计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201000053

基于 k-原型聚类和粗糙集的属性约简方法

李 艳[1,2] 范 斌[2] 郭 劼[2] 林梓源[1] 赵 婴[1]
1 北京师范大学珠海分校应用数学学院 广东 珠海 519087
2 河北大学数学与信息科学学院 河北 保定 071002

摘 要 基于 k-原型聚类和等价关系下的粗糙集理论,对含有连续值和符号值的目标信息系统提出了一种新的适用于混合数据的属性约简方法。首先,k-原型聚类可以通过定义混合数据的距离而得到信息系统的类簇,形成对论域的划分。将所得到的类簇代替粗糙集理论中的等价类,提出基于聚类的近似集、正域以及正域约简的概念,并根据信息熵定义属性重要性度量,建立了变精度正域约简方法。这种属性约简可以同时处理数值型和符号型数据,去除其中的冗余属性,提高分类性能,降低存储和算法运行时间耗费,并通过调节聚类参数 k 得到对论域不同粒度的划分,对所得到的约简进行优化。最后在 UCI 数据集上进行了大量的实验,针对分类问题采用了常见的 4 种分类算法,比较了约简前后的分类精度,详细分析了参数对结果的影响,验证了约简方法的有效性。

关键词:属性约简;k-原型聚类;粗糙集;混合数据;多粒度
中图法分类号 TP181

Attribute Reduction Method Based on k-prototypes Clustering and Rough Sets

LI Yan[1,2],FAN Bin[2],GUO Jie[2],LIN Zi-yuan[1] and ZHAO Zhao[1]
1 School of Applied Mathematics,Beijing Normal University,Zhuhai,Zhuhai,Guangdong 519087,China
2 College of Mathematics and Information Science,Hebei University,Baoding,Hebei 071002,China

Abstract For target information systems containing both continuous and symbolic values,a novel attribute reduction method is proposed based on k-prototypes clustering and rough set theory under equivalent relations,which is suitable for hybrid data. Firstly,k-prototypes clustering is applied to obtain clusters of information systems by defining the distance of hybrid data,forming a division of the universe. Then the obtained clusters are used to replace equivalent classes in rough set theory,and the concepts of cluster-based approximate set,positive region,attribute reduction are correspondingly proposed. An attribute importance measure is also defined based on information entropy and the clusters. Finally,a variable precision positive-region reduction method is established,which can process both numerical and symbolic data,remove redundant attributes,reduce the needed storage and running time cost,and improve classification performance of classification algorithms. Besides,the division of different granularities of the universe can be obtained by adjusting the clustering parameter k and thus the attributed reduction can be optimized. A large number of experiments are carried out on 11 UCI data sets,four common classification algorithms are used for classification problems. The classification accuracy before and after reduction are compared. The influence of parameters on the results is analyzed in detail and verifies the effectiveness of the reduction method.

Keywords Attribute reduction,k-prototypes clustering,Rough set,Hybrid data,Multi-granule

1 引言

传统的粗糙集理论[1-2]是一种能够处理知识粒度较粗所带来不确定性的数学工具。属性约简[3-6]是其重要应用。实际问题中所收集的数据往往含有噪音或冗余属性,会造成时间空间上不必要的耗费,甚至会降低分类算法的精度。粗糙集理论基于等价关系、近似集、正域等概念建立了一套属性约简方法,可以在保持核心信息的前提下最大限度地去除冗余属性。但传统粗糙集理论只能处理离散的符号值属性,Greco等提出用偏好关系来刻画带有序信息的属性,代替经典粗糙集中的等价关系,形成了基于优势关系的粗糙集理论[7-9]。对于同时含有符号值和连续值属性的数据,也可以结合等价关系和优势关系定义优势-等价关系,用于多标准决策问题[10]。但对于很多实际数据来说,连续值属性的取值之间并不一定都带有某种偏好,如鸢尾花分类数据中,花萼的长度是连续型属性,但是取值之间并没有优劣关系,而是长度介于某个范围

基金项目:广东省自然科学基金(2018A0303130026);河北省自然科学基金(F2018201096);国家自然科学基金(61976141);河北省教育厅科学技术研究重点项目(ZD2019021)
This work was supported by the NSF of Guangdong Province(2018A0303130026),NSF of Hebei Province(F2018201096),National Nautral Science Foundation of China(61976141) and Key Science and Technology Foundation of the Educational Department of Hebei Province(ZD2019021).
通信作者:李艳(ly@hbu.edu.cn)

则倾向于被划入某类鸢尾花。此时，用基于距离的相似关系去刻画这种连续值属性会比优势关系更加合理。而大部分聚类方法[11-13]则正是基于某种相似度来生成类簇，对论域形成划分。

经典的 k 均值(k-means)聚类算法[11]只能处理连续值属性，而没有考虑符号值。1998 年 Huang 等人提出了一种处理符号值属性数据的 k-mode 算法，并结合 k-means 算法和 k-mode 算法，提出了可以处理包含符号值和连续值属性数据的 k-prototypes 算法[14-15]，也称为 k-原型聚类算法。可以很方便的对混合数据进行聚类。近期有学者将聚类和属性约简相结合，如采用聚类的思想，在不降低分类性能的前提下，降低求解约简的时间消耗[16]；也有研究基于与用户交互所定义的属性依赖度来对属性进行聚类，再从每类属性中选取约简属性[17]；为了解决核属性不存在的问题，文献[18]基于属性区分度和聚类率确定约简属性集中添加的第一个属性；文献[19]用聚类算法首先对连续值属性进行离散化，减小噪音和孤立点对属性约简过程的影响。以上大部分是将聚类作为一种预处理手段，且只用于包含连续值属性的数据中。

