基于块集的邻域粗糙集的快速约简算法

娄 畅 刘遵仁 郭功振

(青岛大学信息工程学院 青岛 266071)

摘 要 δ-邻域计算是邻域粗糙集模型中操作最为频繁和复杂的步骤。针对当前邻域算法的研究现状,根据样本空间的分布,提出了块集的概念,证明了每个样本的邻域只存在于其相邻的块集中。在此基础上,提出了基于块集的邻域粗糙集快速约简算法,降低了计算邻域的时间复杂性,并利用多个UCI标准数据集对该算法进行了验证。结果表明,该算法是有效的、可行的。

关键词 粗糙集,邻域,属性约简,块集,快速算法

中图法分类号 TP391 **文献标识码** A

Quick Attribute Reduct Algorithm on Neighborhood Rough Set Based on Block Set

LOU Chang LIU Zun-ren GUO Gong-zhen

(College of Information Engineering, Qingdao University, Qingdao 266071, China)

Abstract Calculating each record's δ-neighborhood elements is the most frequent and complex step in neighborhood rough set model. In this paper, we proposed the concept of the block sets according to the distribution of records in the space, then proved that each record's δ-neighborhood elements can only be contained in its own block set and its adjacent block sets. Based on the block-set-neighborhood theory, we presented a quick attribute reduct algorithm on neighborhood rough set, which can reduce the complexity of calculating each record's δ-neighborhood elements. Moreover, the algorithm's validity was verified by several data sets from UCI. Expermental results show that our algorithm is effective and feasible.

Keywords Rough set, Neighborhood, Attribute reduct, Block set, Efficient algorithm

1 引言

Pawlak教授于1982年提出来的经典粗糙集理论[2],将研究对象的全体称为论域,他采用等价关系将论域粒化为若干互斥的等价类,作为描述论域中任意概念的基本信息粒子。目前该理论已经被广泛地应用于属性选择、规则学习和分类器设计等各方面。但是作为一种有效的粒度计算模型,经典粗糙集理论定义在严格的等价关系基础上,只适合处理离散型数据,对于现实中广泛存在的连续型数据却不能直接处理,这严重影响了该方法的实际应用。为了应用该方法处理连续型数据,只能将连续型数据离散化,但这种转换可能丢失重要信息[3],并且计算结果的好坏在很大程度上取决于离散化的方法。因此,邻域粗糙集被引入进行数值属性约简。

邻域粗糙模型方法直观,可以直接处理连续型属性值,而无需进行离散化处理,与经典粗糙集相比,其省去了离散化的过程,从而拓展了Pawlak经典粗糙集的应用范围。

知识约简是指在保持对论域分类能力不变的情况下,删除冗余的知识。该问题是粗糙集理论的核心问题之一。目前,尚未有一个通用的约简算法能够找到最优约简,研究证明,求属性集合的最小约简是NP-难问题,计算量随属性个数呈指数级增长。在邻域粗糙集模型中,由于要通过计算距离来确定样本间的相邻关系,因此邻域模型下的计算量要比经典离散空间下大得多。

本文在邻域粗糙集模型下,首先通过将样本映射到相应的块集中,在相邻的块集中计算某个样本的邻域,从而减少搜索空间,并用实验验证了该算法对提高计算效率的有效性。

2 邻域粗糙集模型

2.1 邻域粗糙集的概念

定义1(度量空间) 给定一个N维的实数空间Ω,$\Delta:R^N\times R^N\to R$,我们称$\Delta$是$R^N$上的一个度量,$\forall x_1,x_2,x_3\in R^N$,$\Delta$满足:

(1)$\Delta(x_1,x_2)\geqslant 0$,$\Delta(x_1,x_2)=0$当且仅当$x_1=x_2$;

(2)$\Delta(x_1,x_2)=\Delta(x_2,x_1)$,$\forall x_1,x_2\in R^N$;

