计算机科学
COMPUTER SCIENCE

超约简求解:效率与性能的提升

王笑笑, 巴婧, 陈建军, 宋晶晶, 杨习贝

引用本文

王笑笑, 巴婧, 陈建军, 宋晶晶, 杨习贝超约简求解:效率与性能的提升[J]. 计算机科学, 2023, 50(2): 166-172.

WANG Xiaoxiao, BA Jing, CHEN Jianjun, SONG Jingjing, YANG Xibei. Searching Super-reduct:Improvement on Efficiency and Effectiveness [J]. Computer Science, 2023, 50(2): 166-172.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.211200292

超约简求解:效率与性能的提升

王笑笑　巴　婧　陈建军　宋晶晶　杨习贝
江苏科技大学计算机学院　江苏 镇江 212100
(w_xiaoxiao14@163.com)

摘　要　利用多重约简的结果搭建一个集成分类框架,已被证实可以显著提升后续学习的性能。超约简方法正是借鉴了这一理念,在约简求解的基础上,通过随机添加额外属性以达到获取多重超约简的目的。显然,基本的约简求解将直接影响超约简方法的效果。鉴于此,从兼顾效率和性能的角度出发,在超约简方法中同时引入属性簇和集成选择机制:属性簇用于加速基本约简的求解过程,集成选择则用于在求解过程中找到更为稳健的属性。在20组UCI数据上的实验结果表明,相比4种前沿的集成策略,所提方法不仅能够显著减少约简求解的时间消耗,而且能够提供更好的分类稳定性和准确率。

关键词:属性簇;集成选择;约简求解;超约简
中图法分类号　TP181

Searching Super-reduct:Improvement on Efficiency and Effectiveness

WANG Xiaoxiao,BA Jing,CHEN Jianjun,SONG Jingjing and YANG Xibei
School of Computer,Jiangsu University of Science and Technology,Zhenjiang,Jiangsu 212100,China

Abstract　Following the derivation of multiple reducts,an ensemble based classification framework can be constructed,which has been demonstrated to be useful in improving the performance of subsequent learning tasks. The approach called super-reduct is exactly suggested with such thinking. Generally,multiple super-reducts are obtained by randomly adding more extra attributes into the fundamental reduct. Therefore,how to search fundamental reduct is the key to performing super-reduct. In view of this,considering both efficiency and effectiveness,not only attribute group but also ensemble selector is introduced into the mechanism of super-reduct:the device of attribute group is used to speed up the process of searching fundamental reduct,the device of ensemble selector is used to find more robust attributes in the procedure of searching reduct. Comprehensive experiments on 20 UCI data sets show that compared with 4 popular strategies,our approach can not only significantly reduce the computational cost but also provide superior stabilities and accuracies for classification tasks.

Keywords　Attribute group,Ensemble selector,Searching reduct,Super-reduct

1　引言

作为粗糙集理论[1-2]研究中的核心内容,属性约简[3-4]一直是众多学者关注的焦点。本质上,属性约简可以被视为依据某种度量来移除数据中的冗余或不相关属性,从而达到筛选出满足给定约束条件的最小属性子集的目的。

在多数文献中,约简的求解方法一般可分为两大类:穷举法和启发式方法。穷举法可以获得所有约简,但其计算复杂度过高,很难满足现实世界中大规模数据处理的需求;启发式方法则是借助启发式函数进行迭代搜索,因其使用了启发式信息,故这一过程能够较快地收敛。但众所周知,利用启发式方法往往容易陷入局部最优的困境中,导致所求约简的性能不尽如人意。因此,可以认为约简的求解效率与约简的性能之间存在一种类似博弈的关系。

为了在上述博弈过程中寻求一种可接受的平衡,已有学者在属性约简的相关研究中引入了集成的理念,设计了诸如集成选择器[5-7]、基于多重约简的集成分类[8-9]等策略。其中利用多重约简的结果搭建一个集成分类框架,已被证实可以显著提升约简在后续学习器上的性能。例如,Jiang等[8]从多模态扰动的视角提出了E_RSRR方法。该方法首先通过重采样技术[10]对原始样本空间进行扰动,以获取多个样本空间;其次依据每一个扰动生成的样本空间,利用贪心的启发式搜索获取基本约简,并在此基础上,采用随机扰动[8]的方式增添额外属性,进一步生成超约简;最后利用所有超约简对测试样本分别进行预测且投票,得出最终的预测结果。

从上述过程来看,需要注意以下两点:1)E_RSRR是在多组随机采样得到的样本集的基础上,进一步获取多个约简,相较于在原始样本空间上获取一个约简而言,会显式地增加时间消耗;2)E_RSRR方法存在较大的随机性,因为不仅不同样本子空间的获取是随机的,而且由约简生成超约简的过程也

到稿日期:2021-12-27　返修日期:2022-06-28
基金项目:国家自然科学基金(62076111,61906078,62006099,62006128)
This work was supported by the National Natural Science Foundation of China(62076111,61906078,62006099,62006128).
通信作者:宋晶晶(songjingjing108@163.com)

具有较大的随机性，这会导致后续的分类、学习等任务面临着波动性较大这一现实问题。

为了解决上述问题，本文将在超约简的方法体系中同时引入约简求解的加速策略以及鲁棒的约简求解机制。首先，就如何加速约简求解这一问题而言，近年来已有丰硕的研究成果[3,11-12]。例如，Chen 等[12]提出了一类基于属性簇的快速约简求解模式，这一模式以属性间的相似度为切入点，对属性进行分簇，进而利用属性簇压缩候选属性空间，减少候选属性评估次数，以达到降低约简求解的时间消耗这一目的。其次，关于如何提升约简及其后续分类结果的鲁棒性这一问题，近年来也受到了一些学者的重点关注。例如，Yang 等[5]在基于贪心的启发式搜索进程中提出了一类集成选择模式，其核心思想是使用一组而非单个属性重要度对候选属性进行综合评估，从而能够更合理地挑选出较为稳健的属性。综上所述，将属性簇和集成选择同时引入超约简方法中，可构造如图 1 所示的框架图。

