一种改进的模糊关联算法及其在IDS中的应用

曾庆花,王文国
(曲阜师范大学 计算机科学学院,山东 日照 276826)

摘 要:关联规则的发现是数据挖掘中的一个重要问题,但只是对离散型数据进行处理。为解决连续数量值属性的划分出现的"尖锐边界"问题,采用模糊划分,实现数据平滑过渡。由于入侵检测系统(IDS)对训练数据要求不高,文中提出了一种使用哈希链表改进模糊关联规则挖掘的新算法,且在挖掘过程中使用了等价类快速查找频繁项集,避免了反复扫描数据库及大量重复计算检验步骤。通过一个入侵检测系统的算例显示了其优越性,来提高对入侵数据的识别能力。

关键词:模糊关联;入侵检测系统;哈希链表;等价类
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2007)07-0236-04

An Improved Algorithm of Fuzzy Association Rules and Its Application in IDS

ZENG Qing-hua, WANG Wen-guo
(Dept. of Computer Science, Qufu Normal University, Rizhao 276826, China)

Abstract: Discovery of association rule is an important problem in database mining, but it is merely used to handle the discrete data. To partition continuous quantitative attribute is handled by using fuzzy partition in order to solve the problem of sharpening boundary, which provides a smooth transition of data partition. On IDS the requirements of training data are very low. In the paper, an improved algorithm using Hashing tables on mining fuzzy association rules is proposed, and equivalence classes are introduced to search frequent itemsets quickly. With this algorithm the usual practice of repeatedly database scanning can be avoided. Its efficiency is showed with a typical use on intrusion detection system (IDS) from network datasets.

Key words: fuzzy association rules; IDS; Hash chain table; equivalence class

0 引 言

随着网络安全技术的发展,入侵检测系统(IDS)越来越受到人们的重视。在有关入侵检测的各项技术中,如何提高入侵检检测系统的自适应性以及对未知入侵的识别能力是当前的研究热点之一。数据挖掘技术能从大量数据中提取有用的信息,因此在IDS中获得了广泛应用。但是,由于数据挖掘通常只能对离散值处理,这就导致所谓"尖锐边界"问题。为了解决这一问题,人们提出了基于模糊集理论的模糊数据挖掘技术[1~5]。

文中将提出一种改进的模糊关联算法,并将其应用于IDS之中,以提高其对入侵数据的识别能力。

收稿日期:2006-09-13
基金项目:国家人事部高层次留学人员回国工作资助项目(国人部发[2004]61号)
作者简介:曾庆花(1981-),女,山东曲阜人,硕士研究生,研究方向为网络信息安全;王文国,博士,教授,研究方向为网络通信与信息安全。

1 关联规则

关联规则是数据挖掘常见的一种形式,其目的是发现隐在数据间的相互关系。令 $D = \{t_1, t_2, \cdots, t_k, \cdots, t_n\}$, $t_k = \{i_1, i_2, \cdots, i_j, \cdots, i_p\}$ 为一条事务;t_k 的元素 $i_j (j = 1,2,\cdots,p)$ 称为项目。设 $I = (i_1, i_2, \cdots, i_m)$ 是 D 中全体项目组成的集合,I 的任何子集 X 成为 D 中的项目集(itemset),$|X|_{=k}$ 称集合 X 为 k 项目集,事务数据库的每一数据项集支持度 $\sup(X) = |X(T)|/|D|$,其中 $|X(T)|$ 和 $|D|$ 分别表示集合 $X(T)$ 和 D 的元素个数,$X(T) = \{T \in D \mid X \subseteq T\}$,如果符合 $\sup(X) \geqslant \min\ \sup$,则称频繁项目集。关联规则的置信度为 $\mathrm{conf}(X \Rightarrow Y) = \sup(X \cup Y)/\sup(X)$。一条关联规则形如 $X \Rightarrow Y$ 的蕴涵式,其中 $X \subseteq I$, $Y \subseteq I$,且 $X \cap Y = \varnothing$。

关联规则挖掘可以分解为以下两个子问题:

(1) 根据 $\sup(X) \geqslant \min\ \sup$ 找出 D 中所有频繁项目集;

