一种新颖的离散化算法及其应用

史志才 周金祖 夏永祥
(上海工程技术大学电子电气工程学院 上海 201620)

摘 要 连续数值属性的离散化是粒计算理论应用的重要步骤。首先对目前的离散化算法进行分类讨论,提出区间粒的概念,融合熵理论定义区间粒的粒度,进而提出基于粒计算的连续数值属性的离散化算法,并将该算法应用于入侵检测过程。实验结果表明该算法简洁高效,能够确保入侵检测系统的检测效果。

关键词 粒度计算 区间粒 离散化 熵

中图分类号 TP393.08 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2014.07.064

A NOVEL DISCRETISATION ALGORITHM AND ITS APPLICATION

Shi Zhicai Zhou Jinzu Xia Yongxiang
(*Electronic and Electrical Engineering Institute, Shanghai University of Engineering Science, Shanghai 201620, China*)

Abstract The discretisation of continuous numerical attributes is an important procedure for the application of granular computing. Some current discretisation algorithms are classified and discussed first. The concept of section granular is proposed. By fusing entropy theory the granularity of section granular is defined, thereby the discretisation algorithm based on granular computing is proposed as well. This algorithm is applied to intrusion detection process. Experimental results show that this algorithm is simple and effective, and can insure the accuracy of intrusion detection system.

Keywords Granular computing Section granular Discretisation Entropy

0 引 言

自 Zadeh 教授发表论文"Fuzzy sets and information granularity"以来,研究人员对信息粒和粒计算产生了浓厚兴趣[1]。Zadeh 教授认为信息粒广泛存在于自然界,只是在不同的领域其表现形式不同,信息粒化是人类对现实世界的一种抽象,是人类存储和处理信息的一种方式。粒计算对于解决实际问题具有非常重要的意义,特别是对于复杂的大型问题,通过粒划分可以转化为若干个简单的问题进行处理,从而降低问题的求解难度[2]。粒计算包括两个基本问题:粒化和基于粒化的计算,即如何构造粒化模型以及根据这个模型进行计算。连续数值属性的离散化是应用粗糙集等粒计算理论的重要步骤,下面我们将粒计算引入连续数值属性的离散化,融合熵理论来控制粒化过程,建立了基于粒计算的连续数值属性的离散化算法;以入侵检测领域的经典数据集:KDDcup99 为处理对象,就算法对于离散化对象即初始数据集的敏感性进行研究,同时应用于入侵检测系统并对其应用效果进行分析和评价。

1 离散化问题的粒度描述

在粒计算理论的应用过程中常采用一种特殊的知识表达系统:决策表对知识进行表达和处理。决策表可以表示为四元组 $S=(U,A,V,f)$,其中 U 是非空对象的有限集合,称为论域;A 是属性的非空集合,$A=C\cup D$,$C\cap D=\Phi$;其中 $C=\{a_1,a_2,\cdots,a_m\}$ 称为条件属性集;$D=\{d\}$ 具有唯一的决策属性(也称为类别属性),称之为决策属性集;$V=\bigcup_{a\in A}V_a$,V_a 是属性 a 的值域;$f:U\times A\to V$ 是一个信息函数,它为对象的每个属性赋以一个值,即:$\forall a\in A,\forall x\in U, f(x,a)\in V_a$。

假设对于 $\forall a\in C$,属性 a 的值域 $V_a=[l_a,r_a)\subset R$ 是实值区间,我们将该区间定义为一个区间粒;令 P_a 是对区间粒 V_a 的细分,即对于某一正整数 k_a,有 $P_a=\{[c_0^a,c_1^a),[c_1^a,c_2^a),\cdots,[c_{k_a-1}^a,c_{k_a}^a)\}$,$l_a=c_0^a<c_1^a<c_2^a<\cdots<c_{k_a-1}^a<c_{k_a}^a=r_a$,$V_a=[c_0^a,c_1^a)\cup[c_1^a,c_2^a)\cup\cdots\cup[c_{k_a-1}^a,c_{k_a}^a)$;这样将属性 a 的取值分成 k_a 个较细的区间粒,每个区间粒可以看作为一个等价类,其中的每个 $c_i^a(0\leqslant i\leqslant k_a)$ 称为属性 a 在其值域 V_a 上的一个断点。显然,离散化的目的就是按照一定的粒化准则对由条件属性所构成的粒空间进行细分,进而确定各个连续属性的最佳断点集;此时有:$f^p(x,a)=i\Leftrightarrow f(x,a)\in[c_{i-1},c_i)$,即细分后的每个区间粒对应一个离散值;这样经过离散化后,原来的决策表就被新的决策表所替代,而且要求这个新的决策表的决策能力保持不变。

