基于粗糙集技术的压缩近邻规则

翟俊海 李胜杰 王熙照

(河北大学数学与计算机学院 河北省机器学习与计算智能重点实验室 保定 071002)

摘 要 近邻(Nearest Neighbor,NN)算法是一种简单实用的监督分类算法。但NN算法在分类未知类标的样例时,需要存储整个训练集,还要计算该样例到训练集中每一个样例之间的距离,所以NN算法的计算复杂度非常高。为了克服这一缺点,P. Hart提出了压缩近邻(Condensed Nearest Neighbor,CNN)规则算法,即从整个训练集中找原样例集的一致子集(一致子集是能正确分类训练集中其他样例的子集)。其计算复杂度依然比较高,特别是对于大型数据库,寻找其一致子集是非常耗费时间的。针对这一问题,提出了基于粗糙集技术的压缩近邻规则算法。该算法分为3步,首先利用粗糙集方法求属性约简(特征选择),以将冗余的属性去掉。然后选取靠近边界域的样例,以将冗余的样例去掉。最后从选出的样例中计算一致子集。该算法能同时沿垂直方向和水平方法进行数据约简。实验结果显示,所提出的方法是行之有效的。

关键词 近邻规则,一致集,样例选择,粗糙集,边界域

中图法分类号 TP181 **文献标识码** A

Condensed Nearest Neighbor Rules Based on Rough Set Technique

ZHAI Jun-hai LI Sheng-jie WANG Xi-zhao

(Key Lab. of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding 071002, China)

Abstract Nearest neighbor(NN) algorithm is a simple practical supervised classification algorithm. When it classifies an instance without class label, the whole training set must be stored in computer memory, and its distance to each one of the training set is computed. The NN algorithm suffers from a problem: very high requirements of memory space and response time. To overcome this drawback, P. Hart proposed the condensed nearest neighbor rules algorithm, with which a consistent subset(CS) is found from the whole training set(the CS is a subset that can classify all instances in training set correctly). The complexity of the CNN is also very high. Especially, it is extremely expensive to find a CS from a large database. In order to solve this problem, a CNN algorithm based on rough set technique was proposed in this paper, which consists of three stages. Firstly, to remove the superfluous attributes an attribute reduct is computed with rough set method. Secondly, the instances within boundary regions are selected. Meanwhile the redundant instances are removed. Finally, a CS is found from the selected instances. The proposed algorithm can simultaneously reduce data in horizontal and vertical directions. The experimental results show that the proposed method is effective and efficient.

Keywords Nearest neighbor rules, Consistent set, Instance selection, Rough sets, Boundary regions

1 引言

近邻(Nearest Neighbor,NN)分类算法[1]是一种简单、实用的监督分类算法,广泛应用于模式识别、数据挖掘、机器学习等领域。对于给定的训练集,用NN算法分类一个未知类标的样例需要在计算机中存储整个训练集,还要计算该样例到训练集中每一个样例之间的距离。所以,NN算法的计算复杂度非常高,特别是对于大型数据库,这已成为NN算法应用的主要瓶颈。针对这一问题,许多学者开展了深入的研究[2-5,7-14]。P. Hart首次提出了CNN算法[2],该算法也是最早被提出的样例选择算法。P. Hart称挑选出的样例子集为一致集(Consistent Set,CS),即用该子集中的样例分类训练集中的其他样例时,都能正确分类。CNN算法只能往候选一致集中添加样例,而不能从中删除样例。所以,用CNN算法求出的样例子集依然含有冗余的样例。另外,CNN算法对噪声和样例挑选的顺序都非常敏感,挑选出的样例子集未必是最小一致集(Minimal Consistent Set,MCS)。针对CNN算法的第一个问题,G. W. Gates提出了约简近邻(Reduced Nearest Neighbor,RNN)规则算法[3]。因为RNN是在CNN的基础上计算原样例集的一致集,所以当用CNN算法求出的

到稿日期:2011-03-10 返修日期:2011-05-21 本文受国家自然科学基金项目(60903088),河北省自然科学基金项目及河北省高校科技重点基金项目(F2010000323,ZD2010139)资助。