(2) 根据生成的频繁项目集和 $\mathrm{conf}(X \Rightarrow Y) \geqslant$

minconf 产生关联规则。

Apriori 是挖掘布尔型频繁项目集的算法,采用逐层搜索迭代的技术。Apriori 寻找频繁项目集要对数据集进行多步处理:第一步找出一维频繁项目集,从第二步开始处理直到再没有频繁项目集生成。到第 k 步循环时,用 Apriori - gen 函数产生候选项目集 C_k,然后对数据库进行搜索,得到候选项目集 C_k 的支持率。接着通过与最小支持率比较,找出 k 维频繁项目集 L_k。在 Apriori - gen 函数中需要用到连接(join)步和剪枝(prune)步。通过剪枝步可以缩小候选项目集 C_k 中的项目数,从而避免求余项目的支持率。尽可能减少项目集中的项目数是对 Apriori 算法进行优化的一个基本出发点。

2　模糊关联

设论域 U 上,映射 $\mu_A: U \to [0,1], \forall X \in U$, $\mu_A(X)$ 为 A 上的隶属度。对 A, B 两个模糊集,$(A \cap B)(U) = \mu_A(X) \wedge \mu_B(X) = \min\{\mu_A(X), \mu_B(X)\}$。对于 $\forall \lambda \in [0,1]$,定义 $A_\lambda = \{\mu_A(X) \geqslant \lambda\}$ 为模糊集 A 的 λ 截集;若 $A_\lambda = \{\mu_A(X) > \lambda\}$ 则为 λ 强截集。

令数据库 $D = \{t_1, t_2, \cdots, t_k, \cdots, t_n\}$, $I = (i_1, i_2, \cdots, i_m)$ 为 D 中属性集,要发现的模糊关联规则是形如以下蕴涵式:$< X, A > \Rightarrow < Y, B >$。其中,$X = \{x_1, x_2, \cdots, x_n\}$, $Y = \{y_1, y_2, \cdots, y_n\}$ 且 $X \cap Y = \varnothing$, $A = \{f_{x1}, f_{x2}, \cdots, f_{xp}\}$, $B = \{f_{y1}, f_{y2}, \cdots, f_{yq}\}$ 分别是与 X, Y 中属性相应的模糊集集合。序偶集 $< X, A >$ 表示 X 中的属性取 A 中的相应值。

定义1　序偶集 $< X, A >$ 表示“项集 - 模糊集”,X 是属性 x_i 的集合,A 是模糊集 a_j 的集合,则支持度:$\sup(< X, A >) = \sum_{i \in D}(\wedge_{x_j \in X}\{\alpha_{a_j}(d_i[x_j])\})/|D|$ 式中 $\alpha_{a_j}(d_i[x_j]) = \mu_{a_j \in A}(d_i[x_j])$;$d_i[x_j]$ 为第 i 事务中 x_j 的值。若 $\sup(< X, A >) \geqslant$ minsup,则称为频繁序偶集;若 X 包含 k 个属性,则 $< X, A >$ 是频繁 k - 序偶集。

定义2　频繁集 $< Z, C >$ 生成规则 $< X, A > \Rightarrow < Y, B >$: if X is A then Y is B,其中 $X \subset Z$, $Y = Z - X$, $A \subset C$, $B = C - A$。则置信度 $\text{conf}(< X, A > \Rightarrow < Y, B >) = \sup < X \cup Y, A \cup B > / \sup < X, A >$。若 $< X, A >$, $< Y, B >$ 都是频繁序偶集,且 $\text{conf}(< X, A > \Rightarrow < Y, B >) \geqslant$ minconf,则称 $< X, A > \Rightarrow < Y, B >$ 是强模糊规则。

3　模糊关联挖掘算法的改进

模糊关联 Apriori 算法的瓶颈在于:

1) 对整个数据库进行多次访问;

2) 识别频繁项目集及连接步骤算法采用模式匹配,效率比较低;

3) 计算支持度需要重复计算。

文献[5]提出利用模糊隶属矩阵方法来表示事务信息,虽然该方法能提高效率,但是事务数据矩阵中有大量零元素和一些支持度比较小的元素存在(挖掘没有意义)。实际矩阵为稀疏矩阵,计算支持度时都要进行扫描,而且多项“与”运算实际上是重复运算,例如 AB 是 $A \wedge B$, ABC 是 $A \wedge B \wedge C$。