收稿日期:2013-01-25。国家自然科学基金项目(61272097);上海工程技术大学学科专业建设项目(XKCZ1212);上海工程技术大学科技发展基金项目(2011XY16)。**史志才**,教授,主研领域:计算机网络与信息安全。**周金祖**,硕士生。**夏永祥**,讲师。

2　连续数值属性的离散化

连续数值属性的离散化在粗糙集等粒计算理论出现前就已经得到研究。粒计算理论出现后,广大研究者对已有的离散化算法进行引用或者改进以解决本领域中的相关问题。

离散化算法常分为无监督算法和有监督算法[3,4]。无监督算法不考虑属性与类别之间的关系,在离散化时往往需要人为地设定一些参数,而这些参数某种程度上带有一定的主观性,因而很难获得理想的离散化效果。有监督算法要考虑样本数据的类别属性,它比无监督算法更科学。有监督算法主要包括基于信息熵的离散化算法,如 Patterson-Niblett、IEM 等;基于统计的离散化算法,如 ChiMerge、Chi2 等;基于属性类别关联度的离散化算法,如 CADD、CAIM 等。

从粒计算角度看,离散化存在自顶向下的逐步细化和自底向上的逐步粗分两种实现方式。自顶向下方式首先将待离散化的样本数据集看成一个完整的区间粒,然后按照一定的粒化方法在区间中选择一个最佳断点将整个区间粒细分成两个子区间粒;以后在每个子区间粒上重复上述过程,直到满足一定的粒化准则;这种算法的典型代表是由 Fayyad 和 Irani 提出的决策树离散化算法[5]。该算法采用样本集的类别熵作为粒化准则,选择使得分割成两个子区间粒的类别熵最小的断点作为最佳断点,然后依此递归,直到算法满足由最小描述长度决定的终止条件时为止;这种算法每次确定一个断点,离散化过程所需时间较长,而且决定终止条件的最小描述长度常常难以确定。基于自底向上的离散化算法用样本集中所有不同的观测值构成初始区间粒分布,然后按照一定的粒化准则选择相邻的两个区间粒进行合并,得到新的区间粒分布,然后重复上述过程,直到得到的区间粒分布满足一定的终止条件,这种算法的典型代表是 ChiMerge 算法[6,7]。该算法基于数理统计中的 χ^2 测试作为粒化准则,在选择相邻区间粒进行合并时依次计算各相邻区间的 χ^2 值,并将 χ^2 值最小的相邻区间粒合并为一个粗的区间粒;这样依此循环,直到所有相邻区间粒的 χ^2 值都小于规定的阈值时为止。这种算法由于每次仅能归并相邻的两个区间粒,当样本数据集规模较大时算法的速度较慢。

显然,无论哪种算法在离散化过程中都应尽量保证系统的粒度不发生太大的变化,以保证无论是粒的细化还是变粗,系统的决策能力应保持不变。

3　基于粒计算的连续属性离散化算法

如前所述,对于自底向上实现方式,连续属性离散化的本质就是利用选取的若干个断点对条件属性所构成的粒空间进行划分,并根据一定粒化准则将区间粒进行聚类合并的一个粗分过程。首先根据条件属性 X 的值对样本空间 S 由小到大进行排序,设 X 中的不同观察值由小到大依次为 $x_1, x_2, \cdots, x_n$,构造 n 个初始区间粒 $I_1, I_2, \cdots, I_n$ 如下:

$$\left[x_1, \frac{x_1+x_2}{2}\right), \left[\frac{x_1+x_2}{2}, \frac{x_2+x_3}{2}\right), \cdots, \left[\frac{x_{n-2}+x_{n-1}}{2}, \frac{x_{n-1}+x_n}{2}\right), \left[\frac{x_{n-1}+x_n}{2}, x_n\right] \tag{1}$$