翟俊海(1964-),男,博士,副教授,CCF会员,主要研究方向为机器学习、模式识别、小波分析,E-mail:mczjh@hbu.cn;李胜杰(1985-),男,硕士生,主要研究方向为机器学习;王熙照(1963-),男,教授,博士生导师,主要研究方向为机器学习与计算智能。

样例子集包含 MCS 时，RNN 算法能求得 MCS，否则求出的样例子集不是 MCS。RNN 算法依然对噪声和样例挑选的顺序非常敏感。B. V. Dasarathy 提出了基于样例重要度的求 MCS 的算法[4]。H. Brighton 等提出了基于可达集和覆盖集的迭代样例滤波(Iterative Case Filtering，ICF)算法，对噪声和样例挑选顺序具有鲁棒性。D. L. Wilson 提出了编辑近邻(Editing Nearest Neighbor，ENN)规则算法[5]，ENN 算法能有效去除噪声。文献[4,6]很好地综述了样例选择算法。

在诸多的样例选择算法中，大都倾向于选取靠近分类边界的样例。这是因为这些样例的分类不确定性大，它们所提供的分类信息量也大，对分类起着重要的作用。基于这种思想，本文提出了基于粗糙集技术的压缩近邻规则算法。该算法分为 3 步，首先利用粗糙集方法求属性约简(特征选择)，这样能将冗余的属性去掉。然后选取靠近分类边界的样例，这样能将冗余的样例去掉。最后从选出的样例中计算一致子集。该算法能同时沿垂直方向和水平方法进行数据约简。实验结果显示，本文提出的方法是行之有效的。

2 基本概念

本节简单介绍要用到的一些基本概念，包括信息系统、决策信息系统、下近似、属性约简等[15,16]。

定义 1 一个信息系统是一个四元组 $IS=\langle U,A,V,f\rangle$。其中，$U$ 是有限对象的集合，称为论域；A 是用于描述对象的属性的集合；$V=\bigcup_{a\in A}V_a$，V_a 是属性 $a\in A$ 的取值范围；$f:U\times A\to V$ 称为信息函数，满足 $\forall a\in A,\forall x\in U,f(x,a)\in V_a$。若 $A=C\cup D,C\cap D=\phi$，C 为条件属性集，D 为单决策属性，则称 IS 为决策信息系统或简称决策表，记为 DT。

定义 2 给定信息系统 $IS=\langle U,A,V,f\rangle$，令 $P\subseteq A,x,y\in U$，称 x 和 y 关于 P 是不可分辨的，当且仅当 $\forall a\in P,f(x,a)=f(y,a)$。这样对于 $\forall P\subseteq A$，定义了一个 U 上的等价关系，记为 $IND(P)$，在不引起混淆时，也简记为 P。$\forall x\in U$，由等价关系 P 产生的 x 的等价类记为 $[x]_P$。这样，等价关系 P 对应论域 U 的一个划分，记为 $U/P=\{U_1,U_2,\cdots,U_k\}$。

定义 3 给定决策表 $DT=\langle U,C\cup D,V,f\rangle$，设 $P\subseteq C$，$X\subseteq U$，定义 X 关于 P 的下近似为

$$\underline{P}(X)=\{x|(x\in U)\wedge([x]_P\subseteq X)\}$$

定义 4 给定决策表 $DT=\langle U,C\cup D,V,f\rangle$，设 $P\subseteq C,U/D=\{U_1,U_2,\cdots,U_k\}$ 是由决策属性 D 导出的论域 U 的划分，定义决策属性 D 关于 P 的正域为

$$POS_P(D)=\bigcup_{i=1}^{k}\underline{P}(U_i)$$

定义 5 给定决策表 $DT=\langle U,C\cup D,V,f\rangle$，设 $P\subseteq C$，定义 D 相对于 P 的依赖度为

$$\gamma_P(D)=\frac{|POS_P(D)|}{|U|}$$

定义 6 给定决策表 $DT=\langle U,C\cup D,V,f\rangle$，设 $P\subseteq C$，$\forall a\in P$，若 $POS_P(D)=POS_{(P-\{a\})}(D)$ 成立，则称属性 a 为 P 中关于 D 是不必要的，否则称属性 a 为 P 中关于 D 是必要的。如果对于 $\forall a\in P$，属性 a 都是 P 中关于 D 是必要的，则称 P 相对于 D 是独立的。

