基于投票式属性重要度的快速属性约简算法

王 蓉[1] 刘遵仁[2] 纪 俊[2]
(青岛大学数据科学与软件工程学院 山东 青岛 266071)[1]
(青岛大学计算机科学技术学院 山东 青岛 266071)[2]

摘 要 作为经典Pawlak粗糙集的扩展,邻域粗糙集能有效处理数值型的数据。但是,因为引入了邻域粒化的概念,所以邻域实数空间下的计算量要比经典离散空间下的计算量大得多。对于邻域粗糙集算法而言,能够有效且快速地找到数据集的属性约简是十分有意义的。为此,针对现有算法中属性重要度定义的不足,首先提出了一种改进的投票式属性重要度,然后进一步提出了一种基于投票式属性重要度的快速属性约简算法。实验证明,与现有算法相比,在保证分类精度的前提下,该算法能更快速地得到属性约简。

关键词 域粗糙集,属性约简,投票,属性重要度
中图法分类号 TP182 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.07.034

Fast Attribute Reduction Algorithm Based on Importance of Voting Attribute

WANG Rong[1] LIU Zun-ren[2] JI Jun[2]
(School of Data Science and Software Engineering, Qingdao University, Qingdao, Shandong 266071, China)[1]
(College of Computer Science and Technology, Qingdao University, Qingdao, Shandong 266071, China)[2]

Abstract As an extension of the classical Pawlak rough set, neighborhood rough sets can efficiently manipulate numerical data. However, because the concept of neighborhood granulation is introduced, computational complexity in the neighborhood real space is much larger than that in the classical discrete space. For the neighborhood rough set algorithm, it is very meaningful to find the attribute reduction of the data set efficiently and quickly. To this end, an improved definition of voting attribute importance was proposed for the shortcomings of the definition of attribute importance in existing algorithms, then a fast attribute reduction algorithm based on importance of voting attribute was proposed. Compared with the existing algorithms, the experiment proves that the algorithm can get the attribute reduction more quickly under the premise of ensuring the classification accuracy.

Keywords Neighborhood rough set, Attribute reduction, Vote, Attribute significance

1 引言

粗糙集理论在特征提取、人工智能、数据挖掘等领域有着广泛的应用。由于定义在严格的等价关系上,经典Pawlak粗糙集[1]的处理对象局限于离散型数据。为了突破这种局限,Zadeh[2]提出了信息粒化和粒度计算的概念;Lin[3]在信息粒化和粒度的基础上提出了邻域模型的概念;Hu等[4]提出了基于邻域粒化和粗糙逼近的决策表属性约简算法。近年来,经各方研究后提出的邻域粗糙模型可以处理数值型数据,进一步扩展了粗糙集理论的应用范围[5-10]。

但是,与经典的Pawlak粗糙集不同,邻域粗糙集中的信息粒子需要通过度量计算来确定,这种方法在一定程度上增加了邻域粗糙集算法的计算量。为了降低时间开销,Hu等[4]于2008年提出了一种基于前向贪心思想的F2HARNRS(Fast Forward Heterogeneous Attribute Reduction Based on Neighborhood Rough Sets)算法。随后,Liu等[11]于2014年对该算法的正域计算进行了改进,提出了更快速的FHARA(Fast Hash Attribute Reduct Algorithm)算法。

分析F2HARNRS算法和FHARA算法可知,在每次对属性进行贪心选择的过程中,需要先对每个待选的属性都计算一遍,然后从中选取重要度最大的属性作为当次选取的属性。针对这个情况,本文提出一种与以上两种算法均不相同的基于投票式属性重要度的快速属性约简算法,其最大的特点是能一次性求得属性重要度大小的排列,避免了在每次贪心选择前都需要对各待选属性进行重要度的计算,从而进一步降低了算法的时间开销。

到稿日期:2017-05-18 返修日期:2017-08-27 本文受国家自然科学基金项目(61503208)资助。

王 蓉(1989-),女,硕士生,主要研究方向为粗糙集理论、数据挖掘,E-mail:475985222@qq.com;刘遵仁(1963-),男,博士,硕士生导师,主要研究方向为粗糙集理论、智能计算、数据挖掘等,E-mail:liuzunren@126.com(通信作者);纪 俊(1982-),男,博士,主要研究方向为数据挖掘、大数据应用、转化医学等,E-mail:1120108823@qq.com。

[7] BADINO H，HUBER D，KANADE T. The CMU Visual Localization Data Set[OL]. http://3dvis.ri.cmu.edu/data-sets/localization.