(3)$\Delta(x_1,x_3)\leqslant\Delta(x_1,x_2)+\Delta(x_2,x_3)$。

我们称(Ω,Δ)为度量空间,本文将以实数空间中最常用的欧氏距离作为距离度量。

定义2 给定实数空间上非空有限集合$U=\{x_1,x_2,\cdots,x_n\}$,对于U上的任意样本x_i,定义其δ-邻域为$\delta(x_i)=\{x\mid x\in U,\Delta(x,x_i)\leqslant\delta\}$,其中$\delta\geqslant 0$。$\delta(x_i)$称为由$x_i$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$NDT=(U,C\cup D,V,f)$,其中U为论域,C和D分别是条件属性集和决策属性集,且$C\cap D\neq\emptyset$,C

娄 畅(1990—),女,硕士生,主要研究方向为粗糙集理论、数据挖掘、智能信息处理,E-mail:louchanglove@163.com。

$\neq\emptyset, D\neq\emptyset$；$V$ 是信息函数 f 的值域。

定义 4 给定一个邻域决策系统 $NDT=(U,C\cup D,V,f)$，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 D 关于 B 的下近似和上近似为：

$$\underline{N_BD}=\bigcup_{i=1}^{N}\underline{N_BD_i},$$

$$\overline{N_BD}=\bigcup_{i=1}^{N}\overline{N_BD_i}$$

其中，

$$\underline{N_BD_i}=\{x_i\mid\delta_B(x_i)\subseteq D_i,x_i\in U\},$$

$$\overline{N_BD_i}=\{x_i\mid\delta_B(x_i)\cap D_i\neq\emptyset,x_i\in U\}$$

在欧氏空间下，$\delta_B(x_i)$ 的计算方法为：$\delta_B(x_i)=\{x\mid f(B(x_i),B(x))\leqslant\delta,x\in U\}$。

决策 D 关于 B 的边界域为 $BN(D)=\overline{N_BD}-\underline{N_BD}$，正域为 $Pos_B(D)=\underline{N_BD}$。

3 邻域粗糙集的快速约简算法

3.1 邻域算法的研究现状

在邻域粗糙集理论的约简算法[1]中，邻域计算是整个约简过程中最频繁的操作之一。

文献[5]中给出的 F2HARNRS 算法[5]根据增加新的样本不会使原来属于正域的样本变为边界样本这一性质，在计算某一个样本的属性时，不需要考虑样本空间中所有的样本，仅需要考虑边界阈的样本。此算法虽然通过减少样本空间的个数来降低计算样本邻域的计算量，但是要计算整个样本空间中所有样本的邻域，仍需要花费 $O(m|U|^2)$ 的时间复杂度。

文献[1]中提出的快速属性约简算法[1]（Fast Hash Attribute Reduct Algorithm，FHARA）根据给定的邻域决策系统 $NDT=(U,C\cup D,V,f)$ 及距离度量 δ，将论域 U 中的样本划分到有限集合 $B_0,\cdots,B_k$。其中 $B_k=\{x_i\mid x_i\in U\wedge\lceil f(x_0,x_i)/\delta\rceil=k\}$，而 x_0 是论域中构造的一个特殊样本，$\forall a\in C$，$a(x_0)=\min[a(x_i)],x_i\in U$。划分结果如图 1 所示。

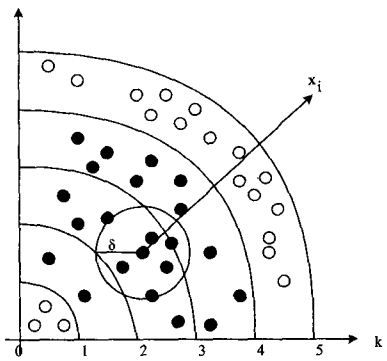

图 1 FHARA 算法划分示意图

在计算某个样本 x_i 的邻域时，只需要考虑相邻的圆环集中的样本。这样缩小了邻域的可能存在范围，减少了计算空间，计算复杂性降为 $O(m\frac{|U|^2}{k})$，其中 k 为分成的 buckets 的个数，即 $k=\left\lceil\frac{f_{\max}(x_k-x_0)}{\delta}\right\rceil$。但在这些圆环集中，依然存在大量不属于 x_i 邻域的样本。