其中,每个初始区间粒只包括一个观察值。

然后对相邻的区间粒进行合并,直到满足一定的粒化准则;此时得到的每个区间粒即对应一个离散值,从而实现了连续属性的离散化。

为了提高区间粒合并的速度,对于初始化得到的区间粒 $I_1, I_2, \cdots, I_n$,若某些相邻的区间粒具有相同的分类属性则将它们合并为一个区间粒,从而减少初始区间粒的个数。

假设每次考虑将 m 个相邻区间粒 $I_{p+1}, I_{p+2}, \cdots, I_{p+m}$ 进行合并,可能存在多段这样的区间粒,那么优先选择哪一段区间粒进行合并呢? 下面通过信息熵来定义另一个概念:粒度损失,它描述了相邻区间粒合并前后的粒度变化,并将它作为粒化准则来选择优先合并的相邻区间粒。

现考虑将 m 个相邻区间粒 $I_{p+1}, I_{p+2}, \cdots, I_{p+m}$ 合并为一个区间粒 I,则合并前区间粒 I 的粒度定义 $Gr_{before}(I)$ 为:

$$Gr_{before}(I) = \sum_{i=1}^{m} \frac{|I_{p+i}|}{|I|} E(I_{p+i}) \tag{2}$$

其中,$|I_{p+i}|$ 和 $|I|$ 分别表示属性 X 的值位于区间粒 I_{p+i} 和 I 上的样本数,$E(I_{p+i})$ 为区间粒 I_{p+i} 的分类信息熵。而合并后区间粒 I 的粒度定义 $Gr_{after}(I)$ 为:

$$Gr_{after}(I) = -\sum_{i=1}^{k} p(D_i, I) \log p(D_i, I) \tag{3}$$

其中 $p(D_i, I)$ 是指区间粒 I 中类别属性值为 D_i 的样本的概率。

有了上述概念,定义区间粒合并前后的粒度变化为粒度损失 $Gr\_Loss(I)$,具体如下:

$$Gr\_Loss(I) = |Gr_{before}(I) - Gr_{after}(I)| \tag{4}$$

显然,区间粒的粒度描述的是系统的分辨能力,随着区间粒的合并粒将逐渐变粗,系统的分辨能力也将会变差,理想的情况是合并前后的粒度损失能够达到最小。下面以粒度损失作为粒化准则,选择粒度损失最小的那些区间粒进行合并,并以当前步的粒度损失大于前一步的 n 倍作为终止条件。具体的离散化算法如下:

Input:具有连续数值属性 X 和分类属性 D 的样本数据集 S,每次合并的区间粒个数 m,参数 n

Output:具有离散数值属性 X 和分类属性 D 的样本数据集 $\overline{S}$

Begin

s1:将集合 S 中的样本按照属性 X 的值由小到大排序;

s2:按照式(1)形成初始区间粒,然后将具有相同分类属性值的相邻区间粒合并为一个区间粒,得到初始区间粒序列 $I_1, I_2, \cdots, I_n$;

s3:依次选择 m 个相邻的区间粒,按照式(2)－式(4)计算这些相邻区间粒合并前后的粒度损失;

s4:选择具有最小粒度损失的相邻区间粒进行合并,得到新的区间粒序列;

s5:if 当前步的粒度损失不大于前一步的 n 倍 goto s3; else 执行 s6;

s6:将样本的 X 属性值离散化,令位于区间粒 $[a,b)$ 或者 $[a,b]$ 内样本的 X 属性值 = int((a + b)/2),从而得到新的样本集 $\overline{S}$;

