计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(7):210−216 [doi: 10.15888/j.cnki.csa.008605]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于近似约简与最优采样的集成剪枝①

王安琪, 江　峰, 张友强, 杜军威

(青岛科技大学 信息科学技术学院, 青岛 266061)
通信作者: 江　峰, E-mail: jiangfeng@qust.edu.cn

摘　要: 集成学习被广泛用于提高分类精度, 近年来的研究表明, 通过多模态扰乱策略来构建集成分类器可以进一步提高分类性能. 本文提出了一种基于近似约简与最优采样的集成剪枝算法 (EPA_AO). 在 EPA_AO 中, 我们设计了一种多模态扰乱策略来构建不同的个体分类器. 该扰乱策略可以同时扰乱属性空间和训练集, 从而增加了个体分类器的多样性. 我们利用证据 KNN (K-近邻) 算法来训练个体分类器, 并在多个 UCI 数据集上比较了 EPA_AO 与现有同类型算法的性能. 实验结果表明, EPA_AO 是一种有效的集成学习方法.

关键词: 集成剪枝; 多模态扰乱; 近似约简; 最优采样; 粗糙集; 属性约简; 数据挖掘

引用格式: 王安琪,江峰,张友强,杜军威.基于近似约简与最优采样的集成剪枝.计算机系统应用,2022,31(7):210–216. http://www.c-s-a.org.cn/1003-3254/8605.html

Ensemble Pruning Based on Approximate Reducts and Optimal Sampling

WANG An-Qi, JIANG Feng, ZHANG You-Qiang, DU Jun-Wei

(College of Information Science & Technology, Qingdao University of Science and Technology, Qingdao 266061, China)

Abstract: Ensemble learning has been widely used for improving classification accuracy. Recent studies show that building ensemble classifiers through a multi-modal perturbation strategy can further improve classification performance. In this study, we propose an ensemble pruning algorithm based on approximate reducts and optimal sampling (EPA_AO). In EPA_AO, we design the multi-modal perturbation strategy to build different individual classifiers. The proposed perturbation strategy can simultaneously perturb the attribute space and training set, which can improve the diversity of individual classifiers. We use the evidential K-nearest neighbor (KNN) algorithm to train individual classifiers and compare EPA_AO with existing algorithms of the same type on multiple UCI data sets. Experimental results show that EPA_AO is an effective ensemble learning approach.

Key words: ensemble pruning; multi-modal perturbation; approximate reducts; optimal sampling; rough sets; attribute reduction; data mining

集成学习近年来得到广泛研究[1−5]. 为了构建一个有效的集成分类器, 研究者提出了不同的方法来训练一组好而不同的个体分类器. 现有的集成方法大致可分为两类, 即基于单模态扰乱的方法和基于多模态扰乱的方法, 其中, 第 1 类方法又包括基于重采样的方法 (如 Bagging、Boosting 等)[6,7]、基于特征子空间的方法[8−10] 等. 与第 1 类方法不同, 基于多模态扰乱的方法[5,11,12] 在训练个体分类器的过程中使用了多种扰乱技术, 其动机是: 很多时候单一类型的扰乱可能不足以产生多样性大的个体分类器. 早期的集成方法通常对所有的个体分类器进行集成, 然而, 这种做法并不一定能够保证个体分类器之间的多样性. 对此, 研究者们进

① 基金项目: 国家自然科学基金 (61973180, 61671261); 山东省自然科学基金 (ZR2021MF092, ZR2018MF007)
收稿时间: 2021-10-18; 修改时间: 2021-11-17; 采用时间: 2021-12-13; csa 在线出版时间: 2022-03-31

一步提出了不同的集成剪枝方法[4,13,14]. 集成剪枝的主要目的是通过删除一些个体分类器来提升个体分类器之间的多样性 (即只利用一部分多样性大的个体分类器来构建集成分类器). 研究表明, 集成剪枝能够有效提升集成学习的性能.

为了增加个体分类器的多样性, 多模态扰乱策略被广泛应用于集成学习, 并形成了不同的基于多模态扰乱的集成方法. 例如, Lattine 等提出一种用于集成 C4.5 决策树的多模态扰乱策略[15], 该策略将 Bagging 和随机子空间 (RSM) 方法结合起来. 与基于单模态扰乱的方法相比, Lattine 等的方法具有更好的性能. Altinçay 提出一种基于多模态扰乱的集成方法, 该方法使用遗传算法 (GA) 来选择特征子集和最优重采样[16]. 实验结果表明: Altinçay 的方法要优于基于单模态扰乱的 RSM 方法. Zhou 等[11] 提出一种集成 KNN 分类器的多模态集成算法, 该算法对训练数据、特征空间和学习参数同时进行扰乱. 与每一种基于单模态扰乱的方法相比, Zhou 等的方法都可以取得更好的性能.