因此，本文提出块集的概念，并基于块集提出一种新算法，以使空间划分更合理，提高计算样本邻域的效率。

3.2 基于块集的快速邻域算法

在本节，将介绍一种基于块集的快速计算邻域的算法，在求样本邻域之前，将整个样本空间划分到块集中。首先，给出块集的定义。

定义 5 给定一个邻域决策系统 $NDT=(U,C\cup D,V,f)$ 及一个距离度量 $\delta(\delta\neq0)$，其中属性集 C 表示为 $C=\{a_1,a_2,\cdots,a_m\}$，则论域 U 就可以将样本划分为块集 $\{B(k_1,k_2,\cdots,k_m)\mid k_1,k_2,\cdots,k_m\in N\}$，其中 $B(k_1,k_2,\cdots,k_m)$ 的定义为：

$$B(k_1,k_2,\cdots,k_m)=\{x_i\mid x_i\in U\wedge k_1=\left\lfloor\frac{a_1}{\delta}\right\rfloor\wedge\cdots\wedge k_m=\left\lfloor\frac{a_m}{\delta}\right\rfloor\}$$

通过定义 5 可知，每个样本可以根据其属性上的取值映射到一个相应的块集上，如图 2 所示。

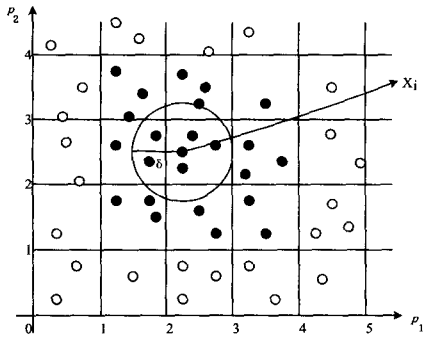

图 2 基于块集的邻域算法划分示意图

如图 2 所示，在二维空间中，根据定义 5，假设样本 x_i 被划分到块集 $B(p_1,p_2)$ 中。由图 2 可以观察出，该样本点 x_i 的邻域只可能分布在 $B(p_1-1,p_2-1)$，$B(p_1-1,p_2)$，$B(p_1-1,p_2+1)$，$B(p_1,p_2-1)$，$B(p_1,p_2)$，$B(p_1,p_2+1)$，$B(p_1+1,p_2-1)$，$B(p_1+1,p_2)$，$B(p_1+1,p_2+1)$ 这相邻的 9 个块集中。也就是说，某个样本的邻域只可能存在于每个属性值上最多相差一个 δ 距离的块集中。

定理 1 $\forall x_i\in B(q_1,q_2,\cdots,q_m)$，$x_i$ 的 δ 邻域就包含在相邻的块集：$B(q_1-1,q_2-1,\cdots,q_m-1),\cdots,B(q_1,q_2,\cdots,q_m),\cdots,B(q_1+1,q_2+1,\cdots,q_m+1)$ 中。

证明：假设样本 $x_i\in B(q_1,q_2,\cdots,q_k,\cdots,q_m)$，$x_j\in B(q_1,q_2,\cdots,q_k+2,\cdots,q_m)$，即样本 x_i 与 x_j 被划分到的块集仅在 q_k 上的取值不同，而 x_i 在属性 a_k 上的取值 $a_{i,k}$ 与 δ 满足 $q_k\delta\leqslant a_{i,k}<(q_k+1)\delta$，$x_j$ 满足 $(q_k+2)\delta\leqslant a_{j,k}<(q_k+3)\delta$。则 $a_{j,k}-a_{i,k}>\delta$，由欧氏距离公式得出：

$$f(x_i,x_j)=\sqrt{(a_{(i,1)}-a_{(j,1)})^2+\cdots+(a_{(i,m)}-a_{(j,m)})^2}$$

即：$f(x_i,x_j)>\sqrt{\delta^2}=\delta$，所以 $f(x_i,x_j)>\delta$。

该定义将论域 U 中的所有样本划分到一系列的块集中，相邻的块集仅在某一个或者某几个属性的取值上相差一个 δ，而在其他属性上的取值相同。在求某个样本的邻域时，只需要搜索在每个属性上最多相差一个 δ 的块集中的样本。