s7:输出离散化的样本数据集 $\overline{S}$。

end

4　实验及其结果分析

实验采用主频 2.0GHz 双核处理器、2GB 内存的计算机,用

C++0 编程;以入侵检测数据集 KDDcup99 中 10% 的数据集为处理对象,该数据集包括近 50 万条记录,每条记录包括 41 个条件属性(包括 34 个数值属性和 7 个符号属性)和 1 个分类属性;分类属性又分成 5 大类:Normal、DOS、Probing、R2L 和 U2R,分别对应正常状态和 4 类网络攻击;对于 41 个条件属性,通过分析各数值属性取值的分布发现 34 个数值属性中有 13 个属性仅取有限的几个离散值,故将这 13 个属性和符号属性一样看作离散量,其他 21 个属性为连续数值属性。由于原始数据比较庞大,所以在研究过程中常常随机选出若干条记录作为决策分析的依据。下面从 21 个连续数值属性出发,探讨基于粒度损失的离散化算法对样本数据集的敏感性以及对入侵检测系统的影响。

为了分析离散化算法以及对入侵检测系统的影响程度,首先引入信息增益的概念[8]。设 S 是样本数据的集合,其类别属性 D 存在 k 个不同的取值,其第 i 个值记为 D_i,则样本集 S 的分类信息熵 $E(S)$ 为:

$$E(S) = -\sum_{i=1}^{k} p(D_i,S)\log p(D_i,S) \tag{5}$$

其中 $p(D_i,S)$ 是指样本集 S 中某个样本的类别属性值为 D_i 的概率。

假设 X 是样本空间上的一个条件属性,即 $X \in C$,X 将 S 分为不相交的 n 个子集,即 $S = \bigcup_{i=1}^{n} S_i$,当 $i \neq j, S_i \cap S_j = \Phi$,其中:

$$S_i = \{sf(s,X) = x_i, s \in S, x_i \in value(X)\}$$

式中,$value(X)$ 是属性 X 取值的集合,则 S 被属性 X 划分所得到的分类信息熵 $E(X,S)$ 可表示为:

$$E(X,S) = \sum_{i=1}^{n} \frac{|S_i|}{|S|} E(S_i) \tag{6}$$

条件属性 X 相对于样本集 S 的信息增益 $IG(X,S)$ 定义为:

$$IG(X,S) = E(S) - E(X,S) \tag{7}$$

根据式(5)-式(7),可以计算每个条件属性对于某个样本集的信息增益。依据信息论的观点,信息增益 $IG(X,S)$ 表示属性 X 及其属性值对决策系统分类能力的贡献程度。显然,对于一个连续的数值属性,当样本数据足够多时,其不同样本子集对决策系统分类能力的贡献程度应相同或者相近,即说明该属性在不同样本子集上经离散化后所得到的信息增益应一致或者相近。下面采用信息增益作为测度来评价基于粒度计算的离散化算法对于不同样本子集的敏感程度。

实验过程中随机选取 9 个数据集,每个数据集各包括 3 万条记录;选取数据时适当控制使得这些数据集的交集尽可能地小,以使方法具有普适意义。离散化时每次选择 3 个相邻区间(即 $m = 3$)进行合并以加快速度;分别对 21 个连续数值属性进行离散化,然后计算各个属性在不同样本集上的信息增益。为了使在不同样本集上计算的信息增益具有可比性,采用式(8)对每个样本集上得到的信息增益进行归一化,式中的 A 表示全部条件属性的集合。

$$\overline{IG(X,S)} = \frac{IG(X,S) - \underset{X\in A}{\mathrm{Min}} IG(X,S)}{\underset{X\in A}{\mathrm{Max}} IG(X,S) - \underset{X\in A}{\mathrm{Min}} IG(X,S)} \tag{8}$$

图 1 给出了信息增益变化较为明显的几个连续数值属性在 9 个不同样本集上进行离散化时的变化情况。从图中可以看出,离散化过程敏感于样本数据的有 6、24、30、35、36、37、38 号属性,其中 6、35、38 号属性的信息增益变化较为明显,其他属性的信息增益的变化较为平缓。上述实验结果说明本文所提出的离散化算法在部分条件属性上敏感于样本数据集,在不同的样本集上进行离散化时损失的信息量有一定差别,可能要影响系统的决策能力。