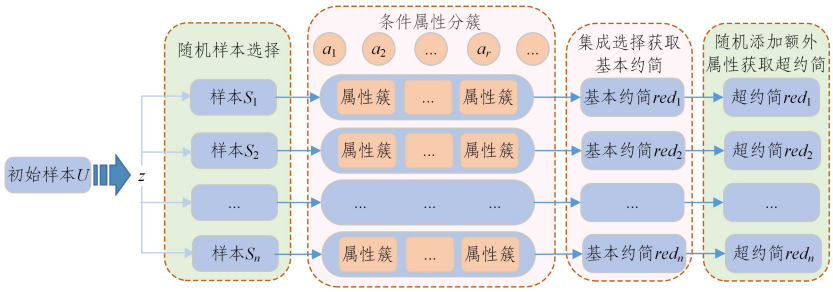

图 1 基于属性簇和集成选择的超约简求解

Fig. 1 Framework to search super-reducts with attribute group and ensemble selector

本文第 2 节简要介绍基于邻域粗糙集的属性约简；第 3 节分别介绍超约简求解、基于属性簇的快速约简求解及基于集成选择的稳定约简求解；第 4 节在超约简方法中同时引入属性簇和集成选择机制，以弥补现有超约简方法存在的不足；第 5 节进行实验分析；最后总结全文。

2 基础知识

2.1 邻域粗糙集

邻域粗糙集是为了弥补经典粗糙集无法处理数值型数据这一不足而提出的。在邻域粗糙集体系中，一个邻域决策系统可表示为三元组形如 $DS=\langle U,A,d\rangle$，其中，U 是一个非空且有限的样本集合，被称为论域；A 为条件属性集合；d 为决策属性。

给定一个邻域决策系统 DS，若考虑分类任务，则可令 $U/IND(d)=\{X_1,X_2,\cdots,X_q\}$ 表示根据决策属性 d 所诱导出的论域上的划分，其中 $IND(d)$ 是由决策属性 d 诱导出的一个等价关系形如 $IND(d)=\{(x,y)\in U\times U:d(x)=d(y)\}$，$\forall x\in U$，$d(x)$ 表示样本 x 的标签。因此，$\forall X_p\in U/IND(d)$，X_p 表示具有相同标签的样本所组成的第 p 个决策类。

定义 1 给定一个邻域决策系统 DS，$\forall B\subseteq A$，可以定义一个邻域关系[13]如式(1)所示：

$$N_B^\delta=\{(x,y)\in U\times U:\Delta_B(x,y)\leqslant\delta\} \tag{1}$$

其中，$\Delta_B(x,y)$ 表示依据条件属性集合 B 得到的样本 x 与样本 y 之间的距离，$\delta\geqslant 0$ 为一个给定的邻域半径。

根据上述邻域关系，$\forall x\in U$，不难得到样本 x 关于条件属性集合 B 的邻域形如 $\delta_B(x)=\{y\in U:(x,y)\in N_B^\delta\}$。

定义 2 给定一个邻域决策系统 DS 和邻域半径 δ，$\forall B\subseteq A$，决策属性 d 关于条件属性集合 B 的上、下近似集分别为：

$$\overline{N_B^\delta}(d)=\bigcup_{p=1}^{q}\overline{N_B^\delta}(Xp) \tag{2}$$

$$\underline{N_B^\delta}(d)=\bigcup_{p=1}^{q}\underline{N_B^\delta}(Xp) \tag{3}$$

其中 $\forall X_p\in U/IND(d)$，有：

$$\overline{N_B^\delta}(X_p)=\{x\in U:\delta_B(x)\cap X_p\neq\varnothing\} \tag{4}$$

$$\underline{N_B^\delta}(X_p)=\{x\in U:\delta_B(x)\subseteq X_p\} \tag{5}$$

定义 3 给定一个邻域决策系统 DS 和邻域半径 δ，$\forall B\subseteq A$，决策属性 d 关于条件属性集合 B 的近似质量定义为：

$$\gamma_U(B,d)=\frac{|\underline{N_B^\delta}(d)|}{|U|} \tag{6}$$

其中，$|\cdot|$ 表示集合的基数。

定义 3 所示的近似质量实际上描述了邻域决策系统中基于下近似所刻画的确定性程度，因此可以被视作是粗糙集方法中一种典型的度量计算方法。

2.2 属性约简

作为一种基于粗糙集的特征选择方法，属性约简已经在诸多领域[14-16]受到了重点关注。一般来说，属性约简的目的是获取能够满足给定约束条件的最小属性子集，以此来降低数据维度，提升模型的学习性能。以下将给出属性约简的形式化定义。

定义 4 给定一个邻域决策系统 DS，C_ρ 是与给定的度量 ρ 相关的约束，$\forall red\subseteq A$，red 被称为 DS 中的约简当且仅当：

(1) red 满足约束 C_ρ；

(2) $\forall red'\in red$，red' 不满足约束 C_ρ。

在定义 4 中，第一个条件表示 red 能够满足给定的约束条件 C_ρ，第二个条件确保 red 中没有冗余属性的存在。

在邻域粗糙集中，对于度量 ρ 而言，有多种计算方法，如定义 3 所示的近似质量，此外，还有条件熵[17-18]以及邻域决策错误率[6,19]等。

进一步地，在约简求解过程中，一般可以使用度量 ρ 对迭代过程中产生的候选属性进行重要度评估[20]：若所使用的度量 ρ 是正向的，则选择较大的度量值作为选择属性的标准，例如近似质量就属于这一类度量；若所使用的度量 ρ 是负向的，则选择较小的度量值作为选择属性的标准，例如条件熵、邻域决策错误率都属于这一度量。