3.3 性能分析

基于块集的快速计算正域的算法，首先根据样本在欧氏空间中的位置划分到相应的块集中，对于某一个属性 a_i，最多可以划分为 $\left\lceil\frac{\max(a_i)-\min(a_i)}{\delta}\right\rceil$ 个段。设划分的块集的最大个数为 k'，则 $k'=\prod_{i=1}^{m}\left(\left\lceil\frac{\max(a_i)-\min(a_i)}{\delta}\right\rceil\right)$ 个。要获得某个样本的邻域，根据定理 1，需要搜索 3^m 个块集，所以计算的时

间复杂度为 $O(\frac{m|U|^2}{k'}\cdot 3^m)=O(\frac{m|U|^2}{\prod_{i=1}^{m}(\lceil\frac{\max(a_i)-\min(a_i)}{3\delta}\rceil)})$。因为 $\prod_{i=1}^{m}(\lceil\frac{\max(a_i)-\min(a_i)}{3\delta}\rceil)$ 趋近于 $|U|$，所以时间复杂度为 $O(m|U|)$。

而在 FHARA 算法[1] 中，样本空间被划分到 $k=\frac{f_{\max}(x_k-x_0)}{\delta}\leqslant\frac{\sqrt{\sum_{i=1}^{m}[\max(a_i)-\min(a_i)]^2}}{\delta}$ 个集合里。在计算某个样本邻域时，只需要搜索 3 个相邻的集合，所以其计算复杂度为 $O(\frac{m|U|^2}{k})$，在极端的情况下，k 也是趋近于 $|U|$ 的，所以其时间复杂度也为 $O(m|U|)$。

两个算法的时间复杂度都为 $O(m|U|)$，但在比较次数上有较大差别。因为：$k=\frac{f_{\max}(x_k-x_0)}{\delta}\leqslant\frac{\sqrt{\sum_{i=1}^{m}[\max(a_i)-\min(a_i)]^2}}{\delta}$，所以 $k<\sum_{i=1}^{m}\frac{\max(a_i)-\min(a_i)}{\delta}<\prod_{i=1}^{m}(\lceil\frac{\max(a_i)-\min(a_i)}{\delta}\rceil)=k'$。

所以基于块集的快速计算正域的算法在空间划分上更精细化，使计算过程中的比较次数减少，计算效率提高。

3.4 阈值的选取

在邻域粗糙集中，邻域的大小即距离度量 δ 是个关键的参数[8]，它的设定将直接影响属性约简的结果。如果 δ 的取值过大，则可能对于某一条件属性集合，大部分的样本就会被划分到同一个邻域中，则使得决策属性对该属性的依赖度变得很小，从而使约简后的属性变得太少；相反，如果 δ 的取值过小，就会使得约简后的属性变得太多；当 δ 的取值降为 0 时，邻域粗糙集就等价于经典的粗糙集模型。

标准差反映了数据在平均值上的平均波动大小。当标准差较小时，所有的数据都接近于平均值，这时需要为邻域设定一个较小的阈值；相反，当标准差较大时，这个阈值也要变大。正是由于阈值与标准差之间存在的关系，我们将每一列的属性值取标准差之后，再将这些标准差取标准差作为阈值 δ。

3.5 基于块集的快速属性约简算法

根据本文给出的块集的概念，给出快速计算正域的算法 $Pos(U,P,D,\delta)$。

算法 1

输入：U,P,D,δ
输出：$Pos_P(D)$
Step 1　对于每一个 $x_i\in U$，根据定义 5 做映射到一个块集 $B(k_1,k_2,\cdots,k_m)$ 中。
Step 2　初始化 Pos=∅
Step 3　for each $x_i\in U$(also,$x_i\in B(k_1,k_2,\cdots,k_m)$)
　　flag=0;
　　for each $x_i\in B(q_1-1,q_2-1,\cdots,q_m-1),\cdots,B(q_1,q_2,\cdots,q_m),\cdots,B(q_1+1,q_2+1,\cdots,q_m+1)$
　　if $f(P(x_i),P(x_j))\leqslant\delta$ and $D(x_i)\neq D(x_j)$
　　flag=1;
　　break;
　　end if
　　end for
　　if(flag=0)
　　Pos←x_i;
　　end if
end for
Step 4 return Pos