[8] DUAN W W. Detection and abnormal recognition of contact network equipment for surveillance video [D]. Chengdu：Southwest Jiaotong University，2016. (in Chinese)
段旺旺. 面向监控视频的接触网设备检测及异常识别[D]. 成都：西南交通大学，2016.

[9] HE Z M. Fast detection of vanishing points of the perspective text image [J]. Shanghai University of Engineering and Technology，2012，23(2)：229-232. (in Chinese)
贺志明. 透视文本图像的灭点快速探测[J]. 上海工程技术大学报，2012，23(3)：229-232.

[10] XIE W H，ZHANG Z X，ZHANG J Q. A new camera calibration method based on vanishing points[J]. Journal of Harbin Institute of Technology，2003，35(11)：1384-1387. (in Chinese)
谢文寒，张祖勋，张剑清. 一种新的基于灭点的相机标定方法[J]. 哈尔滨工业大学学报，2003，35(11)：1384-1387.

[11] HARRIS C，STEPHENS M. A combined corner and edge detector[C]// Alvey Vision Conference. Manchester，UK，1988：147-151.

[12] LUCAS B D，KANADE T. An iterative image registration technique with an application to stereo vision[C]// Proceedings of the 7th International Joint Conference on Artificial Intelligence. 1981：674-679.

[13] JIANG Z J，YI H R. A feature tracking method based on image pyramid optical flow[J]. Journal of Wuhan University(Information Science Edition)，2007，32(8)：680-683. (in Chinese)
奖志军，易华蓉. 一种基于图像金字塔光流的特征跟踪方法[J]. 武汉大学学报(信息科学版)，2007，32(8)：680-683.

[14] HE M Y，DAI Y C. Three-Dimensional Measurement Algorithm for Uncalibrated Multi-view Images [J]. Microelectronics & Computer，2010，27(9)：181-185. (in Chinese)
何明一，戴玉超. 多视角未标定图像三维测量算法[J]. 微电子学与计算机，2010，27(9)：181-185.

[15] LIANGJ J，QIN A K，SUGANTHAN P N，et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactionson Evolutionary Computation，2006，10(3)：281-295.

[16] HU Z Y，WU F C. Calibration Method Based on Active Vision Camera[J]. Chinese Journal of Computers，2002，25(11)：1149-1156. (in Chinese)
胡占义，吴福朝. 基于主动视觉摄像机标定方法[J]. 计算机学报，2002，25(11)：1149-1156.

(上接第 201 页)

[2] ZADEH L A. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic[J]. Fuzzy Sets & Systems，1997，90(90)：111-127.

[3] LIN T Y. Granular Computing on binary relations I：Data - mining and neighborhood systems[J]. Rough Sets in Knowledge Discovery，1998(2)：165-166.

[4] HU Q，YU D，LIU J，et al. Neighborhood rough set based heterogeneous feature subset selection[J]. Information Sciences，2008，178(18)：3577-3594.

[5] CHEN H，YANG J A，ZHUANG Z Q. The Core of Attributes and Minimal Attributes Reduction in Variable Precision Rough Set[J]. Chinese Journal of Computers，2012，35(5)：1011-1017. (in Chinese)
陈昊，杨俊安，庄镇泉. 变精度粗糙集的属性核和最小属性约简算法[J]. 计算机学报，2012，35(5)：1011-1017.

[6] LOU C，LIU Z R，GUO G Z. Quick Attribute Reduct Algorithm on Neighborhood Rough Set Based on Block Set[J]. Computer Science，2014，41(S2)：337-339. (in Chinese)
娄畅，刘遵仁，郭功振. 基于块集的邻域粗糙集的快速约简算法[J]. 计算机科学，2014，41(S2)：337-339.

[7] XU J C，XU T H，SUN L，et al. Feature Selection for Cancer Classification Based on Neighborhood Rough Set and Particle Swarm Optimization[J]. Journal of Chinese Computer Systems，2014，35(11)：2528-2532. (in Chinese)
徐久成，徐天贺，孙林，等. 基于邻域粗糙集和粒子群优化的肿瘤分类特征基因选取[J]. 小型微型计算机系统，2014，35(11)：2528-2532.