本文拟针对混合数据，将 k-原型聚类与粗糙集中的约简方法相结合，把聚类产生的类簇作为论域的划分，建立正域约简方法。k-原型聚类算法可以方便地同时处理符号值和连续值属性，且可通过设定参数 k 可得到不同粒度的簇，作为论域中的基本概念(信息粒)，代替粗糙集中的等价类/优势类，再进一步进行变精度属性约简。通过不同粒度的类簇可以调节约简的效果，建立一种基于多粒度聚类的属性约简方法。

2　基本概念和算法

2.1　基于粗糙集的正域属性约简

定义 1(目标信息系统)[5,10]　目标信息系统 S 是一个四元组 $S=(U,A\cup D,V,f)$，其中 $U=\{x_1,x_2,\cdots,x_n\}$ 为对象的非空有限集合，称为论域；$A\cup D$ 是一个有限属性集合，其中 A 为条件属性集，D 为决策属性集；V 是属性集对应的值域；f 是一个信息函数，它指定 U 中每一个对象 x 的属性值，即对 $x\in U,a\in A,f_a(x)\in V_a$，$V_a$ 为属性 a 的值域。

定义 2(等价关系)[1]　S 为一个目标信息系统，对于 $B\subseteq A$，令：

$$R_B=\{(x_i,x_j)\in U\times U:f_l(x_i)=f_l(x_j),\forall a_l\in B\}$$

R_B 称为目标信息系统 S 定义在 B 上的等价关系。

基于定义 2，记 $[x_i]_B=\{x_j\in U:(x_i,x_j)\in R_B\}=\{x_j\in U:f_l(x_i)=f_l(x_j),\forall a_l\in B\}$ 为 x_i 的等价类；$U/B=\{[x_i]_B\mid x_i\in U\}$ 称为论域 U 关于等价关系 R_B(属性子集 B)的划分。

定义 3(基于等价关系的上下近似集)[1]　对于任意目标概念 $X\subseteq U$，定义 X 关于等价关系 R_B 的下近似和上近似，分别为：

$$\underline{R_B}(X)=\{x_i\in U:[x_i]_B\subseteq X\}$$

$$\overline{R_B}(X)=\{x_i\in U:[x_i]_B\cap X\neq\varnothing\}$$

其分别代表确定属于和可能属于 X 的对象集合。

定义 4(基于等价关系的相对正域)[1]　在目标信息系统中，条件属性集 A 相对于决策属性集 D 分类的正域为：

$$POS_A(D)=\bigcup_{X\in U/D}\underline{R_A}(X)$$

定义 5(基于等价关系的正域约简)[5]　设 S 为一个目标信息系统，对于 $B\subseteq A$，若 $POS_B(D)=POS_A(D)$，则 B 为正域协调集。若 B 为正域协调集，但 B 的任何真子集都不是正域协调集，则称 B 为正域约简。

基于以上概念，对于给定的目标信息系统，正域约简就是能够保持正域不变的属性子集，可以在保持数据分类能力的前提下最大限度消除冗余属性。但这种约简是建立在等价关系上的，因此只能处理符号值属性。

2.2　k-原型聚类算法

k-原型聚类(k-prototype)算法[14]是由 Huang 等提出的一种解决混合属性聚类的算法。它将 k-means 算法和 k-modes 算法结合起来，通过引入参数 γ 来决定分配数值属性和分类属性在聚类过程中的权重，因此它的优势在于可以处理含有连续值和符号值属性的混合数据。

令 $\boldsymbol{X}=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{X}_3,\cdots,\boldsymbol{X}_n\}$ 表示具有 n 个样本的数据集，其中 $\boldsymbol{X}_i=[x_{i1},x_{i2},\cdots,x_{ip},x_{i(p+1)},\cdots,x_{im}]$ 表示第 i 个样本，x_{ip} 为其第 p 个属性值，其中下标为 1 至 p 的属性为连续值属性，下标为 $p+1$ 到 m 的属性为符号值属性。令数据集的聚类数为 k，对应初始的聚类簇中心的集合为 $V=\{V_1,V_2,\cdots,V_k\}$。聚类过程中迭代聚类簇的集合记为 $Y=\{C_1,C_2,\cdots,C_k\}$。

定义 6　样本 X_i 到簇中心 V_l 的距离 $d(\boldsymbol{X}_i,V_l)$ 定义如下：

$$\begin{aligned}d(\boldsymbol{X}_i,V_l)&=\sum_{j=1}^{p}|X_{ij}-V_{lj}|^2+\gamma\sum_{j=p+1}^{m}\delta|X_{ij}-V_{lj}|^2\\&=d_1(\boldsymbol{X}_i,V_l)+\gamma d_2(\boldsymbol{X}_i,V_l)\end{aligned}$$

其中，$d_1(\boldsymbol{X}_i,V_l)$ 为欧氏距离；$d_2(\boldsymbol{X}_i,V_l)$ 为海明距离，它们分别定义了样本到聚类中心在连续值和符号值上的相异度。γ 为符号值属性权重值。

可以证明，类似 k 均值算法，聚类在每次迭代时可更新簇中心为当前各簇内样本的中心点，其中数值型属性取均值，符号值属性取众数，这样就可以逐步接近最优的簇中心。

定义 7　求解聚类优化问题的目标函数(代价函数)：

$$\begin{cases}F(\boldsymbol{X},V)=\sum_{i=1}^{n}\sum_{j=1}^{k}u_{ij}d(\boldsymbol{X}_i,V_j)\\\sum_{j=1}^{k}u_{ij}=1,u_{il}\in\{0,1\}\end{cases}$$

其中，u_{ij} 为 1 时，表示样本 $\boldsymbol{X}_i$ 在类 C_j 中；u_{ij} 为 0，则样本 $\boldsymbol{X}_i$ 不属于类 C_j。