针对这些问题对算法进行了改进。主要办法是采用新的数据结构——Hash 链表将所有关键字为同义词的记录存储在同一线性表中,从而在 H(key)和 key 之间建立一种确定的对应关系,这样就方便了查找。新算法首先对数据库进行访问,通过计算属性序列得到 1 - 频繁项目集,将 1 - 频繁项目集中各个项的事务信息通过 Hash 链表进行存储,其中的节点包含两部分:事务号 Tid 和此事务下该项目属性对应的隶属度 μ。如果此隶属度 μ 为零或者在截集范围外,该事务信息就不被存储。为每个项集建立的 Hash 链表,关键字是事务号,通过扫描 Hash 链表 i 的关键字,到 Hash 链表 j 中去查找是否存在;若存在则隶属度进行取 min 运算,否则 Hash 链表 i 继续扫描,直到没有关键字为止。通过 $k-1$ - 频繁项目集连接而生成 k - 候选项目集,且具有相同事务的隶属度取 min 运算,同时将新建一个 Hash 链表,头节点是两个项集连接之后的 k - 项目集,而链表节点是相同事务号以及此事务下的最小隶属度。通过临时变量计算最小隶属度之和,比较是否不小于 minsup,如果是,就保留新建的链表,否则就删除之。$k-1$ - 频繁集链表生成 k - 频繁集后就完全删除。

上面的运算只在计算频繁 1 - 项目集时需要对整个数据库进行访问,而此后在计算候选 k - 项目集支持度时,仅需要数据库中频繁 $(k-1)$ 项目集合的信息即可,而随着 k 的增大,频繁 $(k-1)$ 项目集的数目不断减小,因此需要访问的数据量也不断减少。另外,利用 $k-1$ 项目集的结果来计算 k - 项目集的支持度,避免了重复计算。如已知 $L_2 = (\{AB\}, \{AC\}, \cdots)$,要计算 $C_3 = (\{ABC\}, \cdots)$,直接计算 $AB \wedge BC$ 即可。还有通过模糊集的截运算可把支持度低的数据去掉,对于隶属度为零的数据在项目集求交时可以省略。因为隶属度为零的事务在后来的 Hash 链表中不存在,所以不需要运算。

定理　频繁项集的所有非空子集一定也是频繁的。非频繁项集的超集一定不是频繁项集。

性质　设事务集 D,模糊模式为 P,事务集中事

务 T 对模式 P 的有效支持度是 $\sup(P/D) = \sum_{r\in A_\lambda} S(P,T)/|D|$[2]。

证明：模糊模式 P 在事务集 D 中的支持度 $\sup(P/D) = \sum_{r\in D} S(P,T)/|D|$，根据定义，它是全体事务对 P 支持的平均值，反映的是 P 在事务集上具有的总支持度。A 模式只有少数事务对它有非常高的支持，B 是一般支持模式，C 模式虽然有很多事务都对它有一定的支持，但是支持的有效性比较低，但 $\sup(B/D) = \sup(A/D) = \sup(C/D)$，显然对于 A、C 两种模式进行挖掘是毫无意义的，应该去除。引入 S 的一个阈值函数 $\beta(S)$，定义如下：当 $S < \beta(S)$，$\beta(S) = 0$；当 $S \geqslant \beta(S)$，$\beta(S) = S$。容易看出有效支持度是计算模糊集 A 属于其截集 A_λ 的模糊支持。

优化步骤：等价类具有与频繁集(定理) 相似特性，即某个等价类的子集必定也是等价类，如果某个项集不是等价类则它的更高一级的项集也不是等价类。设 R 为定义在集合 A 上的一个关系，R 是自反的、对称的和传递的，则 R 为等价关系。对于 $\forall a \in A$，$[a]R = \{x \mid x \in A, aRx\}$ 为 a 形成的 R 等价类[4]。根据等价关系和等价类概念，设 R 为集合 L_k 上具有相同 $k-1$ 长度的前缀关系，显然 R 是 L_k 上的等价关系，通过 R 关系可以确定 L_k 上一个划分，其中每一个分块即为 L_k 集合的 R 等价类。记作：$[a] = \{b[k] \mid a[1:k-1] = b[1:k-1]\}$。当 $k = 1$ 时，去除同一属性的按序连接；当 $k \geqslant 2$ 时，将前缀串 a 与 $[a]$ 集合中每个元素(除去最后一个) 连接形成候选 C_k 的等价类。即 $[ab] = [a] \cap [b]$。再将前缀串 ab 与 $[ab]$ 集合中的每个元素连接即得到 C_k。这种方法避免了连接时的模式匹配以及同一属性项集的检查(项来自相同串等价类中的一个元素)。在这种操作下连接后形成的 C_k 不进行剪枝(即 $k-1$ 频繁子集的检查)，因而省略了反复检验步骤，节省了时间。