[8] MENG Z Q，SHI Z Z. On quick attribute reduction in decision-theoretic rough set models[J]. Information Sciences，2016，330(C)：226-244.

[9] LIU F，LI T R. Accelerated Attribute Reduction Algorithm Based on Probabilistic Rough Sets[J]. Computer Science，2016，43(12)：63-70. (in Chinese)
刘芳，李天瑞. 一种基于概率粗糙集的属性约简加速算法[J]. 计算机科学，2016，43(12)：63-70.

[10] YAN H C，ZHANG F，LIU B X. Rough decision rules extraction and reduction based on granular computing[J]. Journal on Communications，2016，37(Z1)：30-35. (in Chinese)
阎红灿，张奉，刘保相. 基于粒计算的粗决策规则抽取与约简[J]. 通信学报，2016，37(Z1)：30-35.

[11] LIU Y，HUANG W，JIANG Y，et al. Quick attribute reduct algorithm for neighborhood rough set model[J]. Information Sciences，2014，271(7)：65-81.

[12] LIU Z R，WU G F. An Algorithm for Sub-optimal Attribute Reduction in Decision Table Based on Neighborhood Rough Set Model[J]. Computer Science，2012，39(10)：268-271. (in Chinese)
刘遵仁，吴耿峰. 基于邻域粗糙集模型的高维数据集快速约简算法[J]. 计算机科学，2012，39(10)：268-271.

[13] HU Q H，YU D R，XIE Z X. Numerical Attribute Reduction Based on Neighborhood Granulation and Rough Approximation [J]. Journal of Software，2008，19(3)：640-649. (in Chinese)
胡清华，于达仁，谢宗霞. 基于邻域粒化和粗糙逼近的数值属性约简[J]. 软件学报，2008，19(3)：640-649.

2 相关概念与原理

2.1 邻域粗糙集

定义 1 给定 n 维实数空间 R^n，对于空间中的任意两个点 $x_i=(x_{i1},x_{i2},\cdots,x_{in})$ 和 $x_j=(x_{j1},x_{j2},\cdots,x_{jn})$，定义 $d(x_i,x_j)$ 是 R^n 上的一个度量计算，其满足：

$$d(x_i,x_j)=(\sum_{p=1}^{n}|x_{ip}-x_{jp}|^2)^{\frac{1}{2}}$$

定义 2 在实数空间上，定义样本的非空有限集合 $U=\{x_1,x_2,\cdots,x_n\}$，且称 U 为论域。定义 U 上的任意样本 x_i 的 δ-邻域为 $\delta(x_i)=\{x_j\mid x_j\in U,d(x_i,x_j)\leqslant\delta\}$，其中 $\delta\geqslant 0$。$\delta(x_i)$ 称作由 x_i 生成的 δ-邻域信息粒子，简称为 x_i 的邻域粒子。

2.2 邻域决策系统

定义 3 四元组 $NDT=(U,C\cup D,V,f)$ 为一个决策系统，其中 U 是论域；C 是条件属性，D 是决策属性，且 $C\cap D=\varnothing$，$C\neq\varnothing$，$D\neq\varnothing$；V 是信息函数 f 的值域。

定义 4 对于一个给定的决策系统 $NDT=(U,C\cup D,V,f)$，D 将 U 划分为 N 个等价类：$D_1,D_2,\cdots,D_N$，$\forall B\in C$，定义决策属性 D 关于 B 的下近似和上近似为 $\underline{N_BD}=\bigcup_{i=1}^{N}\underline{N_BD_i}$ 和 $\overline{N_BD}=\bigcup_{i=1}^{N}\overline{N_BD_i}$。其中，$\underline{N_BD_i}=\{x_i\mid\delta_B(x_i)\subseteq D_i,x_i\in U\}$，$\overline{N_BD_i}=\{x_i\mid\delta_B(x_i)\cap D_i\neq\varnothing,x_i\in U\}$。根据定义 1，$\delta_B(x_i)=\{x\mid d(B(x_i),B(x))\leqslant\delta,x\in U\}$。

定义决策属性集 D 关于 B 的边界域为 $BN(D)=\overline{N_BD}-\underline{N_BD}$，正域为 $Pos_B(D)=\underline{N_BD}$，以此来刻画一个邻域决策系统。