算法步骤如下：

步骤 1　确定类簇个数 k，并为每个类簇选择 k 个初始聚类中心 $V_1,V_2,\cdots,V_k$；

步骤 2　按照定义 6 计算样本到各聚类中心的距离，并将其划分到离它最近的簇中心所代表的簇中；

步骤 3　计算当前每个簇的中心点(数值型属性上计算均值，符号值属性使用众数)，更新聚类簇中心；

步骤 4　重复步骤 2－3，直到各个聚类中数据对象稳定，使得迭代目标函数值 $F(\boldsymbol{X},V)$ 不变(或其变化在可接受范围内)，算法结束。

3　基于多粒度 k-原型聚类的正域约简

假设分类问题的数据集同时包含连续值和离散值属性，将其类别标签看作决策属性，则此数据集可被看作一个目标信息系统，可以在粗糙集框架下进行属性约简。但由于其是混合数据，不能在此信息系统中定义等价关系和等价类。在连续值上采用优势关系则可能引入不必要的序信息，而对连

续值的离散化又不可避免地造成信息丢失。因此在本节提出基于 k-原型聚类的属性约简方法。

首先采用 k-原型算法可以将混合数据集进行聚类，得到的每一个类簇本质上是相似的一些对象所构成的集合，这里的相似度可以理解为基于距离定义的。这与优势关系下的优势类有着本质区别。之后，将所得到的类簇代替正域约简方法中的等价类(见定义 2-4)，则可以定义基于聚类的上下近似、正域以及约简的定义。另外还可以通过控制聚类数目 k 得到不同粒度的类簇，对论域形成粗细不同的划分，将这些簇作为粗糙集理论中的基本概念，引入基于聚类的正域约简。以下定义所需要的一些必要的概念。

对任何给定的条件属性子集 $B\subseteq A$，记 k-原型聚类(见 2.2 节)生成的类簇集为 $\boldsymbol{\gamma}_B=\{C_1,C_2,\cdots,C_k\}$，形成对论域的一个划分。其中 C_i 为 k-原型聚类最终产生的第 i 个类簇。

定义 8(基于聚类的上下近似) 对目标信息系统 $S=(U,A\cup D,V,f)$ 和聚类后的类簇集合 Y_B，给定目标概念 $X\subseteq U$，定义 X 关于 $\boldsymbol{\gamma}_B$ 的下近似和上近似分别为：

$$\underline{appr_B}(X)=\{x\mid C_x\subseteq X,x\in C_x\},C_x\in\boldsymbol{\gamma}_B$$

$$\overline{appr_B}(X)=\{x\mid C_x\cap X\neq\varnothing,x\in C_x\},C_x\in\boldsymbol{\gamma}_B$$

其中，C_x 即包含对象 x 的类簇。

定义 9(基于聚类和信息增益的属性重要度) 在一个目标信息系统 S 中，属性 $a\in A$ 对 $X\subseteq U$ 进行划分所获得的“信息增益”定义如下：

$$Gain(X,a)=Ent(X)-\sum_{v=1}^{V}\frac{|X^v|}{|X|}Ent(X^v)$$

其中，V 表示 a 上不同值的总个数；X^v 表示样本集 X 中所有在属性 a 上取值为 a^v 的样本；$|*|$ 表示集合 $*$ 的基数。

$Ent(X)$ 为信息熵，定义如下：

$$Ent(X)=-\sum_{i=1}^{|\boldsymbol{\gamma}_A|}p_i\log_2 p_i,i=1,2,\cdots,|\boldsymbol{\gamma}_A|$$

其中，p_i 表示样本集 X 中第 i 个类簇中对象占总对象数 $|X|$ 的比例。

定义 10(基于聚类的相对正域) 在目标信息系统中，对给定的目标概念 $X\subseteq U$ 和条件属性 A 得到的类簇 $\boldsymbol{\gamma}_A$，A 关于决策属性 D 的相对正域为：

$$POS_A(D)=\bigcup_{X\in U/D}\underline{appr_A}(X)$$

定义 11(基于聚类的正域约简) S 为一个目标信息系统，对于 $B\subseteq A$，若 $POS_B(D)=POS_A(D)$，则 B 为正域协调集。若 B 为正域协调集，但 B 的任何真子集都不是正域协调集，则称 B 为正域约简。

若把类簇作为第 2 节中等价类同等处理，即把同一类簇中对象的所有属性取值看作相等，则上述约简也可由方便地由以下可辨识矩阵求到。

定义 12(基于聚类的可辨识矩阵) 给定目标信息系统 S，定义一个 $n\times n$ 的矩阵 $\boldsymbol{DM}$，第 i 行第 j 列元素为：

$$\boldsymbol{DM}(i,j)=\begin{cases}a_k\in A,f(x_i,a_k)\neq f(x_j,a_k)\wedge\\ \quad C_{x_i}\neq C_{x_j}\wedge f(x_i,D)\neq f(x_j,D)\}, & i>j\\ \varphi, & \text{else}\end{cases}$$

其中，C_{x_i}，C_{x_j} 分别为 x_i 和 x_j 所在的类簇；$f(x,D)$ 表示对象 x 在决策属性 D 上的取值。类似传统粗糙集中的可辨识矩阵，可将矩阵中单个元素的并集作为核属性，再按照属性的重要性依次将属性添加至核中，直到与辨识矩阵 $\boldsymbol{DM}$ 中的每个元素交集都不为空为止，则得到一个约简。

定义 13(变精度正域约简) S 为一个目标信息系统，对于 $B\subseteq A$ 和给定的变精度系数 $\beta\in[0,1]$，B 称为 β-正域约简若 $POS_B^\beta(D)=POS_A(D)$ 且任何 B 的真子集均不满足此式。其中：

$$POS_B^\beta(D)=\bigcup_{X\in U/D}\{x\mid x\in U,\frac{|C_x\cap X|}{|C_x|}\geqslant\beta\},C_x\in\boldsymbol{\gamma}_B$$

变精度[20]的思想是放宽对下近似的要求，即不要求下近似中的对象所属的类簇完全包含于目标概念 X 中，进而也放宽了对正域和约简的要求，更加灵活，在可容忍的分类精度范围内可去除更多冗余属性。具体地，β 越接近于 1，则所得到的约简越接近严格正域约简。当 $\beta=1$ 时即退化为定义 11 中的正域约简。