近年来，依据不同的约简求解需求，已有学者提出了形式多样的算法，其中借鉴贪心思想的启发式搜索算法[12]因其时间消耗较少而被广泛应用。在这一搜索过程中，候选属性的重要度可被视作每次迭代进程中所使用的启发式信息。以近似质量这一度量为例，可定义如下所示的重要度函数。

定义 5 给定一个邻域决策系统 DS，$\forall B\subseteq A$，$\forall a\in A-B$，属性 a 相对于条件属性集合 B 的重要度为[4]：

$$SIG_U(a,B,d)=\gamma_U(B\cup\{a\},d)-\gamma_U(B,d) \tag{7}$$

其中，$SIG_U(a,B,d)$表示当属性 a 加入到属性集合 B 后，属性 a 的重要度的计算方法。若 $SIG_U(a,B,d)>0$，则表示属性 a 的加入使得近似质量得到了提升，此时属性 a 可以加入到约简集合中；若 $SIG_U(a,B,d)\leqslant 0$，则表示属性 a 的加入并不能提升近似质量，此时属性 a 不宜加入到约简集合中。

3 约简求解策略

3.1 超约简求解

一般来说，利用基于贪心的启发式搜索可以在决策系统中获取一个约简，但所求结果容易陷入局部最优，致使后续学习器性能下降。鉴于此，已有相关学者从集成的视角设计了如集成选择器[5-7]、基于多重约简[8-9]的集成分类等策略。值得注意的是，Jiang 等[8]以多模态扰动为切入点，提出了一类被称为 E_RSRR 的集成方法，其主要步骤如下：

(1)通过随机重采样策略将训练数据集的原始样本空间打乱，产生多组样本集；

(2)对任意一个样本集，采用贪心搜索这一启发式策略，求得一个基本约简 red；

(3)对于每个基本约简 red，通过向 red 中进一步随机添加若干属性，生成超约简；

(4)利用给定的分类算法，在每个超约简上生成基分类器；

(5)根据基分类器的投票结果求得最终的集成分类结果。

从上述过程不难看出，E_RSRR 需要依据多组随机重采样以求解多个约简，相较于在原始样本空间上仅求解一个约简而言，这一过程会显式地带来较大的时间消耗；其次，E_RSRR 方法存在较大的随机性，原因是不同样本子空间的获取是随机的，并且，在基本约简的基础上进一步生成超约简这一过程也存在着较大的随机性。

3.2 基于属性簇的快速约简求解

为了追求更加高效的约简求解，已有众多学者提出了不同结构的约简加速策略。一般来说，大多数加速方法可以被归为三大类，分别为基于属性的加速模式[11-12]、基于样本的加速模式[21-22]及基于粒度的加速模式[3,14]。例如，Chen 等[12]提出的一类基于属性簇的快速约简求解策略就属于基于属性的加速模式，其主要步骤如下：

(1)将条件属性划分为多个簇，令约简 $red=\varnothing$，集合 $P'=\varnothing$；

(2)在 $A-P'$中找出重要度最高的属性，并将其分别添加到 red 和 P'中；

(3)如果 red 满足给定的约束条件，则输出 red，算法终止，否则，转至步骤(4)；

(4)对于 red 中的每一个属性，找出与其在同一属性簇中的其他属性，并将这些其他属性加入到集合 P'中；

(5)若 $P'=A$，则令 $P'=red$，转至步骤(2)。

从上述步骤可知，属性簇方法从属性间的相似度出发，对条件属性分簇，从而将候选属性空间从 $A-red$ 缩小为 $A-P'$，进而减少了对候选属性的评估次数，最终能够减少约简求解的时间消耗。

3.3 基于集成选择的稳定约简求解

近年来，在属性约简的研究中，除了考虑约简的求解效率，约简的稳定性也受到了一些学者的高度关注。一般来说，稳定性可被视为约简结果伴随着数据扰动的变化程度，约简结果具有较高的稳定性意味着该约简具有较好的鲁棒性。例如，Yang 等[5]在基于贪心的启发式搜索进程中提出了一种集成选择模式，其主要步骤如下：

(1)令约简 $red=\varnothing$；

(2)令多重集合 $T=\varnothing$，对候选属性集合 A –red 中的任意一个属性，采用多种不同的属性重要度度量方法对其进行评估；

(3)对于每一种重要度度量方法，可以在 A –red 中挑选出一个重要度最高的属性并将其加入到多重集合 T 中；

(4)在多重集合 T 中，挑选出一个出现频次最高的属性，并将其加入到约简 red 中；

(5)重复步骤(2)－步骤(4)，直至约简 red 满足给定的约束条件。

从上述步骤可知，集成选择方法可以从多个属性重要度评估的视角，对候选属性进行多方位的评价，从而在每一轮迭代中挑选出一个适应性最强的属性。其中，多重属性重要度评估方法既可以是同质的，也可以是异质的[5]。同质表示这些重要度评估方法具有类似的结构，而异质则表示这些重要度评估方法的结构之间可能存在较大差异。

4 基于属性簇和集成选择的超约简求解

由 3.1 节可知，原始超约简方法的计算流程不仅耗时，而且在约简生成过程中存在着较多的随机因素。鉴于此，在超约简方法中引入加速和鲁棒的策略，有望进一步提升超约简的效率和性能。本节将属性簇和集成选择同时引入到超约简的求解过程中，首先在随机重采样得到的每一组样本集上，利用基于贪心的启发式策略进行搜索时，嵌入属性簇的基本思想，有望在基本约简的求解上减少时间消耗，进而减少求解所有超约简的时间消耗；其次在属性簇的基础上进一步嵌入集成选择的核心思想，有望提升每个超约简的稳定性，进而在最终进行集成分类时，产生更为稳健的分类结果。综上所述，所提方法的形式化算法如算法 1 所示。