2.3 属性约简

定义 5[1] 给定有限集合 $B\subseteq C$，若满足 $Pos_B(D)=Pos_C(D)$，则称 B 是一个独立属性子集；如果对于 $\forall a\in B$，$Pos_{B-\{a\}}(D)\subset Pos_B(D)$，则称 B 为 C 的一个属性约简。

对于一个数据集而言，设计一种有效的既能删除冗余属性又能保持原数据集的分类能力从而得到属性约简的算法，是一个 NP-Hard 问题。

3 基于投票式属性重要度的算法

3.1 F2HARNRS 算法和 FHARA 算法

贪心策略具有能在较短的时间内求解最优解或次优解的特点。Hu 等[4]首先根据属性能使正域中样本个数增加的数量定义了属性重要度的概念，然后根据属性重要度的大小构造了前向贪心的 F2HARNRS 算法。其具体策略是：初始化属性约简集合为空集，当前正域为空集，每次选取使当前正域中样本个数增加最多的属性加入集合，直至对于当前集合而言所有属性的重要度全为 0 或样本全被划入当前正域中时，输出集合。其中，根据决策属性的不同，可将样本分为 D_1，$D_2,\cdots,D_n$，若 $k\neq l$，则有 $D_k\cap D_l=\varnothing$。不失一般性，分析任一样本 $x_i\in U$，如果 $\delta(x_i)\subseteq D_k$，则 $\delta(x_i)\cap D_l=\varnothing(l\neq k)$，从而 $x_i\notin\underline{N_BD_l}$，$x_i\notin\overline{N_BD_l}$。由此可知，对于任一样本而言，如果其属于某一决策的正域，那么它一定不会属于任一决策的边界域，并且新增加的属性不会将已属于正域的样本变为非正域样本[13]，因此在算法的计算过程中，每次仅对还未被判定为正域的样本进行正域计算。

随后，通过分析 F2HARNRS 算法中正域计算的不足，Liu 提出了更快速的 FHARA 算法。

F2HARNRS 算法和 FHARA 算法对属性重要度的定义如下所示。

定义 6[4] 给定邻域决策系统 $NDT=(U,C\cup D,V,f)$，$\forall B\in C$，$\forall a\in C-B$，定义属性 a 相对于集合 B 的属性重要度为：

$$SIG(a,B,D)=|Pos_{B\cup a}(D)|-|Pos_B(D)|$$

由上式可知：对于 $\forall B\in C$，F2HARNRS 算法和 FHARA 算法会选取 $a(\max SIG(a,B,D))$ 加入 B 中，而对于 $\max SIG(a,B,D)$，则需要 $|C-B|$ 次正域计算，即在贪心选择之前，各待选属性都需要被计算一次。

分析 F2HARNRS 算法的正域计算：图 1 中，样本 x_i 需与论域中的所有样本进行计算，时间复杂度为 $O(m|U|^2)$。

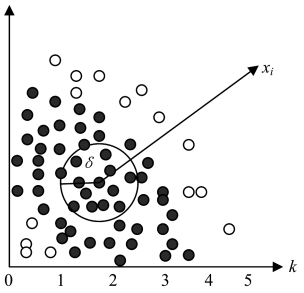


图 1 F2HARNRS 算法的正域计算

Fig. 1 Positive region calculation of F2HARNRS algorithm

FHARA 算法的正域计算过程如下：首先，采用映射划分，即根据各样本与标准样本间的距离大小为各样本划分等级；然后，基于等级将各样本映射到不同的有限集合 B_0，$B_1,\cdots,B_k$ 中，其中 $B_k=\{x_i\mid x_i\in U,k=\lceil d(x_i,x_0)/\delta\rceil\}$，$x_0$ 是标准样本，定义 $x_0=\{\forall a\in C,a(x_0)=\min\lceil a(x_i)\rceil\}$，$x_i\in U$。基于图 1 中的样本分布，基于 FHARA 算法的正域计算如图 2 所示，x_i 只需要与自身所在扇环及相邻扇环中的样本进行计算即可，其正域计算的时间复杂度为 $O(q\cdot n|U|^2)$，其中 $q=4/\lceil\max d(x_i,x_0)/\delta\rceil$。