定义 7 给定决策表 $DT=\langle U,C\cup D,V,f\rangle$，设 $P\subseteq C$，若 P 满足下面两个条件：

(1) P 相对于 D 是独立的；

(2) $POS_P(D)=POS_C(D)$

则称 P 为 C 相对于 D 的一个约简，记为 $P=RED_C(D)$。

本文利用如下的属性约简算法[17]计算决策表的属性约简。

算法 1 基于属性依赖度的约简算法

输入：决策表 $DT=\langle U,C\cup D,V,f\rangle$

输出：约简 $P=RED_C(D)$

算法步骤

Step1 初始化 $P=\phi$

Step2 do

Step3 对于 $\forall a\in C-P$

Step4 if($\gamma_{P\cup\{a\}}(D)>\gamma_P(D)$)

Step5 $P=P\cup\{a\}$

Step6 until $\gamma_P(D)==\gamma_C(D)$

Step7 Return P

CNN 算法是最早被提出的样例选择算法，该算法是从整个训练集中挑选一个一致子集代替原训练集来分类测试样例。因为 CNN 算法是从整个训练集中挑选一致子集，所以其计算复杂度依然比较高。另外，利用 CNN 算法挑选出的样例均位于分类边界附近。本文利用粗糙集方法计算边界域样例，然后从中挑选一致子集。这样，一方面可以去除冗余属性，另一方面不从整个训练集中挑选一致集，可以降低算法的计算复杂度。为了描述的方便，下面给出 CNN 算法。

算法 2 CNN 算法

输入：决策表 $DT=\langle U,C\cup D,V,f\rangle$

输出：样例子集 $CS\subseteq U$

算法步骤

Step1 从 U 中随机选取一个样例加入 CS。

Step2 利用 CS 作为训练集来分类 U-CS 中的样例。

Step3 如果 CS 能正确分类 U-CS 中的所有样例，则算法结束。

Step4 如果 CS 不能正确分类 $x\in U$-CS，则将 x 加入 CS，然后转 Step2。

3 基于粗糙集技术的压缩近邻规则

本文提出的基于粗糙集技术的压缩近邻规则算法分为 3 步，首先利用粗糙集方法求属性约简(特征选择)，这样能将冗余的属性去掉。然后选取靠近分类边界的样例，这样能将冗余的样例去掉。最后从选出的样例中计算一致子集。该算法能同时沿垂直方向和水平方法进行数据约简。算法描述如下。

算法 3 基于粗糙集技术的压缩近邻规则算法

输入：决策表 $DT=\langle U,C\cup D,V,f\rangle$

输出：样例子集 CS

算法步骤

Step1 利用算法 1 计算条件属性集 C 相对于决策属性 D 的约简 P

Step2 计算 $U/D=\{U_1,U_2,\cdots,U_k\}$。

Step3 对于任意的 $U_i(1\leqslant i\leqslant k)$，计算 $\underline{P}(U_i)$。

Step4 对于任意的 $\underline{P}(U_i)(1\leqslant i\leqslant k)$，计算其中心，记为 H_i。

Step5 对于任意的 $x\in U$，计算 x 到 $H_i(1\leqslant i\leqslant k)$ 的欧式距离，记为 $L_i(x)$。

Step6 选取满足条件 $L_i(x)\geqslant T\times\frac{1}{k}\sum_{i=1}^{k}L_i(x)$ 的样例 x，其中 $0.5\leqslant T\leqslant 1$。