算法 1 基于属性簇和集成选择的超约简求解

输入：邻域决策系统 DS=〈U，A，d〉，一组属性重要度度量函数 SIG_1，SIG_2，…，SIG_m

输出：一组超约简 red_1，red_2，…，red_n；For i=1：n

1. 针对 DS，利用随机重采样策略，得到样本集 S_i；
2. 令 $red_i=\varnothing$，$\gamma_U(red_i,d)=-\infty$；
3. 将条件属性 A 划分为多个属性簇；
4. 利用定义 3，计算 $\gamma_U(A,d)$；
5. While $\gamma_U(red_i,d)<\gamma_U(A,d)$

5.1. 令 $P=A-red_i$，$P'=\varnothing$；

5.2. For $\forall a \in A - red_i$

若属性 a 与 P′中的某个属性在同一个属性簇中，则 $P = A - red_i - \{a\}$；

End

5.3. 若 $P = \emptyset$，则转至 5.4，否则转至 5.5；

5.4. $P = A - red_i$，$P' = \emptyset$；

5.5. 令 $T = \emptyset$；

5.6. For j=1:m

$\forall a \in P$，计算重要度值 $SIG_j(a, red_i, d)$；

依据上述重要度值，选择一个重要度最大的属性 b；

$T = T \cup \{b\}$；

End

5.7. 在 T 中挑选出现频次最大的属性 c，$red_i \cup \{c\}$，$P' = P' \cup \{c\}$；

5.8. 计算 $\gamma_U(red_i, d)$；

End

6. 向 red_i 中随机添加若干其他属性，更新 red_i 为超约简；

End

输出一组超约简 $red_1, red_2, \cdots, red_n$.

上述算法的时间消耗主要在于求取每个基本约简时对于候选属性的评估次数。在所提算法中，当属性簇的数目为 $l(1<l<|A|)$ 时，假设 l 个簇分别包含属性的个数为 N_1，N_2，…，N_l，则 $N_1+N_2+\cdots+N_l=|A|$，在最坏的情况下，所提算法需要评估候选属性的次数为 $|A|+(|A|-N_1)+(|A|-N_1-N_2)+\cdots+1$。然而对于基于贪心的启发式搜索来说，需要评估候选属性的次数为 $|A|+(|A|-1)+(|A|-2)+\cdots+1$。显然，所提算法能够减少候选属性的评估次数，从而加快算法的搜索进程。

此外，在算法 1 的步骤 5.6 中，算法采用了一组而非单个重要度度量函数对由属性簇方法筛选出来的候选属性进行评估，其目的在于利用多重重要度的组合力量，挑选出一个最合适的属性加入到 *red* 中，以提升所求约简结果的鲁棒性。

5 实验分析

为了验证所提算法的有效性，本节将进行对比实验分析。所有算法均采用 Matlab R2017b 实现，实验平台的操作系统为 Windows 10，CPU 为 Intel © Core(TM) i5-7200U，内存为 16.00GB。选取了 20 组 UCI 数据集用于测试算法性能，数据集的具体描述如表 1 所列。

表 1 数据集描述

Table 1 Description of data sets

ID	数据集名称	样本数	属性数	决策类数	领域
1	Amphetamines Consumption	1 885	12	7	Medicine
2	Breast Cancer Wisconsin(Diagnostic)	569	30	4	Life
3	Breast Tissue	106	9	6	Life
4	Cardiotocography	2 126	21	2	Computer
5	Dermatology	366	34	5	Life
6	Diabetic Retinopathy Debrecen	1 151	19	2	Medicine
7	Ionosphere	351	34	2	Physical
8	Libras Movement	360	90	3	Astronomy
9	LSVT Voice Rehabilitation	126	256	2	Computer
10	Parkinson Multiple Sound Recording Data	1 208	26	2	Life
11	QSAR Biodegradation	1 055	41	2	Biology
12	Quality Assessment of Digital Colposcopies	287	62	2	Life
13	Sonar	208	60	2	Physical
14	SPECTF Heart	267	44	2	Life
15	Statlog(Image Segmentation)	2 310	19	7	Life
16	Steel Plates Faults	1 941	33	2	Physical
17	Synthetic Control Chart Time Series	600	60	6	Life
18	Ultrasonic Flowmeter Diagnostics-Meter D	180	43	4	Computer
19	Urban Land Cover	675	147	9	Physical
20	Waveform Database Generator	5 000	21	3	Computer

本节实验均采用 10 折交叉验证[16]的方法测试算法的性能，即将数据中的样本按照规模分为 10 等份，每次取其中的 9 份进行超约简求解，剩余的 1 份作为测试集，测试所求得的超约简的分类性能，分类器均采用 CART 和 KNN。

此外，所有实验都是基于邻域粗糙集模型实现的，其中，邻域半径设置为：0.02，0.04，…，0.40，共 20 个。回顾以往的研究不难发现，较大的邻域半径会使得约简求解过程中的约束条件过于宽松，进而导致所求得的约简长度过短，此时所求约简仅包含少量信息。鉴于此，本文基于多组相对较小的邻域半径进行了实验。在多组半径上的实验结果均充分验证了所提方法的有效性。其中，在 0.02～0.40 这组半径上，所提方法对多方面性能的改善效果最为明显。因此，为了更好地与其他方法进行对比分析，本文在实验中选取了 0.02～0.40 这组半径，并在这组半径下，求得进行约简求解所需时间消耗的均值。

同时，实验采用了两种不同的策略对训练数据进行随机重采样：1)在训练数据中，从属性层面进行随机重采样，每次取 60%的原始属性，将这一方法记为 E_AGES1；2)在训练数据中，从样本层面进行随机重采样，每次取 60%的原始样本，将这一方法记为 E_AGES2。

E_AGES1 和 E_AGES2 将与 4 种借鉴集成思想的属性约简方法进行对比分析，分别是：1)文献[8]提出的基于随机超约简和重采样的集成学习(E_RSRR)；2)文献[9]提出的 R_集成选择器(R_ES)；3)文献[23]提出的用于属性约简的数据引导的多粒度选择器(DMSAR)；4)文献[5]提出的用于属性约简的集成选择器(ESAR)。