基于以上概念，我们可以建立基于 k-原型聚类的多粒度聚类的属性约简方法，算法具体步骤如算法 1 所示。

算法 1 基于多粒度 k-原型聚类和粗糙集的属性约简算法

输入：数据集 Data={$x_1,x_2,\cdots,x_n$}，属性集 A={$a_1,a_2,\cdots,a_n$}，聚类数目的范围 k_range，约简阈值取值范围 β_range

输出：属性约简 REDUCT

1. for k in k_range：
2. 对数据集进行 k-prototypes 聚类，返回带有类簇的数据集 D^*；
3. D^* 中，计算 A 中每个 a_i 的信息增益 Gain(D^*，a_i)并排序，返回排序后的属性集 A^*；
4. 计算可辨识矩阵 **DM**：
   for i in 数据行数：
     for j < i：
       for k in 属性列表：
         if $f(x_i,a_k)\neq f(x_j,a_k)\wedge C_{x_i}\neq C_{x_j}\wedge f(x_i,D)\neq f(x_j,D)$
           将 a_k 加入 **DM**(i,j)；
5. 确定核属性：
   取辨识矩阵中单点集中的元素作为核属性，将核加入约简属性集 REDUCT，然后将 **DM**(i,j) 中含有核的元素置为空集；并更新 $A^*=A^*-${核属性}；
   if REDUCT 为空：return REDUCT；
6. 变精度属性约简：
   for β in β_range：
     将 A^* 中属性按先后顺序逐个加入约简属性集 REDUCT；
     消去可辨识矩阵 **DM** 含有该属性的元素，设为空集；
     if 当前可辨识矩阵非空元素个数/原始可辨识矩阵元素个数$\leqslant(1-\beta)$：
   return REDUCT

注意：算法 1 实现中的变精度约简与定义 12 中的概念不完全相同，是为了计算方便，但两者本质上都是放宽对约简的要求，提高约简效果。$\beta=1$ 时都退化为严格正域约简。

算法 1 的步骤说明如下。

首先，使用 k-prototypes 算法将数据聚类。

确定聚类的数目 k，选定 k 个初始的聚类簇中心 V_1，$V_2,\cdots,V_k$，按照定义 6 计算样本到各簇中心的距离，并将其划分到离它最近的簇中心所代表的簇中，重新计算聚类簇中心。进行迭代直到各个聚类中数据对象稳定，迭代目标函数值 $F(X,V)$ 不变。

其次，基于聚类计算属性重要度。

以 k-prototypes 算法增益生成的标签为类标签，根据定义 9 计算每个属性的信息增益作为属性重要性度量，并按信

息增益的大小对属性从大到小排序。

然后，基于聚类进行属性正域约简。

(1)把得到的类簇作为基本概念(信息粒)，对论域形成一个划分。从小到大设定 k 的数值，使得划分由粗到细。

(2)计算每个目标概念的上下近似及信息系统的正域。

(3)对按属性重要度排序后的属性集进行约简。这里采用基于等价关系的变精度约简算法[20]，通过使用聚类代替其中的等价类，得到属性约简。这里变精度的作用是通过设置变精度系数 β 放宽对正域的要求，最终可得到一个近似正域约简。

最后，选取约简效果最优的划分粒度。

在 k 变化过程中，类簇的大小发生变化，会使得划分粗细不同。这里选择使得约简效果最好的 k，对应的约简作为最终结果。

4　实验分析

为了验证所提方法的效果，取 UCI 数据库[21]中的 11 个数据集，样例数量从 132 到 32561，且同时包含连续值属性和无序的符号值属性，属性个数从 4 个到 25 个不等，均为分类问题，数据集详细信息见表 1。我们分别在这些数据集上实现所提出的基于聚类和粗糙集的属性约简方法，通过 4 种分类算法所得到的分类精度来比较约简前后的效果。所有实验均用 Python3 进行编程，环境配置为 Intel(R)Core(TM)i7-8550U CPU@1.8GHz 1.99GHz，内存 8.0GB。

表 1　实验数据集

Table 1　Experimental data set

数据集	样例数	分类属性#	连续值属性#	属性总个数	类别数
hayes-roth	132	1	3	4	3
crx	666	9	6	15	2
Australian	690	8	6	14	2
german	1000	13	7	20	2
messidor_features	1151	3	16	19	2
cmc	1473	7	2	9	3
seismic_bumps	2584	3	15	18	2
abalone	4177	1	7	8	29
Frogs_MFCCs	7195	3	22	25	60
letter-recognition	20000	2	14	16	27
adult	32561	8	6	14	2

4.1　约简效果

首先，根据数据集中的条件属性实现第 3 节中的多粒度 k-原型聚类约简算法，得到近似正域约简，即一个属性子集。再使用 4 种常用分类方法 KNN、决策树(DecisionTree)、支持向量机(SVC)、神经网络(NN)在约简前后数据集上的分类精度来衡量约简的效果，实验结果如表 2 所列。其中“原始”指没有经过属性约简，使用所有的属性进行分类的精度；“聚类约简后”指经过多粒度聚类约简算法后，利用所得到的属性约简(属性子集)来进行分类的精度；最后一列给出了相应的 β 取值。