根据上述思想，对模糊 Apriori 算法进行了改进：

算法：利用模糊集概念产生频繁项集。

输入：事务数据库 D；隶属度函数 F_k；最小支持度 minsup。

输出：频繁项集。

```
L1 = ∅;为每一项建立哈希链表;
for 数据库 D 的每个项(即属性列){ /* 生成 1 - 频繁集及其哈希链表 */
int w = 0;
for 每个事务 ∈ D{
将数据经过隶属函数 Fk 转换成模糊隶属度 μ;
  if μ >= λ insert(对应项哈希链表,事务号,隶属度 μ);
  w += μ; }
  if(w/n >= minsup)L1 = L1 ∪ {项};
else
删除非频繁项的哈希链表; }
for(k = 2;Lk-1 ≠ ∅;k ++) {
Ck = Apriori_ gen_ equ(Lk-1);
for 每一个 c ∈ Ck+1{
if(compute(c,support) >= minsup)
Lk+1 = Lk+1 ∪ {c};
  else delete hashlist c;
                    }
  删除 k - 1 频繁集 hashlist;
  }
  int compute(c,support){ /* 计算项集的支持度 */
  create(hashlist c);
  取前 k - 1 项 s,后 k - 1 项 j;
  for hashlist s 中的每一个关键字{
  if(查找 hashlist j 的关键字)
  insert(hashlist c,关键字,min(关键字节点对应 μ 的大小));
  support += min(关键字节点对应 μ 的大小);}
  return support = support/n;
  }
  Apriori_ gen_ equ(Lk-1:frequent(k - 1) - items ets){
    if k = 2 按序连接 Lk-1 集合中去除同一属性的项;else {
  将 Lk-1 按相同前缀划分生成等价类 Lk-1/R,并将集合元素记数 ki;
  if k = 3 存储 L2/R 集合;
  for 任一等价类 Lk-1/R {
  if ki <= 1 continue;
  else for(int i = 0;i < ki - 1;i ++){
  [Ck 前缀 s] = [等价类 Lk-1/R] ∩ [此前缀下 i 元素];
  c = s + [s]集合中每一个元素 j;//连接字符串;
  Ck = Ck ∪ c;}
  }
```

4 在 IDS 中运用模糊关联示例

IDS 一般分为两类：异常检测和滥用检测。在异常检测中运用模糊关联规则挖掘的具体思路是：建立

系统正常模式下的关联规则集 S_1，然后挖掘系统在某暂态模式下的关联规则集 S_2,再计算两规则集的相似度 similarity(S_1,S_2),即用相似度来表示系统当前状态与正常状态的背离程度,以确定系统是否处于异常状态。

给定两个关联规则 $R_1: X \rightarrow Y, c, s, R_2: X' \rightarrow Y', c', s'$。如果 $X = X'$ 且 $Y = Y'$,两规则之间的相似度 similarity(R_1,R_2) = max[0,1, − max[| $c - c'$ | /c, | $s - s'$ | /s]], 两规则集 S_1、S_2 的相似度为 similarity(S_1,S_2) = μ^2/ | $S_1 \| S_2$ |, 其中,μ = $\sum$similarity(R_1,R_2), | $S_1 \| S_2$ | 为规则的数量。

而在滥用检测中运用模糊关联思路是将挖掘出的规则进行与已知匹配,对入侵行为进行识别。

在表 1 算例中网络流数据通过 F − 均值聚类方法求出隶属函数 F_k,然后转换为模糊数据。

取 $\lambda = 0.5$,minsup = 0.25,表 1 得到 L_1 = ({$A.L$},{$A.H$},{$B.L$},{$C.H$},{$D.H$}),然后按序连接通过扫描频繁项目的哈希链表得到 L_2,结果如表 2。