此外，所提方法 E_AGES1 和 E_AGES2 这二者中，若有任何一种优于其他 4 种对比方法(E_RSRR，R_ES，DMSAR 和 ESAR)的实验结果，则在文中以粗体表示。

5.1 时间消耗对比

在本小节的实验中，将对比 E_AGES1，E_AGES2，E_RSRR，R_ES，DMSAR 和 ESAR 方法的时间消耗，具体实验结果如表 2—表 4 所列。

表 2 不同方法求解约简的时间消耗

Table 2 Time consumption of different approaches to derive reducts

(单位：s)

ID	E_AGES1	E_AGES2	E_RSRR	R_ES	DMSAR	ESAR
1	**3.2412**	8.9696	12.0462	490.0576	42.0701	4.4359
2	**0.3370**	1.3673	2.0011	53.2591	4.9907	0.5767
3	**0.0331**	0.1175	0.1579	0.3416	0.1138	0.0493
4	**2.1492**	18.4952	22.1034	838.1074	76.4262	8.1369
5	**0.0927**	1.5337	2.3482	16.5347	4.9377	0.5475
6	**2.4428**	4.4246	7.5690	262.2551	37.6081	2.5431
7	**0.1618**	0.7352	1.0408	12.3851	1.9878	0.2185
8	**0.2707**	14.2266	39.9173	360.4144	96.3595	10.6939
9	**0.0776**	6.2287	16.1404	23.4780	24.4076	2.3795
10	**1.3453**	5.1974	7.3367	308.7355	20.5977	2.3132
11	**1.1737**	10.5619	14.5904	417.3398	37.6583	5.4169
12	**0.1141**	0.8626	2.3740	3.8021	4.0248	0.4918
13	**0.1091**	0.7616	1.6063	9.3641	2.7839	0.2663
14	**0.1390**	1.0085	1.6046	18.3256	3.0311	0.2955
15	**2.2616**	17.6967	22.1824	855.9618	110.2972	8.4492
16	**3.5482**	13.9849	27.8029	758.7372	81.4166	10.3706
17	**0.1612**	3.8987	4.6252	73.4788	14.2295	1.2667
18	**0.1374**	2.0005	2.6074	7.3178	3.5871	0.4593
19	**1.7835**	40.3986	61.6517	594.5454	240.5120	21.3761
20	18.8884	26.1483	47.2090	11960.0000	127.2434	14.5341
均值	**1.9234**	8.9309	14.8457	853.2221	46.7142	4.7411

表 3 E_AGES1 与其他方法的加速比

Table 3 Speed-up ratios of E_AGSE1 and other comparison approaches

ID	E_AGES1&E_RSRR	E_AGES1&R_ES	E_AGES1&DMSAR	E_AGES1&ESAR
1	3.7166	**151.1963**	12.9798	1.3686
2	5.9380	**158.0389**	14.8092	1.7113
3	4.7704	**10.3202**	3.4381	1.4894
4	10.2845	**389.9625**	35.5603	3.7860
5	25.3312	**178.3679**	53.2654	5.9061
6	3.0985	**107.3584**	15.3955	1.0411
7	6.4326	**76.5457**	12.2855	1.3504
8	147.4595	**1331.4163**	355.9642	39.5046
9	207.9948	302.5515	**314.5309**	30.6637
10	5.4536	**229.4919**	15.3109	1.7195
11	12.4311	**355.5762**	32.0851	4.6152
12	20.8063	33.3225	**35.2743**	4.3103
13	14.7232	**85.8304**	25.5170	2.4409
14	11.5439	**131.8388**	21.8065	2.1259
15	9.8083	**378.4762**	48.7695	3.7359
16	7.8358	**213.8372**	22.9459	2.9228
17	28.6923	**455.8238**	88.2723	7.8579
18	18.9767	**53.2591**	26.1070	3.3428
19	34.5678	**333.3588**	134.8539	11.9855
20	2.4994	**633.1929**	6.7366	0.7695
均值	29.1182	**280.4883**	63.7954	6.6324

表 4 E_AGES2 与其他方法的加速比

Table 4 Speed-up ratios of E_AGSE2 and other comparison approaches

ID	E_AGES2&E_RSRR	E_AGES2&R_ES	E_AGES2&DMSAR	E_AGES1&ESAR
1	1.3430	**54.6354**	4.6903	0.4945
2	1.4635	**38.9520**	3.6500	0.4218
3	1.3438	**2.9072**	0.9685	0.4196
4	1.1951	**45.3149**	4.1322	0.4399
5	1.5311	**10.7809**	3.2195	0.3570
6	1.7107	**59.2720**	8.4998	0.5748
7	1.4157	**16.8459**	2.7038	0.2972
8	2.8058	**25.3338**	6.7732	0.7517
9	2.5913	3.7693	**3.9186**	0.3820
10	1.4116	**59.4019**	3.9631	0.4451
11	1.3814	**39.5137**	3.5655	0.5129
12	2.7521	4.4077	**4.6659**	0.5701
13	2.1091	**12.2953**	3.6553	0.3497
14	1.5911	**18.1711**	3.0056	0.2930
15	1.2535	**48.3684**	6.2326	0.4774
16	1.9881	**54.2540**	5.8218	0.7416
17	1.1863	**18.8470**	3.6498	0.3249
18	1.3034	**3.6580**	1.7931	0.2296
19	1.5261	**14.7170**	5.9535	0.5291
20	1.8054	**457.3911**	4.8662	0.5558
均值	1.6854	**49.4418**	4.2864	0.4584