表 2　聚类约简前后分类精度比较

Table 2　Comparison of classification accuracy before and after reduction

数据集	KNN 精度		DecisionTree 精度		SVC 精度		NN 精度		最优 β
	原始	约简后	原始	约简后	原始	约简后	原始	约简后	
hayes-roth	0.76	0.77	0.70	0.70	0.52	0.52	0.58	0.61	0.95
crx	0.73	0.74	0.77	0.77	0.77	0.77	0.72	0.77	0.80
Australian	0.82	0.86	0.87	0.87	0.75	0.86	0.86	0.86	0.95
german	0.69	0.70	0.73	0.73	0.70	0.70	0.70	0.70	1.00
messidor_features	0.67	0.66	0.66	0.63	0.68	0.73	0.72	0.72	1.00
cmc	0.48	0.50	0.46	0.46	0.49	0.52	0.50	0.53	0.95
seismic_bumps	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	1.00
abalone	0.23	0.23	0.25	0.25	0.23	0.24	0.27	0.25	1.00
Frogs_MFCCs	0.88	0.86	0.75	0.75	0.83	0.86	0.76	0.73	1.00
letter-recognition	0.84	0.85	0.84	0.85	0.71	0.68	0.75	0.70	1.00
adult	0.82	0.83	0.84	0.83	0.78	0.79	0.83	0.82	1.00
Average	0.7155	**0.7227**	**0.7109**	0.7082	0.6736	**0.6927**	**0.6945**	**0.6945**	—

从表 2 中的实验结果可以看出，在绝大多数的数据集上和所有的分类算法中，使用多粒度聚类属性约简后的平均结果和约简前的结果相近，甚至在 KNN 和 SVC 算法中，平均精度优于原始精度。这说明我们所提出的属性约简方法可以在去除冗余信息的同时保留对分类重要的信息。从参数 β 的最优取值可以看出，大部分情况下 $\beta=1$ 得到的是严格的正域约简，这是因为最优参数的取值定义为使分类精度最大的取值，要在约简的同时最大程度地保留原始信息。

表 3 列出了使用基于聚类的方法约简前后的属性个数以及约简率(为约简后的个数/原始属性个数)，来体现约简后对数据起到的压缩效果。其 11 个数据集上的平均约简率达到 72%，在数据集 german 上达到 95%，在 seismic_bumps 上达到 94%，在保留了核心信息的情况下，明显降低了存储，从而也会使得分类算法在约简后的数据上运行的效率得到明显提升。

表 3　基于聚类方法的属性个数约简率

Table 3　Reduction rate of attribute number based on clustering method

数据集	原始个数	约简后个数	约简率
hayes-roth	4	2	0.5
crx	15	2	0.87
Australian	14	2	0.85
german	20	1	0.95
messidor_features	19	6	0.68
cmc	9	4	0.56
seismic_bumps	18	1	0.94
abalone	8	1	0.88
Frogs_MFCCs	25	10	0.60
Frogs_MFCCs	16	7	0.56
letter-recognition	14	6	0.57

为了体现属性约简后分类算法在数据上的运行效率，使用时间约简率(约简后节省的运行时间/原始未约简的运行时

间)来衡量,结果如表 4 所列。其中"约简前"指没有经过属性约简,使用 4 种分类算法计算精度的总时间;"约简后"指经过多粒度聚类约简算法后,利用所得到的属性约简(属性子集),使用 4 种分类算法计算精度所花费的总时间。可见,时间约简率平均为约 51%,最高达到了 75%;而在最后 3 个较大的数据集上,约简率分别为 62%,44%和 37%,约简效果明显。不同数据集由于所含的属性冗余程度不同,被约简属性的数量有所不同(如 german 原有 20 个属性可约简掉 19 个,约简效果明显;而 cmc 中原有的 9 个属性只被约简 4 个),而且不同分类算法的运行效率与数据集的类别数及样本分布都有关系,因此约简后的时间对比(见表 4)在不同数据集上会有波动。

表 4 属性约简前后分类算法的运行时间

Table 4 Running time of classification algorithm before and after attribute reduction

数据集	运行时间/s		时间约简率
	约简前	约简后	
hayes-roth	53.68	24.17	0.55
crx	121.76	45.63	0.63
Australian	146.09	87.19	0.40
german	209.90	52.99	0.75
messidor_features	266.11	152.08	0.43
cmc	326.82	289.65	0.11
seismic_bumps	417.81	133.28	0.68
abalone	3823.92	1329.36	0.65
Frogs_MFCCs	8517.59	3270.41	0.62
letter-recognition	24690.68	13774.68	0.44
adult	30017.67	18959.02	0.37
Average	**6235.64**	**3465.31**	**0.51**

把数据集的样例数(行数)*属性数(列数)作为数据集的规模,按从小到大排列。以数据规模为横轴,4 种分类算法在 11 个数据集上的运行时间为纵轴绘制图 1,时间单位为秒(s)。处于上方的蓝色折线表示属性约简前精度计算时间,下方红色折线表示属性约简后精度计算时间。可以看出,运行时间随着数据规模的增大而近似呈线性增加的趋势,可见所提出算法可以用在较大数据集上。

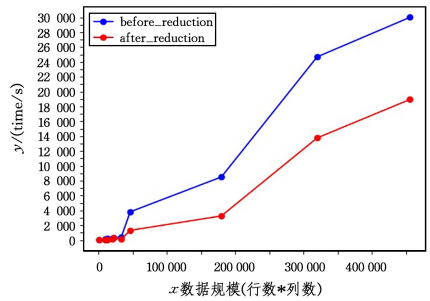

图 1 约简前后算法运行时间与数据规模(电子版为彩色)

Fig. 1 Running time and data size of algorithm before and after reduction

4.2 参数的影响

基于多粒度聚类的属性约简算法中主要涉及两个参数:聚类个数 k 和变精度阈值 β。k 的大小影响到聚类后的类簇大小、对论域划分的粗细,以及最后属性约简所用到的监督信息。而变精度参数 β 决定了对正域约简的放松程度,$\beta=1$ 时所求得的约简是严格的正域约简,$\beta<1$ 时则是近似约简,一般会含有更少的属性,约简率更高,精度可能略低。

首先,本节设置聚类个数从 2 到 7 进行变化,保持 $\beta=1$ 不变的情况下,观察相对应的分类性能是否稳定,其结果如表 5 所列。

表 5 聚类数目 k 对精度的影响(聚类数目从 2 到 7)