Step7 计算 $BD(x)=\bigcup_{i=1}^{k}L_i(x)$。

Step8　利用算法 2 从 $BD(x)$ 中计算并输出样例子集 CS。

基于属性依赖度的属性约简算法的计算时间复杂度为 $O(m)$，空间复杂度为 $O(p)$。其中，$|C|=m$，$|P|=p$，P 为由算法 1 求出的属性约简。CNN 算法的计算时间复杂度为 $O(ns)$。其中，$|U|=n$，$|S|=s$，S 为由 CNN 算法选出的样本子集。空间复杂度为 $O(s)$。经简单计算可得，算法 3 的计算时间复杂度为 $O(n+n's')$，空间复杂度为 $O(s')$。其中，$|BD(x)|=n'$，$|S'|=s'$，S' 是由 CNN 算法从集合 $BD(x)$ 中求出的样例子集。算法 3 的空间复杂度明显低于 CNN 算法的空间复杂度。当 $n>\frac{n's'}{s-1}$ 时，$O(n+n's')\leqslant O(ns)$ 成立，即算法 3 的计算时间复杂度低于 CNN 算法的计算时间复杂度。而一般情况下 $n>\frac{n's'}{s-1}$ 是容易满足的，特别是对于较大规模的数据库。所以，算法 3 的计算时间复杂度和空间复杂度均优于 CNN 算法。

4　实验结果

表 1　实验结果（$T=1.0, 0.9, 0.8, 0.7$）

DB	Num1	Num2	Time1	Time2	T	R	Aver
Iris	4	3	0.0806	0.0708	1.0	0.50	83.52
					0.9	0.57	84.87
					0.8	0.71	90.85
					0.7	0.77	90.47
Ionosphere	34	18	0.4719	0.4273	1.0	0.73	88.60
					0.9	0.80	88.32
					0.8	0.87	87.75
					0.7	0.93	91.74
Breast	6	4	0.8022	0.7813	1.0	0.80	96.09
					0.9	0.86	95.65
					0.8	0.92	96.92
					0.7	0.96	97.58
Pima	8	8	2.7352	2.6612	1.0	0.92	78.91
					0.9	0.92	80.34
					0.8	0.96	81.90
					0.7	0.98	83.33
Parkinsons	22	11	0.1290	0.1237	1.0	0.59	71.79
					0.9	0.64	68.21
					0.8	0.70	69.74
					0.7	0.77	73.85
Heart	13	13	0.5465	0.5448	1.0	0.47	68.89
					0.9	0.55	68.52
					0.8	0.64	64.81
					0.7	0.73	75.19
Image Segmentation	19	9	0.1782	0.2266	1.0	0.63	70.48
					0.9	0.70	74.29
					0.8	0.76	76.19
					0.7	0.80	77.62
Glass	10	8	0.4545	0.4228	1.0	0.83	85.51
					0.9	0.84	84.11
					0.8	0.87	88.32
					0.7	0.90	92.52
Balance Scale	4	4	1.6611	1.6956	1.0	0.87	89.12
					0.9	0.90	91.68
					0.8	0.92	92.16
					0.7	0.95	92.48
Vowel	13	10	2.5603	3.4227	1.0	0.59	90.20
					0.9	0.70	91.82
					0.8	0.81	95.86
					0.7	0.92	98.28

为了进一步验证本文算法的有效性，我们在能力保持的前提下（即进行样例选取后的平均分类精度和未进行样例选取的平均分类精度保持在相同的水平上），采用十折交叉验证的方法，分别在 10 个 UCI 数据集[18]上进行实验。实验环境是 PC 机，双核 1.86G CPU，2G 内存，Windows XP 操作系统，Matlab 7.1 实验平台。10 个 UCI 数据集分别是 Iris Dataset，Ionosphere Dataset，Breast Dataset，Pima Dataset，Parkinsons Dataset，Heart Dataset，Image Segmentation Dataset，Glass Dataset，Balance Scale Dataset，Vowel Dataset。分别对参数 $T=1.0, 0.9, 0.8, 0.7$ 进行了实验，实验结果列在表 1 中。其中 *DB* 表示实验所用的数据库，*Num1* 表示原属性个数，*Num2* 表示约简包含的属性个数，*Time1* 表示 CNN 算法所用 CPU 时间，*Time2* 表示本文算法所用 CPU 时间（注：*Time2* 是参数 T 取 4 个不同值时算法所用的 CPU 时间的平均值），T 表示算法中的参数，R 表示选出的边界样例数占总样例数的比例，*Aver* 表示平均分类精度。从实验结果可以看出，选出的边界样例数越少，本文提出的算法越有效。边界样例的多少，取决于数据的分布。总体上，与 CNN 算法相比，本文提出的算法在时间复杂度（算法所用 CPU 时间）和空间复杂度（压缩比）上都有较大改进。实验结果证实了本文方法的有效性。