结合文献[5]提出的 F2HARNRS 算法及基于块集的快速计算正域算法，下面给出基于块集的邻域粗糙集的快速属性约简算法(B_FHARA 算法)。

算法 2

输入：决策表(U,C,D,V,f)
输出：属性子集 red
Step 1　初始化 red=∅，初始化待检验样本 smp_chk=U
Step 2　while smp_chk≠∅
　　for each $k_i\in(C-red)$
　　$Pos_i=Pos(smp\_chk,red\cup k_i,D,\delta)$;
　　if $|max\_pos|<|Pos_i|$
　　$max\_pos=Pos_i$
　　$max\_i=k_i$
　　end if
　　end for
　　if max_pos≠∅
　　red=red∪max_i;
　　smp_chk=smp_chk−max_pos;
　　else
　　break;
　　end if
end while
Step 3　return red

在该算法下，假设决策系统中有 m 个属性，约简结果中包含 k 个属性，且每增加一个属性即有 $\frac{|U|}{k}$ 个样本转化为正域，则约简的计算量为：

$$m|U|+(m-1)|U|\frac{k-1}{k}+\cdots+(m-k)|U|\frac{1}{k}<\frac{m|U|(1+\cdots+k)}{k}=\frac{m|U|(1+k)}{2}$$

4 实验分析

4.1 数据集

为了验证该算法的有效性，从 UCI 数据集中选取了 3 个非常典型的数据集作为实验的数据，即 Wine 数据集、Ionosphere 数据集和 Wdbc 数据集。表 1 详细给出了 3 个数据集的样本个数、属性个数以及各自的类别种类。

表 1　数据集描述

No	数据集	对象数	属性数	类别
1	Iris	152	4	3
2	Wine	178	13	3
3	Ionosphere	351	34	2
4	Wdbc	569	32	2
5	pima-indians-diabetes	769	8	2

本次实验环境为 CPU：酷睿 i7；内存：8G；系统：Mac OS X10.9；Xcode 5.3。采用的距离表示为最常用的欧氏距离。

4.2 实验结果

将 FHARA 算法与 B_FHARA 算法进行了实验对比 S，本次实验采用 C++编程，在计算属性约简之前，为了去掉量

(下转第 363 页)

Verification. Springer Berlin Heidelberg, 2013: 151-167

[47] Stoller S D, Bartocci E, Seyster J, et al. Runtime verification with state estimation[C]// Runtime Verification. Springer Berlin Heidelberg, 2012: 193-207

[48] Bonakdarpour B, Fischmeister S. Runtime monitoring of time-sensitive systems[C]// Runtime Verification. Springer Berlin Heidelberg, 2012: 19-33

[49] Fischmeister S, Lam P. Time-aware instrumentation of embedded software[J]. IEEE Transactions on Industrial Informatics, 2010, 6(4): 652-663

[50] Fischmeister S, Lam P. On time-aware instrumentation of programs[C]// 15th IEEE Real-Time and Embedded Technology and Applications Symposium, 2009 (RTAS 2009). IEEE, 2009: 305-314

[51] Bartocci E, Grosu R, Karmarkar A, et al. Adaptive runtime verification[C]// Runtime Verification. Springer Berlin Heidelberg, 2013: 168-182

[52] Colombo C, Pace G J, Abela P. Compensation-aware runtime monitoring[C]// Runtime Verification. Springer Berlin Heidelberg, 2010: 214-228

[53] Demsky B, Zhou J, Montaz W. Recovery tasks: an automated approach to failure recovery[C]// Runtime Verification. Springer Berlin Heidelberg, 2010: 229-244

[54] Colombo C, Pace G J. Fast-Forward Runtime Monitoring—An Industrial Case Study[C]// Runtime Verification. Springer Berlin Heidelberg, 2013: 214-228

(上接第 339 页)