Table 5 Effect of clustering number k on accuracy (number of clusters from 2 to 7)

数据集	约简后 KNN 精度							约简后 DecisionTree 精度						
	2	3	4	5	6	7	标准差	2	3	4	5	6	7	标准差
hayes-roth	0.67	0.67	0.61	0.70	0.77	0.72	0.0493	0.61	0.61	0.58	0.64	0.70	0.64	0.0374
crx	0.74	0.54	0.54	0.74	0.74	0.74	0.0943	0.77	0.60	0.60	0.77	0.77	0.77	0.0801
Australian	0.64	0.86	0.64	0.64	0.64	0.86	0.1037	0.68	0.87	0.68	0.68	0.68	0.87	0.0896
german	0.70	0.68	0.68	0.68	0.68	0.68	0.0075	0.73	0.71	0.71	0.71	0.71	0.71	0.0075
messidor_features	0.60	0.60	0.60	0.63	0.66	0.60	0.0229	0.59	0.59	0.59	0.59	0.63	0.54	0.0261
cmc	0.49	0.49	0.50	0.50	0.49	0.48	0.0069	0.46	0.46	0.46	0.46	0.46	0.46	0
seismic_bumps	0.95	0.95	0.95	0.95	0.95	0.95	0	0.95	0.95	0.95	0.95	0.95	0.95	0
abalone	0.23	0.14	0.22	0.22	0.20	0.20	0.0297	0.25	0.18	0.23	0.23	0.25	0.25	0.0248
Frogs_MFCCs	0.86	0.35	0.35	0.52	0.28	0.30	**0.2017**	0.75	0.47	0.47	0.47	0.30	0.30	**0.1503**
letter-recognition	0.85	0.79	0.78	0.77	0.79	0.78	0.0262	0.85	0.82	0.83	0.82	0.82	0.82	0.0111
adult	0.83	0.83	0.83	0.84	0.83	0.83	0.0037	0.83	0.82	0.83	0.83	0.83	0.84	0.0058
Average	**0.69**	**0.63**	**0.61**	**0.65**	**0.64**	**0.65**	**0.0496**	**0.68**	**0.64**	**0.63**	**0.65**	**0.65**	**0.65**	**0.0393**
数据集	约简后 SVC 精度							约简后 NN 精度						
	2	3	4	5	6	7	标准差	2	3	4	5	6	7	标准差
hayes-roth	0.39	0.48	0.45	0.42	0.52	0.42	0.0431	0.58	0.39	0.55	0.55	0.61	0.64	0.0797
crx	0.77	0.53	0.53	0.77	0.77	0.77	0.1131	0.77	0.61	0.53	0.77	0.77	0.76	0.0959
Australian	0.66	0.86	0.66	0.65	0.65	0.86	0.0967	0.55	0.86	0.55	0.67	0.67	0.86	0.1276
german	0.70	0.70	0.70	0.37	0.33	0.70	0.1654	0.70	0.70	0.70	0.70	0.70	0.70	0
messidor_features	0.63	0.65	0.55	0.70	0.73	0.55	0.0683	0.63	0.63	0.56	0.66	0.72	0.56	0.0559
cmc	0.50	0.47	0.41	0.49	0.48	0.52	0.0344	0.44	0.46	0.43	0.52	0.51	0.51	0.0362
seismic_bumps	0.95	0.95	0.95	0.95	0.95	0.95	0	0.95	0.95	0.95	0.95	0.95	0.95	0
abalone	0.24	0.18	0.21	0.20	0.22	0.22	0.0186	0.25	0.18	0.25	0.25	0.24	0.24	0.0250
Frogs_MFCCs	0.86	0.47	0.47	0.47	0.30	0.31	**0.1853**	0.73	0.33	0.36	0.40	0.37	0.35	**0.1388**
letter-recognition	0.68	0.70	0.72	0.70	0.76	0.75	0.0285	0.70	0.78	0.78	0.77	0.78	0.78	0.0293
adult	0.79	0.78	0.79	0.79	0.79	0.80	0.0058	0.82	0.82	0.81	0.81	0.82	0.82	0.0047
Average	**0.65**	**0.62**	**0.59**	**0.59**	**0.59**	**0.62**	**0.0690**	**0.65**	**0.61**	**0.59**	**0.64**	**0.65**	**0.65**	**0.0539**

从各分类算法所得到结果的标准差开看，聚类个数对于约简的影响比较大，在 11 个数据和 4 种分类算法上分类精度的总平均标准差大概在 0.0529(5.3%左右)。这是因为聚类所得到的类簇将作为约简过程的监督信息，直接影响到最终约简的属性子集，进而对最终分类精度起到关键作用。所以聚类个数 k 是需要进行仔细优化的参数。4 种算法中支持向量机对 k 最敏感，而决策树最不敏感，结果最稳定。并且不同的数据集上的表现差别较大，如 seismic_bumps 数据集上无论聚类个数如何变化，分类精度始终保持稳定；而 Frogs_MFCCs 的标准差最大，可能的原因是其真实类别数较大，为 60 类问题，而为了降低计算损耗我们在聚类中所设定的 k 的取值范围与 60 相差较大。

其次，取固定值 $k=2$，令变精度阈值 β 从 0.8 到 1 范围内进行取值，步长为 0.05，观察其对于结果的影响，结果如表 6 所列。其中，β_1，β_2，β_3，β_4，β_5 分别为 1，0.95，0.90，0.85，0.80；所有结果均为使用对应参数取值约简后的分类精度。