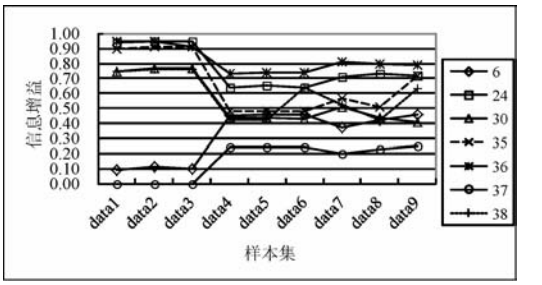

图 1 不同数据集上部分属性信息增益的变化情况

在上述离散化的基础上根据文献[9]提出的方法分两种情况将该算法应用于入侵检测。其一是采用 41 个属性直接建立 SVM 模型;其二是先进行属性约简,得到约简后的条件属性集合为{3,5,23,24,29,32,33,34},然后建立相应的 SVM 模型;进而分析离散化算法在不同情况下对检测模型的影响。实验时随机选择 1 万条记录作为训练数据,再选择另外完全不同的 1 万条记录作为测试数据,分别得到两种模型下 DOS 攻击的检测率为 99.26 和 98.57,probing 攻击的检测率为 99.81 和 99.19,U2R 和 R2L 攻击的检测率分别保持为 99.97 和 99.74 附近,几乎没有变化。显然,尽管两种模型下均含有离散化过程敏感于样本集的属性(约简后仅包括 24 号属性),但是对于各种类型攻击的检测率几乎保持不变,充分说明本文所提出的离散化算法尽管部分数值属性的离散化敏感于样本数据,但对入侵检测能力的影响很小,充分说明该离散化算法具有较强的鲁棒性。

5 结 语

连续数值属性的离散化是应用粗糙集等粒计算理论的一个重要环节,而寻求最优的离散化过程已被证明属于 NP 完全问题。本文对目前各种离散化算法进行了分类分析,采用粒计算对连续数值属性的离散化过程进行了描述,提出了基于粒计算的离散化算法,并就算法对于样本集的敏感性以及是否影响入侵检测效果等方面进行了分析。实验结果表明该离散化算法对于部分属性敏感于样本数据,但是无论使用原始的 41 个属性还是经过约简后的 8 个属性,尽管都含有敏感于样本集的部分条件属性,所建立的入侵检测模型均具有较高的分类能力,所以基于粒计算的离散化算法具有较强的鲁棒性,适合应用于入侵检测等决策系统。

参 考 文 献

[1] 王国胤,张清华,胡军. 粒计算研究综述[J]. 职能系统学报,2007,2(6):8-26.

[2] 张钹,张铃. 粒计算未来发展方向探讨[J]. 重庆邮电大学学报,2010,21(5):538-540.

[3] 刘业政,焦宁,姜元春. 连续属性离散化算法比较研究[J]. 计算机应用研究,2007,24(9):28-31.

[4] 花海洋,赵怀慈. 一种新的无监督连续属性离散化方法[J]. 计算机工程与应用,2011,47(6):208-210.

[5] Fayyad U, Irani K. Multi-interval discretization of continuous-valued attributes for classification learning[C]//Proceedings of the 13th International Joint Conference on Artificial Intelligence. San Mateo: Morgan Kaufmann Publisher, 1993:1022-1027.

(下转第 266 页)