纲对数据的影响，先对样本数据进行了归一化处理，并根据 3.4 节给出的方法计算了阈值 δ，实验运行结果如表 2 所列。

表 2 约简后属性集合

数据集	δ	FHARA 算法	B_FHARA 算法
Iris	0.0547	0,2	0,2
Wine	0.0310	1,7,10	1,7,10
Ionosphere	0.0275	4,12,21	4,12,21
Wdbc	0.0340	2,24,26	2,24,26
pima-indians-diabetes	0.026	0,1,5,7	0,1,5,7

由表 2 可知，FHARA 算法与 B_FHARA 算法在阈值取值相同的情况下，运行结果是一致的。所以结果的属性依赖度也是一致的。但是在运行时间上，B_FHARA 算法则要快很多，如表 3 所列。

表 3 运行时间(单位：s)

数据集	FHARA 算法	B_FHARA 算法
Iris	0.016	0.011
Wine	0.061	0.053
Ionosphere	0.678	0.526
Wdbc	1.541	0.921
pima-indians-diabetes	0.880	0.503

折线图如图 3 所示。

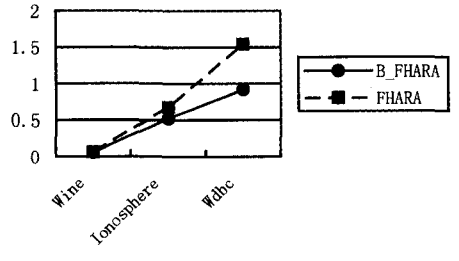

图 3 两种算法在不同数据集上运行时间的折线图

通过对比运行时间，可以发现随着样本属性的增多及样本数量的增长，B_FHARA 算法的增长速度相对较平缓。

为去除程序差异造成的计算时间差别，实验统计了样本的比较次数，如表 4 所列。

表 4 比较次数

数据集	FHARA 算法	B_FHARA 算法
Iris	15968	11221
Wine	58217	22849
Ionosphere	477927	150994
Wdbc	1106403	540604
pima-indians-diabetes	692879	131517

柱状图如图 4 所示。

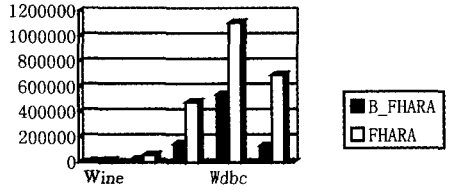

图 4 两种算法在不同数据集上比较次数的比较

通过比较次数的对比，我们发现 B_FHARA 算法能明显地减少约简过程中的比较次数。

结束语 本文研究了当前邻域粗糙集模型中邻域计算的研究现状，针对当前算法的不足，根据样本空间的分布，提出了块集的概念及其与样本的映射算法，并进一步证明了每个样本的邻域只存在于其相邻的块集中，在此基础上，提出了一种基于块集的属性快速约简算法。

经过多个 UCI 标准数据集的实验验证，在运算结果相同的情况下，该算法能有效地减少比较次数，提高计算效率。

参 考 文 献

[1] Yong L, Wenliang H, Yunliang J, et al. Quick attribute reduct algorithm for neighborhood rough set model[J]. Information Sciences, 2014, 271: 65-81

[2] Pawlak Z. Rough Sets—Theoretical Aspects of Reasoning about Data[M]. Dordrecht: Kluwer Academic, 1991

[3] 曾宇. 高效能计算机若干关键技术的研究与实现[D]. 北京：中国科学院，2009

[4] 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

[5] 胡清华，赵辉，于达仁. 基于粗糙集的符号与数值属性的快速约简算法[C]// 第七届中国 Rough 集与软计算学术会议. 太原，2007：640-649

[6] 刘遵仁，吴耿锋. 基于邻域粗糙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

[7] Hu Q, Yu D, Liu J, et al. Neighborhood rough set based heterogeneous feature subset selection[J]. Inform. Sci., 2008, 178(18): 3577-3594

[8] 张冬雯，王鹏，仇计清. 基于邻域粗糙集和蚁群优化的属性约简算法[J]. 河北科技大学学报，2011，32(5)：403-408

[9] 王丽娟，吴陈，杨习贝，等. 邻域系统粗糙集和覆盖粗糙集[J]. 计算机科学，2013，40(1)：221-224