表 6　阈值 β 对精度的影响

Table 6　Influence of threshold β on accuracy

数据集	KNN						DecisionTree					
	β_1	β_2	β_3	β_4	β_5	标准差	β_1	β_2	β_3	β_4	β_5	标准差
hayes-roth	0.67	0.61	0.61	0.61	0.61	0.0240	0.61	0.58	0.58	0.58	0.58	0.0120
crx	0.74	0.74	0.74	0.74	0.74	0	0.77	0.77	0.77	0.77	0.77	0
Australian	0.64	0.64	0.64	0.64	0.64	0	0.68	0.68	0.68	0.68	0.68	0
german	0.70	0.70	0.68	0.70	0.68	0.0098	0.73	0.73	0.71	0.73	0.71	0.0098
messidor_features	0.60	0.60	0.60	0.60	0.60	0	0.59	0.59	0.59	0.59	0.59	0
cmc	0.49	0.41	0.41	0.41	0.41	0.0320	0.46	0.39	0.39	0.39	0.39	0
seismic_bumps	0.95	0.95	0.95	0.95	0.95	0	0.95	0.95	0.95	0.95	0.95	0
abalone	0.23	0.23	0.23	0.22	0.23	0.0040	0.25	0.25	0.25	0.25	0.25	0
Frogs_MFCCs	0.86	0.22	0.22	0.22	0.22	**0.2560**	0.75	0.25	0.25	0.25	0.25	**0.2000**
letter-recognition	0.85	0.62	0.37	0.37	0.37	**0.1930**	0.85	0.68	0.33	0.33	0.29	**0.2266**
adult	0.83	0.81	0.81	0.82	0.82	0.0075	0.83	0.82	0.82	0.82	0.82	0.0040
Average	**0.69**	**0.59**	**0.57**	**0.57**	**0.57**	**0.0478**	**0.68**	**0.61**	**0.57**	**0.58**	**0.57**	**0.0411**
数据集	SVC						NN					
	β_1	β_2	β_3	β_4	β_5	标准差	β_1	β_2	β_3	β_4	β_5	标准差
hayes-roth	0.39	0.45	0.45	0.45	0.45	0.0240	0.58	0.55	0.55	0.55	0.55	0.0120
crx	0.77	0.77	0.77	0.77	0.77	0	0.77	0.77	0.77	0.77	0.76	0.0040
Australian	0.66	0.65	0.65	0.65	0.65	0.0040	0.55	0.68	0.66	0.66	0.67	0.0476
german	0.70	0.70	0.70	0.70	0.70	0	0.70	0.70	0.70	0.70	0.70	0
messidor_features	0.63	0.60	0.63	0.60	0.63	0.0147	0.63	0.61	0.63	0.61	0.63	0.0098
cmc	0.50	0.49	0.49	0.49	0.50	0.0049	0.44	0.49	0.48	0.47	0.50	0.0206
seismic_bumps	0.95	0.95	0.95	0.95	0.95	0	0.95	0.95	0.95	0.95	0.95	0
abalone	0.24	0.24	0.23	0.23	0.23	0.0049	0.25	0.25	0.26	0.26	0.25	0.0049
Frogs_MFCCs	0.86	0.17	0.17	0.17	0.17	**0.2760**	0.73	0.25	0.25	0.25	0.25	**0.1920**
letter-recognition	0.68	0.25	0.14	0.14	0.11	**0.2134**	0.70	0.35	0.18	0.18	0.13	**0.2097**
adult	0.79	0.76	0.76	0.76	0.58	0.0759	0.82	0.80	0.80	0.79	0.79	0.0109
Average	**0.65**	**0.55**	**0.54**	**0.54**	**0.52**	**0.0562**	**0.65**	**0.58**	**0.57**	**0.56**	**0.56**	**0.0465**

简单计算可以得出，在 11 个数据上使用 4 种分类算法得到的精度标准差总平均为 0.0479(4.8%左右)，比聚类个数 k 对于结果的影响有所减小，但通过仔细观察发现结果的变化主要是来自于 Frogs_MFCCs 和 letter-recognition 这两个数据集，其属性个数和原始类别数目都是较大的。如果去掉这两个数据的影响，剩余 9 个数据集上的总的平均标准差为 0.0095(约 0.9%)。可见，总体来说当 β 取值发生变化时，大部分数据集和算法的结果是稳定的。如果数据集的属性个数较多或类别个数较大时，则要仔细进行参数寻优。

结束语　本文在粗糙集框架下提出一种基于多粒度聚类的属性约简方法。采用 k 原型聚类来处理连续值和符号值并存的混合数据，基于聚类相应定义了上下近似、正域、属性重要度，以及约简的概念。建立了基于聚类的变精度属性约简方法，通过调节聚类个数和变精度系数来优化所得到的近似正域约简。大量的实验结果表明，所提出方法可以在不降低原始分类性能的前提下，去除冗余属性，达到较高的约简率，这对于降低存储空间和算法运行时间耗费是很有实际意义的。另外，对于属性和类别个数较大的数据集要仔细调参，而一般数据集对参数是不敏感的。

参 考 文 献

[1] PAWLAK Z. Rough sets[J]. International Journal of Information & Computer Sciences, 1982, 11(3): 289-296.

[2] PAWLAK Z. Rough sets: Theoretical Aspects of Reasoning about Data[M]. Boston: Kluwer Academic Publishers, 1991.

[3] SKOWRON A, RAUSZER C. The discernibility matrices and functions in information systems[M]. Dordrecht: Springer, 1992: 331-362.

[4] KRYZKIEWICZ M. Comparative study of alternative types of knowledge reduction in inconsistent systems [J]. International Journal of Intelligent Systems, 2001, 16(1): 105-120.

[5] CHEN J, WANG G Y, HU J. Positive Domain Reduction Based on Dominance Relation in Inconsistent System[J]. Computer Science, 2008, 35(3): 216-218, 227.

[6] LIU G, FENG Y, YANG J. A common attribute reduction form for information systems[J]. Knowledge-Based Systems, 2020, 193: 105466.