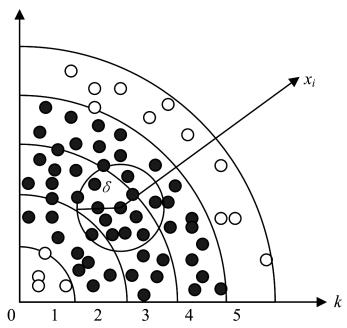


图 2 FHARA 算法的正域计算

Fig. 2 Positive region calculation of FHARA algorithm

根据以上分析，假设某一数据集有 m 个属性，约简结果

中包含 k 个属性，且每增加一个属性，正域中就增加 $\frac{|U|}{k}$ 个样本，则 F2HARNRS 算法与 FHARA 算法的计算量可表示为：

$$m|U|\cdot pos_1+(m-1)\frac{k-1}{k}|U|\cdot pos_2+\cdots+$$

$$(m-k)\frac{1}{k}|U|\cdot pos_k$$

其中，pos_i 表示第 i 次属性选择时对某个样本进行正域判定所需要的时间开销。

3.2　基于投票式的属性重要度

根据以上分析，算法的时间开销与正域计算次数以及每次正域计算的时间有关。相较于 F2HARNRS 算法，FHARA 算法通过映射划分的策略减少了样本间计算的次数，从而减少了正域计算的时间开销，达到了降低算法时间开销的目的；而本文提出的基于投票式属性重要度的快速属性约简算法则给出了一种新的属性重要度的概念。之前有学者提出了正域重要度的概念，指出主要是通过计算正域的个数来判断属性的重要程度；还有学者提出了信息熵重要度的概念，通过对概率的计算判断属性的重要程度。这些已有方法均需要对属性重要度的排序进行 n 轮计算，而本文提出的属性重要度只需要通过一轮计算即可获得整个重要度的排序，这是本文提出的属性重要度求解与方法的特点所在 。

本文提出一种针对全局的投票式属性重要度。对于一个具有 m 个 n 维样本的数据集，定义属性重要度的过程可以通俗地理解为：对于 m 个样本，从第一个样本开始，依次给 n 个属性投票打分，直至最后一个样本投票结束，此时统计 n 个属性各自的总得分，并根据总得分对所有属性从大到小排序，得分越高说明属性越重要。具体的策略如下：首先，对于样本 x_i，通过计算欧氏距离选出距离该样本最近的同类样本和异类样本；然后，对比该样本与同类样本各属性取值的差值，将属性按照差值从大到小排序，该样本对属性进行第一次投票计分，投票数按照属性的序列从 1 开始依次递增 1；接着，对比该样本与异类样本各属性取值的差值，将属性按照差值从小到大排序，此时该样本对属性进行第二次投票，投票数按照属性的序列从 1 开始依次递增 1；最后，将数据集中每个样本对属性集两次投票的总数按照从大到小排序，以票数的多少作为属性重要度的衡量标准，即票数越多的属性重要度越高，票数越少的属性重要度越低。

在定义投票式属性重要度的概念之前，首先给出以下定义。

定义 7　任取样本 x_i，其中决策属性值记为 D_{x_i}，根据定义 1 可得 $d(x_i,x_j)(x_j\in U,x_j\neq x_i)$。记距离样本 x_i 最近的同类样本点和异类样本点分别为：

$$SMIN_{x_i}=\{x_j\,|\,\min[d(x_i,x_j)],D_{x_i}=D_{x_j}\}$$

$$HMIN_{x_i}=\{x_j\,|\,\min[d(x_i,x_j)],D_{x_i}\neq D_{x_i}\}$$

定义 8　对于一个给定的邻域决策系统 $NDT=(U,C\cup D,V,f)$，其中属性集 C 可表示为 $C=\{C_1,C_2,\cdots,C_m\}$，由定义 7 可得样本 x_i 的 $SMIN_{x_i}$ 和 $HMIN_{x_i}$，计算

$$SADV_{x_i}=\{|C_{x_i1}-C_{SMIN_{x_i}1}|,|C_{x_i2}-C_{SMIN_{x_i}2}|,\cdots,$$
$$|C_{x_im}-C_{SMIN_{x_i}m}|\}$$

$$HADV_{x_i}=\{|C_{x_i1}-C_{HMIN_{x_i}1}|,|C_{x_i2}-C_{HMIN_{x_i}2}|,\cdots,$$
$$|C_{x_im}-C_{HMIN_{x_i}m}|\}$$