表 1　模糊数据集

	Ptcp		Pudp		Avgpacket/s		AvgMibt/s	
Tid	$A.L$	$A.H$	$B.L$	$B.H$	$C.L$	$C.H$	$D.L$	$D.H$
1	0	0.90	1	0	0	0.65	0	1
2	0	0.89	1	0	0	0.73	0	1
3	0	0.57	1	0	0	1	0	1
4	0	0.93	0	1	0	0.72	0	0.86
5	0	0.86	0.5	0	0.89	0	1	0
6	0	0	0	0.5	0	0.62	0	1
7	0	1	0.5	0	0	0.91	0	1
8	1	0	1	0	0	1	0	1
9	0.64	0	1	0	0.63	0	1	0
10	1	0	1	0	0	1	0	0.89
sup	0.26	0.51	0.7	0.15	0.15	0.66	0.2	0.78

表 2　等价类及模糊关联规则的计算

L_2	Sup
$A.H, B.L$	0.34
$A.H, C.H$	0.36
$A.H, D.H$	0.42
$B.L, C.H$	0.49
$B.L, D.H$	0.54
$C.H, D.H$	0.65

L_3	Sup
$A.H, B.L, C.H$	0.25
$A.H.B.L, D.H$	0.29
$A.H, C.H, D.H$	0.36
$B.L, C.H, D.H$	0.48

由表 2 中 L_2/R 得到等价类为:

$[A.H] = \{B.L, C.H, D.H\}$,$[B.L] = \{C.H, D.H\}$,$[C.H] = \{D.H\}$,$[A.H, B.L] = [A.H] \cap [B.L] = \{C.H, D.H\}$,$[A.H, C.H] = [A.H] \cap [C.H] = \{D.H\}$,$[B.L, C.H] = [B.L] \cap [C.H] = \{D.H\}$,$C_3$({$A.H, B.L, C.H$}, {$A.H, B.L, D.H$}, {$A.H, C.H, D.H$}, {$B.L, C.H, D.H$})。同理,可求 C_4。

取 minconf = 0.60 可挖掘出如下强模糊规则:

$A.H \wedge B.L \rightarrow C.H$　$s = 25\%$　$c = 74\%$

$A.H \wedge B.L \rightarrow D.H$　$s = 29\%$　$c = 85\%$

$A.H \wedge D.H \rightarrow B.L$　$s = 29\%$　$c = 70\%$ ……

挖掘出 $A.H \wedge B.L \rightarrow D.H$ 即“网络流中 TCP 包数量为高,且 UDP 包为低,则每秒平均数据位为高。”

5　结束语

文中运用模糊集理论中的截运算,采用 Hash 链表结构实现快速定位查找,减少了支持度的重复计算,且通过等价类优化算法避免了多余模式匹配。该算法的不足之处是新建哈希链表本身增加了一定的空间和时间开销。下一步工作是通过时间序列的模糊关联挖掘来识别入侵事件。

参考文献:

[1] Han Jiawei, Kamber M. 数据挖掘:概念与技术[M]. 范明,孟小峰译. 北京:机械工业出版社,2002.

[2] 张保稳,何华灿. 有效支持度和模糊关联规则挖掘[J]. 小型微型计算机系统,2002(9):1004 − 1006.

[3] Park J S, Chen M S, Yu P S. An effective hash − based algorithm for mining association rules[C]// In Proc. 1995 ACM Int. Conf. Management of Data. San Jose, CA: [s. n.], 1995:175 − 186.

[4] 王　翔,袁兆山. 基于等价类和最大完全图集聚类的关联规则发现算法[J]. 小型微型计算机系统,2000(6):614 − 616.

[5] 孙建勋,陈绵云,张曙红. 用模糊方法挖掘量化关联规则[J]. 计算机工程与应用,2003(18):190 − 192.

(上接第 235 页)

参考文献:

[1] 刘志俭. MATLAB 应用程序接口用户指南[M]. 北京:科学出版社,2000.

[2] 苏金明,黄国明,刘　波. MATLAB 与外部程序接口[M]. 北京:电子工业出版社,2004.

[3] 飞思科技产品研发中心. MATLAB7 基础与提高[M]. 北京:电子工业出版社,2005.

[4] 张友兵,田漫柳. 基于 Matcom 与 VC 混合编程的数字图像处理方法研究[J]. 湖北汽车工业学院学报,2005,19(1):38 − 41.

[5] Sonka M, Hlavac V, Boyle R. Image Processing, Analysis, and Machine Vision[M]. Second Edition. United States of America: Thomson Learning and PT Press, 2003.

[6] 周　军,彭培欣. 自动磁粉探伤系统中的图像技术[J]. 仪器仪表学报,2003,24(4):461 − 462.