[7] GRECO S, MATARAZZO B, SLOWINSKI R. Rough sets theory for multicriteria decision analysis[J]. European Journal of

Operational Research, 2001, 129(1): 1-47.
[8] GRECO S, MATARAZZO B, SLOWINSKI R. Rough approximation by dominance relations[J]. International Journal of Intelligent Systems, 2002, 17(2): 153-171.
[9] CAO B R, LIU Y. Variable Precision Rough Set Model Based on Set Pair Situation Dominance Relationship[J]. Computer Engineering, 2015, 41(11): 35-40.
[10] LI Y, ZHANG L, WANG X J, et al. Attribute Reduction for Sequential Three-way Decisions Under Dominance-Equivalence Relations[J]. Computer Science, 2019, 46(2): 242-248.
[11] ANDERBERG M R. Cluster Analysis for Applications[M]. New York: Academic Press, 1973.
[12] SUN J G, LIU J, ZHAO L Y. Clustering algorithms research [J]. Journal of Software, 2008, 19(1): 48-61.
[13] LIU Y H, MA H F, LIU H J, et al. An overlapping subspace K-Means clustering algorithm[J]. Computer Engineering, 2020, 46(8): 58-63.
[14] HUANG Z. Extensions to the K-means algorithm for clustering large data sets with categorical values[J]. Data Mining and Knowledge Discovery, 1998, 2(3): 283-304.
[15] HUANG Z, NG M. Fuzzy K-modes algorithm for clustering categorical data[J]. IEEE Transactions on Fuzzy Systems, 1999, 7(4): 446-452.
[16] CHEN Y, SONG J J, YANG X B. Accelerator for finding reduct based on attribute group[J]. Journal of Nanjing University of Science and Technology, 2020, 44(2): 216-223.
[17] CHEN Y, ZENG D S, XIE C. A Method of Attribute Reduction Based on Clustering[J]. Computer Systems Applications, 2009, 18(5): 173-176.
[18] LU J, ZHANG T, REN H L. Reduction of attribute in decision table based on clustering rate[J]. Computer Engineering and Application, 2012(28): 135-138, 233.
[19] CHEN Y C, LI O, SUN Y. Attribute reduction based on clustering discretization and variable precision neighborhood entropy [J]. Control and Decision, 2018, 33(8): 1407-1414.
[20] ZIARKO W. Variable precision rough set model[J]. Journal of Computer and System Sciences, 1993, 46(1): 39-59.
[21] UCI Machine Learning Repository[OL]. https://archive.ics.uci.edu/ml/index.php.

LI Yan, born in 1976, Ph.D, professor, master supervisor, is a member of China Computer Federation. Her main research interests include Granular computing and knowledge discovery and machine learning.

(上接第 333 页)

写基于 B+树存储的碰撞检测代码。实验的最终结果如图 8 所示。

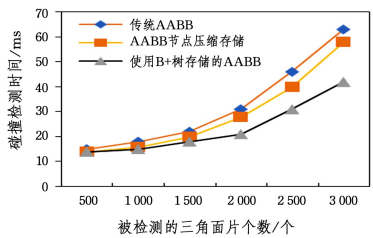

图 8 实验环境同等条件下 3 种算法的性能比较

Fig. 8 Comparison of performance of three algorithms

从图 8 可以看出，在相同的实验环境和空间中，针对相同数量的碰撞物体的碰撞检测效率而言，使用 B+树存储改进的算法相比于传统算法和结点压缩存储算法在检测时间上明显缩短，且随着被检测物体的增加，该算法的优势更加明显。

结束语 本文主要针对传统 AABB 包围盒空隙大和层次树各节点的存储空间大等问题，结合学者们对 AABB 包围盒结点压缩存储的思想，提出使用 B+树存储结点的概念，对可能发生碰撞的两个对象分别构造平衡的 AABB 层次树，通过降低每个结点存储的字节数进一步减少 AABB 树所须的内存空间大小，降低了遍历 AABB 树所需的时间，加快了算法的执行效率，最终使得碰撞检测效率得到了很大的提高。但是，在生产环境中，物体对象往往是多种形态存在，在本实现环境中涵盖的物体对象的形状有限而可能导致不同的环境下结果有略微差异。为了达到生产环境要求，后续还需要不断地进行实验对比以得出更为精确的结果。

参 考 文 献

[1] 王志强，洪嘉振，杨辉. 碰撞检测问题研究综述[J]. 软件学报，1999(5): 98-104.
[2] GARICA-ALONSO A, SERRANO N. Solving the Collision Detection Problem[J]. IEEE Computer Graphics and Application, 1994, 13(3): 36-43.
[3] 王伟. 轴对齐包围盒算法的研究[J]. 网络安全技术与应用，2013(10): 127-128.
[4] 潘振宽，李建波. 基于压缩的 AABB 树的碰撞检测算法[J]. 计算机科学，2005, 33(2): 213-215.
[5] 沈学利，吴琼. 基于包围盒和空间分割的混合碰撞检测算法[J]. 计算机工程，2012, 38(6): 256-258.
[6] LI C F, FENG Y T, OWEN D R J. SMB: Collision detection based on temporal coherence[J]. Computer Methods in Applied Mechanics and Engineering, 2005, 195(19): 2252-2269.
[7] 施恩，顾大权，冯径，等. B+树索引机制的研究及优化[J]. 计算机应用研究，2017, 34(6): 1766-1769.
[8] 王崴，周诚，杨云，等. 面向虚拟维修的碰撞检测算法[J]. 计算机应用与软件，2016, 33(4): 235-238.
[9] AKENINE-MÖLLSER T. Fast 3D Triangle-Box Overlap Testing[J]. Journal of Graphics Tools, 2001, 6(1): 29-33.
[10] 金汉均. 虚拟环境中物体碰撞检测算法研究[D]. 武汉：华中科技大学，2006.
[11] 孙黎阳，毛少杰，林剑柠，等. 基于 XYZ/ADL 的网络中心化仿真运行支撑平台体系结构形式化描述[J]. 计算机科学，2012, 39(S1): 365-369.

YANG Fan, born in 1995, postgraduate. His main research interests include artificial intelligence and big data.