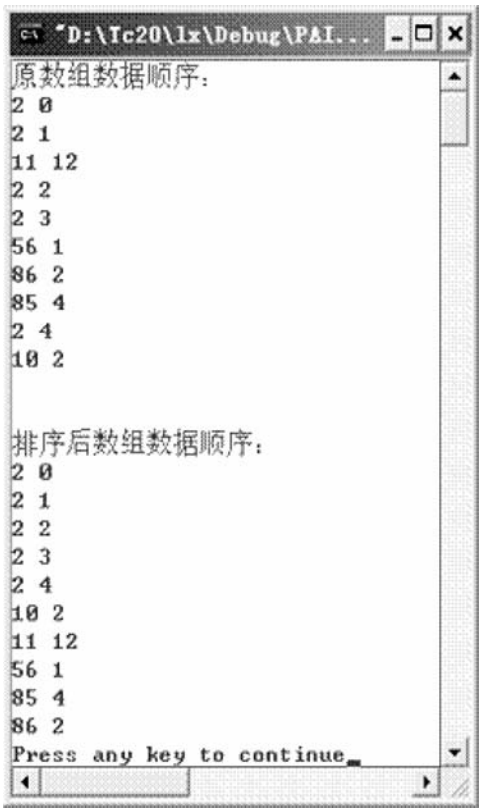

图 10 对十条记录排序后结果

从图中的结果可以看出,根据字段 1 排序后,字段 1 中相同的数据的先后顺序没有发生改变,字段 2 为 0 的排第一位,字段 2 为 1 的排序第二位,字段 2 为 4 的排第五位,因此算法是稳定的。

3.2 时间复杂度测试

在 Windows XP 的 VC 6.0 平台上,对大量数据进行多次测试(数据由随机函数 Rand()产生)。

1) 随机产生 50 000 个整型数据用各种排序方法排序所用时间基本如图 11 所示。由图 11 可知,稳定快速排序有很好的性能表现,速度与原快速排序基本上相同,与稳定排序的归并排序也基本相同。

图 11 对 50 000 个整型数据排序相对时间结果

2) 随机产生 50 000 个整型数据,然后用求余法得到 50 000 个 1 位整数、50 000 个 2 位整数、50 000 个 3 位整数、50 000 个 4 位整数和 50 000 个 5 位以上整数用稳定快速排序法进行排序所用时间基本如图 12 所示。由图 12 可知,当数据的位数少时,重复的数据多,需要移动的数据多,排序效率较慢,3 位以上数据的排序效率基本相同,达到了最好。

图 12 对不同位数排序相对时间结果

4 结 语

稳定快速排序算法是在原快速排序算法的基础上,对算法的稳定性进行的完善,理论和实践证明,使用稳定快速排序算法,可以确保排序是稳定的,且本算法的时间复杂度没有大的改变,只是多用了与排序数据数量相同的存储空间。

参 考 文 献

[1] 庞建雄. 排序算法稳定性的深入讨论[J]. 桂林电子工业学院学报,1996,16(1):1-4.
[2] 汪沁,奚李峰. 数据结构[M]. 北京: 清华大学出版社,2009.
[3] 秦锋. 数据结构[M]. 合肥: 中国科学技术大学出版社,2007.
[4] 王善坤,陶祯蓉. 一种三路划分快速排序的改进算法[J]. 计算机应用研究,2011,29(7):2513-2516.
[5] 汤亚玲,秦锋. 高效快速排序算法研究[J]. 计算机工程,2010,36(7):77-78.
[6] Cantone D, Cincotti G. Quic kHeapsort: An Efficient Mix of Classical Sorting Algorithms[J]. The Oretical Computer Science, 2002, 285: 25-42.
[7] 周建钦. 超快速排序[J]. 计算机工程与应用,2006,42(29):41-43.
[8] 郭晶旭. 基于快速排序的改进算法[J]. 计算机科学,2006,39(4A):343-344.

(上接第 254 页)

[6] 刘磊,闫德勤,桑雨. 连续属性离散化的 Bayesian-Chi2 算法[J]. 计算机工程与应用,2008,44(18):39-41.
[7] Kerber R C. Discretization of Numeric Attributes[C]//Proceedings of the 10th National Conference on Artificial Intelligence. MIT Press, 1992:123-128.
[8] 李刚,李霁伦. WILD:基于加权信息损耗的离散化算法[J]. 南京大学学报:自然科学版,2001,37(2):148-152.
[9] Xia Yongxiang, Shi Zhicai. An incremental SVM for intrusion detection based on key feature selection[C]//Proceedings of the Third International Symposium on Intelligent Information Technology Application, 2009:205-208.