观察表 2 不难发现，对比前 5 种方法，采用 E_AGES1 和 E_AGES2，求解约简所需的时间消耗要远远低于其他 3 种方法，这主要是因为：1)在算法流程中，随机重采样可以减少数据的规模，因此可以减少一定的时间消耗；2)所提算法中引入了属性簇这一加速机制，也为减少时间消耗提供了一定的帮助。此外，E_AGES1 相较于 E_AGES2 方法来说，时间消耗也有了显著减少，这主要是因为 E_AGES1 方法是在属性层面上进行随机重采样，因此属性个数减少了，致使约简耗时进一步减少。

在所测试的 6 种方法中，E_AGES2 相较于 ESAR 方法耗时较长，原因是在整个算法流程中，ESAR 方法只求解一个约简，而 E_AGES2 需求解多个约简。此外，R_ES 方法耗时最长，主要是因为 R_ES 方法分别求解了 3 种不同形式的约简，而其中无监督约简[24]求解耗时较长。

根据表 3 和表 4 所列的加速比结果也可以很清晰地看出，所提方法能够大幅度提升约简求解效率。以数据集 Steel Plates Faults 为例，E_AGES1 与其他 4 种方法的加速比分别是 7.8358，213.8372，22.9459 和 2.9228；E_AGES2 与其他 4 种方法的加速比分别是 1.9881，54.2540，5.8218 和 0.7416。证明了所提方法在时间效率上的优越性。

5.2 分类稳定性对比

本节将对比 E_AGES1，E_AGES2，E_RSRR，R_ES，DMSAR 和 ESAR 方法所求得的约简在测试集上得到的分类结果的稳定性的均值，其中，分类稳定性的计算采用文献[5]中的方法。具体实验结果如表 5 和表 6 所列。观察表 5 和表 6 不难发现，无论是采用 CART 还是 KNN，利用 E_AGES1 和 E_AGES2 方法所求得的约简都能得到较好的分类结果稳定性。相较于 R_ES，DMSAR 和 ESAR 来说，这种优势尤为明显。这主要是因为在所提算法的流程中引入了集成选择机制，而集成选择机制可以帮助求得较为稳定的约简，因此会

带来较为稳定的分类结果。以数据集 Synthetic Control Chart Time Series 为例,E_AGES1,E_AGES2 与其他 4 种方法的 CART 分类结果的稳定性分别是 0.9429,0.9298,0.9263,0.7223,0.7361 和 0.7382;E_AGES1,E_AGES2 与其他 4 种方法的 KNN 分类结果的稳定性分别是 0.9657,0.9571,0.9458,0.7547,0.8339 和 0.8232。由此可见,本文所提方法在分类稳定性上的确具有较大的优越性。

表 5 CART 分类结果的稳定性

Table 5 Classification stability based on CART classifier

ID	E_AGES1	E_AGES2	E_RSRR	R_ES	DMSAR	ESAR
1	**0.8373**	0.7268	0.7220	0.7062	0.7133	0.7089
2	**0.9514**	0.9439	0.9418	0.9306	0.9169	0.9155
3	**0.8786**	0.8567	0.8590	0.8519	0.8367	0.8462
4	0.8600	**0.8759**	0.8742	0.7864	0.8530	0.8437
5	0.9316	0.9715	0.9758	0.7008	0.9108	0.8805
6	**0.7639**	0.6780	0.6846	0.6377	0.6563	0.6442
7	0.9536	0.9487	0.9524	0.8684	0.9543	0.9220
8	**0.7775**	0.7624	0.7369	0.7108	0.7006	0.7082
9	0.7927	0.7888	0.8027	0.7015	0.7288	0.7196
10	**0.7589**	0.6622	0.6674	0.6405	0.6354	0.6394
11	**0.9124**	0.8661	0.8513	0.8039	0.8100	0.8001
12	**0.9041**	0.9007	0.9026	0.7212	0.7226	0.7448
13	0.7717	**0.7729**	0.7469	0.5869	0.6629	0.6483
14	**0.8774**	0.8489	0.8475	0.7458	0.7536	0.7500
15	0.9557	0.9587	0.9598	0.9546	0.9526	0.9496
16	0.8949	**0.9417**	0.8832	0.8794	0.7828	0.7789
17	**0.9429**	0.9298	0.9263	0.7223	0.7361	0.7382
18	**0.8792**	0.8656	0.8625	0.7808	0.8161	0.7853
19	0.8491	**0.8919**	0.8844	0.6913	0.8316	0.8167
20	**0.8207**	0.8046	0.7766	0.6913	0.7133	0.7080
均值	**0.8657**	0.8498	0.8429	0.7556	0.7844	0.7774

表 6 KNN 分类结果的稳定性

Table 6 Classification stability based on KNN classifier

ID	E_AGES1	E_AGES2	E_RSRR	R_ES	DMSAR	ESAR
1	0.8150	0.8284	0.8277	0.7950	0.8342	0.8316
2	0.9817	**0.9888**	0.9870	0.9753	0.9754	0.9758
3	0.7848	0.8067	0.7986	0.8157	0.8100	0.8076
4	0.8415	0.8736	0.8714	0.8007	0.8817	0.8778
5	0.9362	0.9839	0.9920	0.6289	0.9584	0.9076
6	**0.8110**	0.7935	0.7957	0.7141	0.7917	0.7866
7	0.9239	0.9360	0.9430	0.8215	0.9554	0.9100
8	0.8460	0.8513	0.8660	0.7143	0.8311	0.8221
9	0.8627	0.8869	0.8885	0.7681	0.8065	0.8131
10	0.8387	0.8398	0.8359	0.7954	0.8428	0.8283
11	**0.9386**	0.9278	0.9212	0.8468	0.9152	0.9082
12	**0.9450**	0.9431	0.9367	0.8224	0.8717	0.8764
13	0.8755	0.9010	0.9176	0.6512	0.8736	0.8488
14	0.8604	**0.8617**	0.8600	0.7991	0.8555	0.8408
15	0.9510	0.9756	0.9757	0.9745	0.9738	0.9736
16	0.9065	**0.9346**	0.9014	0.9177	0.9068	0.8815
17	**0.9657**	0.9571	0.9458	0.7547	0.8339	0.8232
18	**0.9089**	0.9075	0.9033	0.7800	0.8572	0.8197
19	**0.8898**	0.8799	0.8830	0.7021	0.8876	0.8677
20	0.8519	**0.8831**	0.8824	0.7162	0.8764	0.8608
均值	0.8867	**0.8980**	0.8966	0.7897	0.8769	0.8631