本文从粗糙集[17] 中的属性约简技术出发, 提出了一种新的可同时扰乱属性空间和训练集的集成剪枝算法 EPA_AO. EPA_AO 的主要思路如下: (1) 假设 T 和 V 分别表示训练集和验证集, 我们通过近似约简技术从 T 的属性集中产生 M 个近似约简 AR_1, …, AR_M. (2) 针对任意 AR_i ($1\leqslant i\leqslant M$), 对 T 中的样本进行 H 次 bootstrap 采样, 获得 H 个采样集 S_1, …, S_H 并从中选择关于 AR_i 的最优采样集 O_i. 具体而言, 对于任意采样集 S_j, 我们利用 AR_i 来对 S_j 进行约简, 再利用分类算法在 S_j 上训练个体分类器 C_j ($1\leqslant j\leqslant H$). 然后, 利用 V 来分别评价个体分类器 C_1, …, C_H 的性能, 并将性能最好的个体分类器所对应的采样集作为 AR_i 的最优采样集 O_i. (3) 利用 AR_i 以及 AR_i 所对应的最优采样集 O_i 来训练个体分类器 IC_i ($1\leqslant i\leqslant M$). (4) 基于多数投票的策略, 将个体分类器 IC_1, …, IC_M 集成在一起, 从而得到集成分类器.

本文的工作主要包括 3 个部分: 首先, 在粗糙集中引入近似约简的概念, 并提出一种计算近似约简的算法; 其次, 设计出一种基于近似约简与最优采样的多模态扰乱策略: 通过近似约简来扰乱属性空间, 并通过最优采样技术来扰乱训练集; 最后, 基于上述多模态扰乱策略, 提出一种集成剪枝算法. 该算法通过计算近似约简及其对应的最优采样集, 可以在保证个体分类器性能的前提下, 有效提升个体分类器的多样性. 证据 KNN (K-近邻) 分类算法[18] 简单、有效, 并且在许多领域都得到了广泛应用. 相对于传统的 KNN 算法, 证据 KNN 算法通常具有更好的性能. 因此, 在实验中我们使用证据 KNN 算法来训练个体分类器. 我们利用多个 UCI 数据集来比较 EPA_AO 与现有的集成学习算法的性能. 实验结果表明, EPA_AO 具有更好的分类性能.

1 预备知识

在粗糙集理论中, 信息系统是一个四元组 $IS=(U, A, V, f)$, 其中, U 是一个非空、有限的对象集; A 是一个非空、有限的属性集; $V=\cup_{a\in A}V_a$ 是所有属性论域的并集 (V_a 表示属性 a 的值域); f: $U\times A\rightarrow V$ 是一个信息函数, 使得对任意 $a\in A$, $u\in U$, $f(u, a)\in V_a$ [17,19]. 如果进一步把属性集 A 划分为条件属性集 C 和决策属性集 D, 则这种特殊的信息系统被称为决策表 $DT=(U, C, D, V, f)$.

定义 1. 不可分辨关系. 给定决策表 $DT=(U, C, D, V, f)$, 对任意 $B\subseteq C\cup D$, 我们将由 B 所确定的不可分辨关系 $IND(B)$ 定义为[17,19]:

$$IND(B)=\{(u_1,u_2)\in U\times U:\forall a\in B(f(u_1,a)=f(u_2,a))\} \tag{1}$$

不可分辨关系 $IND(B)$ 实际上是 U 上的一个等价关系, 它将 U 划分为多个不相交的等价类. 令 $U/IND(B)$ 表示 $IND(B)$ 所对应的所有等价类的簇, 我们称 $U/IND(B)$ 为由 $IND(B)$ 所确定的论域 U 的划分. 对任意 $u\in U$, 令 $[u]_B$ 表示 $U/IND(B)$ 中包含对象 u 的等价类[17,19].

定义 2. 正区域. 给定决策表 $DT=(U, C, D, V, f)$, 对任意 $B\subseteq C$, 我们将 D 的 B-正区域 $POS_B(D)$ 定义为[17,19]:

$$POS_B(D)=\cup\{Y|Y\subseteq X, X\in U/IND(D), Y\in U/IND(B)\} \tag{2}$$

定义 3. 核属性. 给定决策表 $DT=(U, C, D, V, f)$, 对任意 $b\in C$, 如果 $POS_{C-\{b\}}(D)\neq POS_C(D)$, 则我们称 b 是 C 中相对于 D 的一个核属性[17,19].

给定决策表 $DT=(U, C, D, V, f)$, C 中所有核属性的集合被称为 C 相对于 D 的核.

定义 4. 约简. 给出决策表 $DT=(U, C, D, V, f)$, 对任意 $B\subseteq C$, 如果 $POS_B(D)=POS_C(D)$ 并且对任意 $b\in B$, $POS_{B-\{b\}}(D)\neq POS_B(D)$, 则我们称 B 为 C 相对于 D 的一个约简[17,19].