将 $SADV_{x_i}$ 中每项所代表的属性按照绝对值从大到小排序，得到同类样本属性序列，记为 $SADVO_{x_i}$；将 $HADV_{x_i}$ 中每项所代表的属性按照绝对值从小到大排序，得到异类样本属性序列，记为 $HADVO_{x_i}$。

由定义 7 和定义 8 可得到同类样本属性序列 $SADVO_{x_i}$ 和异类样本属性序列 $HADVO_{x_i}$。下面给出投票式属性重要度的定义。

定义 9　$NDT=(U,C\cup D,V,f)$ 表示一个邻域决策系统，其中 $C=\{C_1,C_2,\cdots,C_m\}$，任取样本 $x_i\in U$，使 x_i 根据 $SADVO_{x_i}$ 和 $HADVO_{x_i}$ 得到的属性顺序对每个属性进行两轮投票，并将每个属性所得票数记为 $SVOTE_{x_i}(C_m)$ 和 $HVOTE_{x_i}(C_m)$。统计属性 C_m 所得的所有票数即为投票式属性重要度，记为：

$$VSig(C_m)=\sum_{i=1}^{U}[SVOTE_{x_i}(C_m)+HVOTE_{x_i}(C_m)]$$

将所得到的 $VSig(C_m)$ 从大到小排序，即可得到投票式属性重要度序列 VSigOrder。下面给出 VSigOrder 算法的具体步骤。

算法 1　VSigOrder 算法

```
Input:NDT=(U,C∪D,V,f)
Output:投票式属性重要度序列 VSigOrder
Step1  初始化 dist[U×U],VSigOrder=∅
Step2  计算各样本之间的距离
       for i=1:n-1
         for j=i+1:n
           dist(i,j)=d(x_i,x_j);
           dist(j,i)=dist(i,j);
         end
       end
Step3  SMIN_xi=min[d(x_i,x_j)] & D_xi=D_xj;
       HMIN_xi=min[d(x_i,x_j)] & D_xi≠D_xj;
Step4  for i=1:n
         SADV_xi=abs(x_i-SMIN_xi);
         HADV_xi=abs(x_i-HMIN_xi);
       end
Step5  for i=1:n
         SADVO_xi=sort(SADV_xi,"descend");
         HADVO_xi=sort(HADV_xi);
       end
Step6  vote=1;
       for i=1:n
         for cNum=SADVO_xi
           Vote(cNum)=Vote(cNum)+vote;
           vote=vote+1;
         end
         for cNum=HADVO_xi
             Vote(cNum)=Vote(cNum)+vote;
```

```
            vote=vote+1;
          end
        end
Step7   VSigOrder=sort(Vote(C));
Step8   return VSigOrder
```

3.3 FRABVAI 算法

根据 3.2 节中定义 9 给出的投票式属性重要度的定义，给出 FRABVAI 算法的具体步骤。

算法 2 FRABVAI 算法

```
Input:NDT=(U,C∪D,V,f),VSigOrder,δ
Output:属性约简 red
Step1   初始化 red=Ø,pos=Ø,待检验样本
        smp_chk=U,num=0
Step2   while sum(smp_chk)≠0
          num=num+1;
          pos=Pos(smp_chk,[red,VsigOrder(num)]);
          if pos≠Ø
            red=[red,VsigOrder(num)];
            smp_chk=setdiff(smp_chk,pos);
          else
            break;
          end
        end
Step3   return red
```

假设某一数据集有 m 个属性，约简结果中包含 k 个属性，且每增加一个属性，正域中增加$\frac{|U|}{k}$个样本，则 FRABVAI 算法的计算量为：

$$m\frac{|U|(|U|-1)}{2}+|U|+1\cdot|U|\cdot pos_1+1\cdot\frac{k-1}{k}|U|\cdot pos_2+\cdots+1\cdot\frac{1}{k}|U|\cdot pos_k$$

其中，$m\frac{|U|(|U|-1)}{2}$是计算样本间的欧氏距离时的计算量，$|U|$是投票计算量。

通过以上对计算量的分析可知，FRABVAI 算法首先在贪心选择之前按照重要度完成对属性的排序，在贪心时只需按照顺序直接进行正域的计算，每一次正域的计算都是增加一维的计算；而 F2HARNRS 算法和 FHARA 算法需要在贪心过程中对属性重要度和正域进行计算，对于正域，每一次均是进行 n 维计算。因此，FRABVAI 算法的计算量少于 F2HARNRS 算法和 FHARA 算法的计算量。