结束语　CNN 算法找出的一致集中的样例均位于分类边界附近，但是该算法是从整个训练集中寻找这些位于分类边界附近的样例，算法的计算复杂度依然比较高。针对这一问题，本文提出了基于粗糙集技术的压缩近邻规则算法，该算法分为 3 步，首先利用粗糙集方法求属性约简（特征选择），这样能将冗余的属性去掉。然后选取靠近边界域的样例，这样能将冗余的样例去掉。最后从选出的样例中计算一致子集。该算法能同时沿垂直方向和水平方法进行数据约简。实验结果显示，本文提出的方法是行之有效的。

参 考 文 献

[1] Cover T, Hart P. Nearest neighbor pattern classification [J]. IEEE Transactions on Information Theory, 1967, 13(1): 21-27

[2] Hart P. The condensed nearest neighbor rule [J]. IEEE Transaction on Information Theory, 1968, 14(5): 515-516

[3] Gates G W. The reduced nearest neighbor rule [J]. IEEE Transactions on Information Theory, 1972, 18(3): 431-433

[4] Brighton H, Mellish C. Advances in instance selection for instance-based learning algorithms [J]. Data Mining and Knowledge Discovery, 2002, 6(2): 153-172

[5] Wilson D L. Asymptotic properties of nearest neighbor rules using edited data [J]. IEEE Transactions on Systems, Man, and Cybernetics, 1972, SMC-2(3): 408-421

[6] Wilson D R, Martinez T R. Reduction techniques for instance-based learning algorithms [J]. Machine Learning, 2000, 38(3): 257-286

[7] Fayed H A, Atiya A F. A novel template reduction approach for the k-nearest neighbor method [J]. IEEE Transactions on Neural Networks, 2009, 20(5): 890-896

[8] Angiulli F. Condensed nearest neighbor data domain description [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(10): 1746-1758

[9] Angiulli F. Fast nearest neighbor condensation for large datasets classification [J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(11): 1450-1464

[10] Angiulli F, Astorino A. Scaling up support vector machines u-

sing nearest neighbor condensation [J]. IEEE Transactions on Neural Networks, 2010, 2(1): 351-357

[11] Ferrandiz S, Boulle M. Bayesian instance selection for the nearest neighbor rule [J]. Machine Learning, 2010, 81(3): 229-256

[12] Marchiori E. Class conditional nearest neighbor for large margin instance selection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(2): 364-370

[13] Nikolaidis K, Goulermas J Y, Wu Q H. A class boundary preserving algorithm for data condensation [J]. Pattern Recognition, 2011, 44(3): 704-715

[14] Barandela R, Ferri F J, Sanchez J S. Decision boundary preserving prototype selection for nearest neighbor classification [J]. International Journal of Pattern Recognition and Artificial Intelligence, 2005, 19(6): 787-806

[15] Pawlak Z. Rough sets [J]. International Journal of Information and Computer Sciences, 1982, 11: 341-356

[16] 苗夺谦，李道国. 粗糙集理论、算法与应用[M]. 北京：清华大学出版社，2008：34-66

[17] Parthalain N M, Shen Q. Exploring the boundary region of tolerance rough sets for feature selection [J]. Pattern Recognition, 2009, 42(5): 655-667

[18] Blake C L, Merz C J. UCI Repository of machine learning databases[EB/OL]. http://www.ics.uci.edu/~mlearn/MLRepository.html, 1996

(上接第 208 页)

