的算法，可在入侵序列中捕捉到更多异常信息。并在保持一组规模很小的模式集的情况下，取得了很低的误报率和漏报率。

参考文献

1 Forrest S, Hofmeyr S A, Somayaji A, et al. A Sense of Self for Unix Processes[C]. Proceedings of the IEEE Symposium on Security and Privacy, 1996:120-128.

2 Warrender C, Forrest S, Pearlmutter B. Detecting Intrusions Using System Calls: Alternative Data Models[C]. Proceedings of the IEEE Symposium on Security and Privacy, 1999: 133-145.

3 Lee W, Stolfo S J. Data Mining Approaches for Intrusion Detection[C]. Proceedings of the 7th USENIX Security Symposium, San Antonio, Texas, 1998.

4 Eskin E, Lee W, Stolfo S J. Modeling System Calls for Intrusion Detection with Dynamic Window Sizes[C]. Proceedings of DARPA Information Survivability Conference & Exposition II, 2001.

5 Kosoresow A P, Hofmeyer S A. Intrusion Detection via System Call Traces[J]. IEEE Software, 1997, 14(5): 35-42.

6 Hofmeyr S A, Forrest S, Somayaji A. Intrusion Detection Using Sequences of System Calls[J]. Journal of Computer Security, 1998, 6(3): 151-180.

7 Wespi A, Dacier M, Debar H. Intrusion Detection Using Variable-length Audit Trail Patterns[C]. Proceedings of Workshop on Recent Advances in Intrusion Detection, Toulouse, France, 2000.

(上接第 85 页)

a=1 ⟶ d=0，一致程度为 1。
a =2 且 b =0 ⟶ d =1，一致程度为 1。
a =2 且 c=0 ⟶ d =0，一致程度为 1/2。
a =2 且 b =1 ⟶ d =1，一致程度为 1/2。
a =3 且 b =0 ⟶ d =0，一致程度为 1。
b=1 且 c=0 ⟶ d =0，一致程度为 1/2。
b =1 且 c=0 ⟶ d =1，一致程度为 1/2。
a =3 且 c=1 ⟶ d =1，一致程度为 1。

表 3 属性值约简

U	a	b	c	d
1	1	×	×	0
2	1	×	×	0
3	2	0	×	1
4	2	×	0	0
5	2	1	×	1
6	3	0	×	0
7	×	1	0	0
8	×	1	0	1
9	3	×	1	1

通过算法对决策表进行属性值的约简，即能够得到更为简洁的决策规则，但是却只需要少量的数据处理。同其它规则相比较，获取方法将节省大量的资源和时间，是值得考虑的方法。

5 结束语

本文提出了基于粗糙集的网络用户访问规则的获取新方法，该算法的优势就在于在处理少量数据的基础上就能得到更为简洁的决策规则。实例分析验证了该算法的可行性，并能很好地处理一致决策表和不一致决策表。需要进一步研究的是取得更为有效的特征属性及其数值。

参考文献

1 Chen M S, Park J S, Yu P S. Efficient Data Mining for Path Traversal Patterns[J]. IEEE Trans. on Knowledge and Data Engineering, 1998, 10(2): 209-221.

2 Han J, Kamber M. 数据挖掘: 概念与技术[M]. 北京: 机械工业出版社, 2001.

3 张文修, 吴伟志. 粗糙集理论与方法[M]. 北京: 科学出版社, 2001.

4 Pawlak Z. Rough Set[J]. International Journal of Computer and Information Science, 1982, 11(5): 341-356.

5 王国胤. Rough 集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001.

6 刘少辉, 盛秋戬, 吴斌等. Rough 集高效算法的研究[J]. 计算机学报, 2003, 26(5): 524-529.