4 实验分析

因为 FHARA 算法基于 F2HARNRS 算法在计算量上进行了改进，所以本文只对 FHARA 算法和 FRABVAI 算法进行时间和精度比较，以验证 FRABVAI 算法的有效性。

4.1 实验环境

UCI(University of California Irvine)提供了一系列用于分类测试的标准数据集。本文从 UCI 中选取了 7 个具有代表性的数据集，其描述如表 1 所列。

表 1 数据集描述

Table 1 Description of data set

编号	数据集	样本数	属性数	类别数
1	Iris	150	4	3
2	Wine	178	13	3
3	Sonar	208	60	2
4	Ionosphere	351	34	2
5	WDBC	569	30	2
6	Biodeg	1055	41	2
7	Segmentation	2310	19	7

本次实验在一台 CPU 为 G645 和内存为 4GB 的 PC 机上进行，并采用 Windows 7 环境下的 MATLAB R2014a 进行算法仿真。

4.2 δ 的取值

在邻域粗糙集中，δ 的取值十分重要，它的设定直接影响到约简的结果。若取值过大，则会使得大部分样本被划分到同一邻域，所得约简个数偏少；若取值过小，则会使得约简个数太多，约简效果不理想。

在传统的邻域粗糙集中，δ 的取值一般是根据经验来设定，这种做法对不同的数据集有不同的效果。因此，本文采用文献[13]中提出的利用标准差来度量 δ 取值的方法，即假设条件属性 $C=\{C_1,C_2,\cdots,C_m\}$，δ 的取值公式如下：

$$\delta=\sigma(\sigma(C_1),\sigma(C_2),\cdots,\sigma(C_m))$$

4.3 实验结果

为了消除量纲对数据的影响，首先对数据集中的数据进行归一化处理。实验过程中随机选取 2/3 的样本作为训练样本，其余 1/3 的样本作为测试样本。将 FHARA 算法和 FRABVAI 算法各执行 20 次，最后的分类精度取均值，并在 SVM，KNN 和 CART 这 3 个分类器下测试属性约简后的分类精度。

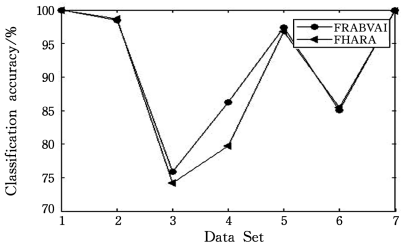
两种算法在 3 种分类器下所得运行时间和分类精度如表 2 所列。

表 2 算法运行时间和分类器下的分类精度

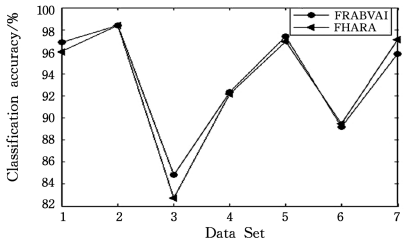
Table 2 Running time and classification accuracy of classifiers

编号	数据集	FRABVAI				FHARA			
		Time/s	SVM	KNN	CART	Time/s	SVM	KNN	CART
1	Iris	0.195	100	96.88	94.33	0.36	100	96	93.56
2	Wine	0.311	98.5	98.39	97.95	1.7	98.62	98.39	96.98
3	Sonar	0.36	75.92	84.84	64.38	8.04	74.23	82.76	63.62
4	Ionosphere	0.83	86.35	92.34	84.96	8.99	79.7	92.2	85.74
5	WDBC	1.57	97.34	97.42	92.55	12.43	96.84	96.94	92.55
6	Biodeg	7.49	85.09	89.2	76.25	146.37	85.49	89.45	78.37
7	Segmentation	33.03	99.92	95.83	93.48	760	99.82	97.07	94.14

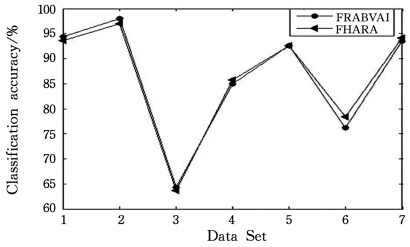
表 2 中，3 种分类器下 FRABVAI 算法和与 FHARA 算法的分类精度的对比折线图如图 3 所示。