人工生成少量的数据，设定支持度为 3，分别执行算法，其挖掘结果如表 3 所列。

由表可知，两个算法挖掘出的频繁序列是相同的，验证了 NGCWAP 算法的挖掘结果是准确的，因而算法是可行的。

4.2 同一数据集不同支持度下算法执行的时间变化和内存变化

采用生成器生成 5 万行的测试用例，分别设置不同的最小支持度，观察两个算法执行时间和内存占用情况，结果如图 4 和图 5 所示。由图 4 可知，对于同一数据集，随着最小支持度的增大，两个算法的执行时间都随之降低。这主要是因为当最小支持度增大时，满足条件的频繁序列减少，算法将花费较少的时间去挖掘频繁序列。Wap 和 NGCWap 算法抛弃了“候选-测试”的方式，直接在树上挖掘频繁序列，NGCWAP 算法虽然避免了条件树的构建，但在构建 NGCWAP_tree 过程中，增加了两次遍历，故时间跟 Wap 算法相差不大。

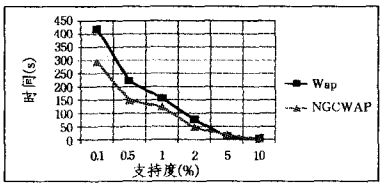

图 4 算法在不同支持度下的执行时间曲线图

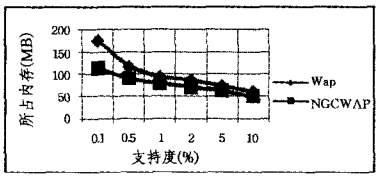

图 5 算法在不同支持度下的内存消耗曲线图

由图 5 可以看出，NGCWAP 算法在内存消耗上优于 Wap 算法，尤其当支持度比较小时更明显，这是因为 Wap 算法在挖掘的过程中需要保存大量的物理条件树，导致该算法占用了大量的内存。

4.3 同一支持度不同数据集下算法执行的内存变化

采用数据生成器分别生成 0.1 万，1 万，2 万，4 万，8 万，10 万行 6 组测试数据，最小支持度设为 3%，Wap 算法和 NGCWAP 算法的内存变化如图 6 所示。由图 6 可知，随着事务量的增大，两个算法的内存消耗呈线性增长。这主要是因为数据量增加，导致了树的规模增大，但 NGCWAP 算法的内存消耗总体小于 Wap 算法，尤其当数据量比较大时，更能体现 NGCWAP 算法的优势。

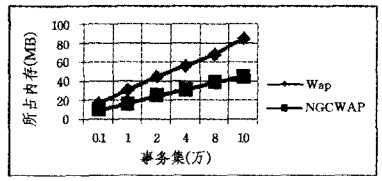

图 6 算法在不同数据集下的内存消耗曲线图

由本节实验可知，NGCWAP 算法相对于 Wap 算法，优势主要集中在内存消耗方面，其执行时间相差不是很大。总体来看，NGCWAP 算法的性能优于 Wap 算法。

结束语 在 Wap 算法的基础上提出了一种改进算法 NGCWAP，该算法改变了序列模式挖掘的方向，采用前序遍历号和后序遍历号优先发现频繁序列的前缀，能够有效避免条件树的构建。实验表明，NGCWAP 算法在内存消耗上优于 Wap 算法，尤其当数据量比较大时更明显。NGCWAP 算法虽然在挖掘过程中增加了两次遍历，但最后花费的时间与 Wap 算法相差不是很大。总体来看，改进后的算法在序列模式挖掘中是高效的。

参 考 文 献

[1] Agrawal R, Srikant R. Mining Sequential Patterns [C]// Proceedings of 11th International Conference on Data Engineering. 1995

[2] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases[C]// Proceedings of the 20th International Conference on Very Large Databases. Santiago, Chile, 1994: 487-499

[3] Pei Jian, Han Jia-wei. Mining Access Patterns Efficiently from Web Logs[C]//PADKK '00 Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2000

[4] 李朋轩. 基于 Web 挖掘的电子商务推荐技术的研究[D]. 哈尔滨：哈尔滨工程大学，2009

[5] Ezeife C I, Lu Yi. Mining Web Log Sequential Patterns with Position Coded Pre-Order Linked WAP-Tree [J]. Journal Data Mining and Knowledge Discovery, 2005, 10(1): 6-36

[6] Liu Li-zhi, Liu Jun. Mining web log sequential patterns with layer coded breadth-first linked WAP-tree[C]// Proceedings 2009 ISECS International Colloquium on Computing. Communication, Control, and Management. 2009