5.3 分类准确率对比

在本小节的实验中,将对比利用 E_AGES1,E_AGES2,E_RSRR,R_ES,DMSAR 和 ESAR 方法所求得约简后,这些约简在测试集上得到的分类结果的准确率的均值。实验结果如表 7 和表 8 所列。

表 7 CART 分类结果的准确率

Table 7 Classification accuracy based on CART classifier

ID	E_AGES1	E_AGES2	E_RSRR	R_ES	DMSAR	ESAR
1	**0.4992**	0.4382	0.4323	0.4242	0.4243	0.4229
2	**0.9210**	0.9138	0.9172	0.9059	0.9025	0.9038
3	0.7333	**0.7676**	0.7590	0.7581	0.7648	0.7524
4	0.8286	**0.8393**	0.8372	0.7227	0.8177	0.8120
5	0.9478	**0.9774**	0.9766	0.6303	0.9097	0.8870
6	**0.6217**	0.5931	0.5962	0.6094	0.5842	0.6003
7	0.9163	**0.9170**	0.9157	0.8554	0.9160	0.8980
8	**0.6243**	0.6075	0.5944	0.2596	0.4646	0.4642
9	0.7819	**0.7827**	0.7742	0.7335	0.7131	0.7115
10	**0.6947**	0.6516	0.6546	0.6386	0.6392	0.6325
11	**0.8342**	0.8132	0.8115	0.7877	0.7852	0.7821
12	0.8140	**0.8166**	0.8162	0.7131	0.7240	0.7333
13	**0.8052**	0.7993	0.7807	0.6343	0.6726	0.6510
14	**0.8066**	0.7928	0.7974	0.7413	0.7196	0.7215
15	0.9518	0.9592	0.9593	0.9547	0.9518	0.9516
16	0.9049	**0.9514**	0.8819	0.6745	0.7865	0.7801
17	**0.9602**	0.9448	0.9340	0.6171	0.5264	0.5258
18	**0.8886**	0.8808	0.8758	0.8119	0.7992	0.8097
19	0.7331	**0.8308**	0.8259	0.6073	0.7813	0.7783
20	**0.8681**	0.7951	0.7771	0.5039	0.7024	0.7011
均值	**0.8068**	0.8036	0.7959	0.6792	0.7293	0.7260

表 8 KNN 分类结果的准确率

Table 8 Classification accuracy based on KNN classifier

ID	E_AGES1	E_AGES2	E_RSRR	R_ES	DMSAR	ESAR
1	**0.4956**	0.4693	0.4670	0.4459	0.4637	0.4571
2	**0.9477**	0.9464	0.9436	0.9409	0.9249	0.9251
3	**0.5990**	0.5890	0.5867	0.5952	0.5862	0.5805
4	0.7692	0.7810	0.7837	0.6864	0.7835	0.7809
5	0.9350	**0.9607**	0.9543	0.5674	0.9314	0.8746
6	0.6244	0.5850	0.5885	0.6287	0.5861	0.5910
7	0.8401	0.8469	0.8474	0.8234	0.8643	0.8477
8	0.6879	0.6958	0.7047	0.2699	0.6082	0.5999
9	0.8681	**0.8735**	0.8723	0.7846	0.7008	0.7008
10	**0.7000**	0.6955	0.6952	0.6873	0.6841	0.6854
11	**0.8318**	0.8185	0.8170	0.7910	0.8117	0.8039
12	0.7914	**0.7922**	0.7903	0.7433	0.7067	0.7034
13	**0.8769**	0.8690	0.8731	0.6979	0.7521	0.7471
14	**0.7640**	0.7334	0.7223	0.7043	0.6977	0.7008
15	0.9425	0.9448	0.9458	0.9463	0.9363	0.9418
16	0.8842	**0.9240**	0.8370	0.6747	0.7895	0.7838
17	**0.9606**	0.9488	0.9443	0.5428	0.4615	0.4632
18	**0.8581**	0.8303	0.8175	0.7931	0.7903	0.7864
19	0.7153	0.7999	0.8070	0.5770	0.7877	0.7689
20	**0.8351**	0.8316	0.8192	0.5172	0.7489	0.7455
均值	0.7963	**0.7968**	0.7908	0.6709	0.7308	0.7244

观察表 7 和表 8 不难发现,无论是采用 CART 还是 KNN,利用 E_AGES1 和 E_AGES2 方法所求得的约简都能得到较高的分类准确率。相较于 R_ES,DMSAR 和 ESAR 来说,这种优势尤为明显,这一结论与表 5 和表 6 所列的分类结果的稳定性一致。以数据集 Waveform Database Generator 为例,E_AGES1,E_AGES2 与其他 4 种方法的 CART 分类结果的准确率分别是 0.8681,0.7951,0.7771,0.5039,0.7024 和 0.7011;E_AGES1,E_AGES2 与其他 4 种方法的 KNN 分类结果的准确率分别是 0.8351,0.8316,0.8192,0.5172,0.7489 和 0.7455。由此可见,所提方法在分类准确率上的确具有较大的优越性。

综合以上实验分析,可以得出如下结论:

(1)由表 2—表 4 可知,在约简的求解过程中,引入属性簇机制能够有效地降低超约简求解所需的时间消耗。

(2)由表 5—表 8 可知,引入集成选择机制能够使得求解得到的超约简具有更好的泛化性能。

结束语 与传统约简求解方式不同的是，超约简求解框架可以为约简问题的探索带来更多的灵活性，有利于借鉴集成的思想，提升后续学习器的性能。然而，由于原始超约简方法存在着耗时较长、分类性能有待进一步提升等实际问题，因此本文在其框架中进一步引入了属性簇和集成选择策略，分别用于加快约简的求解速度以及选择出更为稳健的属性。在此基础上，今后将就以下问题做进一步研究。