(a)SVM 分类器下两算法分类精度的对比



(b)KNN 分类器下 F 两算法分类精度的对比



(c)CART 分类器下两算法分类精度的对比

图 3　3 种分类器下 FRABVAI 与 FHARA 的分类精度对比

Fig. 3　Comparison of classification accuracy for FRABVAI and FHARA under three classifiers

分析图 3 可以看出，两种算法在分类精度上的走势大致相同。在误差允许的范围内，针对数据量较小的数据集，FRABVAI 算法的分类精度略高于 FHARA 算法的分类精度；但随着数据量的增多，FRABVAI 算法的分类精度逐渐低于 FHARA 算法的分类精度。这是因为随着样本数量的增加，通过距离样本点最近的样本来进行属性重要度的判断的条件太单一，会受到数据集中噪声数据的干扰，从而影响对属性重要度的判断。虽然噪声数据对属性重要度的判断会产生影响，但可以看出 FRABVAI 算法的分类精度依然能够保持与 FHARA 算法同等的效果。

表 2 中，两种算法的运行时间的对比折线图如图 4 所示。

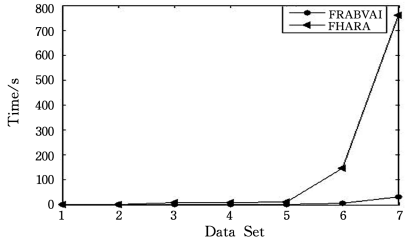


图 4　FRABVAI 与 FHARA 运行时间的对比

Fig. 4　Comparison of running time of FRABVAI and FHARA

从图 4 可以看出，FRABVAI 算法的折线一直处于 FHARA 算法的折线的下方，说明 FRABVAI 算法的运行时间较 FHARA 算法更短，具有更少的时间开销。这一点也验证了 3.3 节中对算法计算量所进行的性能分析。

针对各个数据集，用在 FRABVAI 算法下得到的运行时间与在 FHARA 算法下得到的运行时间的比值来表示 FRABVAI 算法相对于 FHARA 算法的效率，比值越小，说明 FRABVAI 算法的效率越高。

表 3　FRABVAI 算法的效率

Table 3　Efficiency of FRABVAI algorithm

（单位：%）

编号	数据集	比值
1	Iris	54.17
2	Wine	18.29
3	Sonar	4.48
4	Ionosphere	9.23
5	WDBC	12.63
6	Biodeg	4.98
7	Segmentation	4.34

表 3 的折线图如图 5 所示。

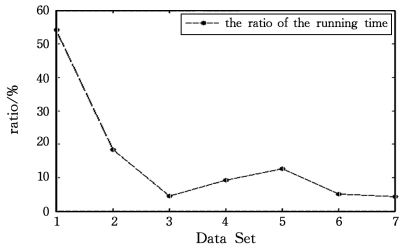


图 5　FRABVAI 算法的效率折线图

Fig. 5　Line chart of efficiency of FRABVAI algorithm

分析图 5 可以看出，FRABVAI 算法的效率波动较大，在 4%～55%的区间上均有取值，取值跨度比较大。其中，大部分点的取值较低，在 4%～20%之间，这说明对于大部分数据集而言，FRABVAI 算法的效率较高。造成这种波动性的原因是，FRABVAI 算法首先在贪心选择之前按照重要度完成对属性的排序，在贪心时只需按照顺序直接计算即可而 FHARA 算法需要在贪心过程中计算属性重要度，且该计算取决于样本的个数和属性的个数，样本个数和属性个数越多，FHARA 算法对属性重要度的计算时间就越长。因此，当数据集的样本个数较多、属性个数较多时，FRABVAI 算法和 FHARA 算法的运行时间的对比就越明显。

结束语　为了更快速、全面地求得数据集的约简属性，提出一种基于投票式属性重要度的约简算法 FRABVAI，且通过多个 UCI 标准数据集验证了该算法在没有降低分类精度的情况下能更快速地得到约简属性。但是当数据集过大时，对属性重要度的判断仅仅依靠两个点可能会比较单一，如何更精确地刻画属性重要度并提高约简的速度和分类的精度需要进一步研究。

参 考 文 献

[1] PAWLAK Z, SO-WINSKI R. Rough set approach to multi-attribute decision analysis[J]. European Journal of Operational Research, 1994, 72(3): 443-459.

（下转第 229 页）