(1)本文仅从属性层面对超约简框架中的效率问题进行了探索，未来可以进一步从样本和粒度等其他层面进行探索。

(2)如何构造形式多样的重要度度量函数是集成选择的核心，本文仅从同构的视角，使用了文献[5]所提的局部策略构造重要度度量函数，未来可以考虑从异构的视角构造差异性更大的度量函数。

参 考 文 献

[1] PAWLAK Z. Rough Sets: Theoretical Aspects of Reasoning about Data[M]. Dordrecht: Kluwer Academic Publishers, 1992.

[2] SONG J J, TSANG E C C, CHEN D G, et al. Minimal Decision Cost Reduct in Fuzzy Decision-theoretic Rough Set Model[J]. Knowledge-Based Systems, 2017, 126: 104-112.

[3] BA J, CHEN Y, YANG X B. Attribute Partition Strategy for Quick Searching Reducts Based on Granular Ball Rough Sets [J]. Journal of Nanjing University of Science and Technology, 2021, 45(4): 394-400.

[4] LI J Z, YANG X B, DOU H L, et al. Research on Ensemble Significance Based Attribute Reduction Approach[J]. CAAI Transactions on Intelligent Systems, 2018, 13(3): 414-421.

[5] YANG X B, YAO Y Y. Ensemble Selector for Attribute Reduction[J]. Applied Soft Computing, 2018, 70: 1-11.

[6] LIU K Y, YANG X B, YU H L, et al. Rough Set Based Semi-supervised Feature Selection via Ensemble Selector [J]. Knowledge-Based Systems, 2019, 165: 282-296.

[7] JIANG Z H, WANG Y B, XU G, et al. Multi-scale Based Accelerator for Attribute Reduction [J]. Computer Science, 2019, 46(12): 250-256.

[8] JIANG F, YU X, ZHAO H B, et al. Ensemble Learning Based on Random Super-reduct and Resampling[J]. Artificial Intelligence Review, 2021, 54(2): 1-26.

[9] BANIA R K, HALDER A. R-Ensembler: A Greedy Rough Set Based Ensemble Attribute Selection Algorithm with KNN Imputation for Classification of Medical Data [J]. Computer Methods and Programs in Biomedicine, 2019, 184(4): 105122.

[10] NIU K, ZHANG Z M, LIU Y, et al. Resampling Ensemble Model Based on Data Distribution for Imbalanced Credit Risk Evaluation in P2P Lending[J]. Information Sciences, 2020, 536: 120-134.

[11] QIAN Y H, WANG Q, CHENG H H, et al. Fuzzy-rough Feature Selection Accelerator[J]. Fuzzy Sets and Systems, 2015, 258: 61-78.

[12] CHEN Y, LIU K Y, SONG J J, et al. Attribute Group for Attribute Reduction[J]. Information Sciences, 2020, 535: 64-80.

[13] MENG J, ZHANG J, JIANG D L, et al. Selective Ensemble Classification Integrated with Affinity Propagation Clustering [J]. Journal of Computer Research and Development, 2018, 55(5): 986-993.

[14] JIA X Y, SHANG L, ZHOU B, et al. Generalized Attribute Reduct in Rough Set Theory[J]. Knowledge-Based Systems, 2016, 91: 204-218.

[15] LI W W, JIA X Y, WANG L, et al. Multi-objective Attribute Reduction in Three-way Decision-theoretic Rough Set Model [J]. International Journal of Approximate Reasoning, 2019, 105: 327-341.

[16] SINGH U, SINGH S N. A New Optimal Feature Selection Scheme for Classification of Power Quality Disturbances Based on Ant Colony Framework[J]. Applied Soft Computing, 2019, 74: 216-225.

[17] ROMAGNOLI P P. Local Conditional Entropy in Measure for Covers with Respect to a Fixed Partition [J]. Nonlinearity, 2018, 31(5): 2201-2220.

[18] JIANG Z H, LIU K Y, YANG X B, et al. Accelerator for Supervised Neighborhood Based Attribute Reduction[J]. International Journal of Approximate Reasoning, 2020, 119: 122-150.

[19] YANG X B, LIANG S C, YU H L. Pseudo-label Neighborhood Rough Set: Measures and Attribute Reductions[J]. International Journal of Approximate Reasoning, 2019, 105: 112-129.

[20] BA J, LIU K Y, JU H R, et al. Triple-G: A New MGRS and Attribute Reduction[J]. International Journal of Machine Learning and Cybernetics, 2022, 13(2): 337-356.

[21] YANG X B, YAN X, XU S P, et al. New Heuristic Attribute Reduction Algorithm Based on Sample Selection[J]. Computer Science, 2016, 43(1): 40-43.

[22] WANG N, PENG Z H, CUI L. EasiFFRA: A Fast Feature Reduction Algorithm Based on Neighborhood Rough Set[J]. Journal of Computer Research and Development, 2019, 56(12): 2578-2588.

[23] JIANG Z H, DOU H L, SONG J J, et al. Data-guided Multigranularity Selector for Attribute Reduction[J]. Applied Intelligence, 2021, 51: 876-888.

[24] YUAN Z, CHEN H M, LI T R, et al. Unsupervised Attribute Reduction for Mixed Data Based on Fuzzy Rough Sets[J]. Information Sciences, 2021, 572: 67-87.

WANG Xiaoxiao, born in 1996, postgraduate. Her main research interests include rough set and granular computing.

SONG Jingjing, born in 1990, Ph.D, associate professor. Her main research interests include rough set, granular computing and machine learning.

(责任编辑：何杨)