接下来, 我们分析证据 KNN 的原理. 证据 KNN

是由 Denoeux 所提出的一种分类算法[18], 它将训练样本在输入空间不同部分的分布信息融入到传统的 KNN 中, 其主要思想是: 从所有训练样本中计算出测试样本 t 的 k 个最近邻, 将每个最近邻都作为支持 t 所属类的证据, 然后结合所有最近邻的基本概率赋值来得到 t 的类别[18].

令 $L=\{l_1, \cdots, l_h\}$表示由 h 个类标签所组成的标签集, $T=\{(u_1, C(u_1)), \cdots, (u_g, C(u_g))\}$表示由 g 个样本所组成的训练样本集, T 中每个样本都用一个二元组 $(u_j, C(u_j))$ 来表示, 其中, u_j 表示第 j 个样本, $C(u_j)\in L$ 表示 u_j 所属的类别标签, $1\leqslant j \leqslant g$. 令 $N_w=\{(v_1, C(v_1)), \cdots, (v_k, C(v_k))\}\subset T$ 表示 T 中距离当前的测试样本 t_w 最近的 k 个训练样本 (即 t_w 的 k-最近邻), N_w 中每个元素也用一个二元组 $(v_j, C(v_j))$ 来表示, 其中, v_j 表示 t_w 的第 j 个最近邻, $C(v_j)\in L$ 表示 v_j 所属的类别标签, $1\leqslant j \leqslant k$. 对于 t_w 的任意一个最近邻 $v_j(1\leqslant j \leqslant k)$, 如果 $C(v_j)=l_p\in L$, 则我们将 (v_j, l_p) 看作是支持 t_w 被分类为 l_p 的一个独立证据.

每一个独立证据 (v_j, l_p) 的可信任度都由一个基本概率赋值函数来度量, 即该基本概率赋值函数的值越大, 则证据 (v_j, l_p) 的可信任度越高. 具体而言, (v_j, l_p) 中所包含的分布信息将采用以下的基本概率赋值函数来刻画: $E^{w,j}(\{l_p\})=\beta$, $E^{w,j}(L)=1-\beta$. 在上述基本概率赋值函数中, $\beta\in[0, 1]$, β 的具体取值由 v_j 与 t_w 的距离 $d(v_j, t_w)$ 所决定 (通常情况下, 距离越大, 则 β 越小). 很多学者通过定义相似函数来描述 β 与距离 $d(v_j, t_w)$ 的关系, 例如 Denoeux 将相似函数定义成[18]: $\beta=\beta_0\times e^{-\gamma s\times d(vj,tw)^2}$. 在上面的相似函数中, $0<\beta_0<1$ 是一个预先指定的参数, 而 $\gamma_s>0$ 则是另一个预先指定的参数. 对任意 v_i, $v_j\in N_w$, 如果 $i\neq j$, 则 $E^{w,i}$ 与 $E^{w,j}$ 是相互独立的, 这是因为这两个基本概率赋值函数是由不同的训练样本所生成的. 利用证据理论的组合规则, 将 $E^{w,1}$, $\cdots$, $E^{w,k}$ 都组合起来, 从而获得 $E^w=\oplus_{vi\in Nw}E^{w,i}$. 在获得 E^w 之后, 就可以利用其来计算 t_w 相对于各个类标签的置信度函数, 从而实现对测试样本 t_w 的分类.

2 近似约简与 EPA_AO 算法

在粗糙集中, 给定决策表 $DT=(U, C, D, V, f)$, 一个约简 R 是条件属性集 C 的最小子集, 它能充分识别具有不同类别的对象. 理论上已经证明, 基于约简 R 所训练的分类器与基于 C 所训练的分类器具有相等的分类能力[20]. 一些学者提出在 C 的每一个约简上训练一个个体分类器, 并利用这些个体分类器来构建集成分类器. 由于 C 的约简与 C 具有相同的区分能力, 因此, 基于这些约简所构建的个体分类器不仅可以保证个体分类器的分类能力, 还可以保证每个个体分类器之间具有较好的差异性, 降低个体分类器训练的复杂度, 从而高效率地训练出性能优异的分类器. 也就是说, 我们可以利用粗糙集的属性约简技术来扰乱属性空间. 然而, 利用属性约简技术对属性空间进行扰乱时将面临以下问题: 约简个数过少. 很多时候, 在决策表 DT 中, 只存在少量的 C 的约简. 如果没有足够多的约简, 那么我们就不能训练出足够多的个体分类器. 特别是, 如果只存在一个约简, 那么我们就只能获得一个个体分类器. 为解决上述问题, 本文提出近似约简的概念, 通过放宽对约简的严格要求 (即 C 的约简必须与 C 具有完全相同的区分能力), 可以获得足够多的 C 的近似约简.下面, 我们首先在粗糙集中引入近似约简的概念, 利用近似约简我们可以构建足够多的个体分类器; 其次, 我们将近似约简与最优采样技术相结合, 从而设计出一种新的多模态扰乱策略; 最后, 基于上述多模态扰乱策略提出集成剪枝算法 EPA_AO.

2.1 近似约简

近似约简是对粗糙集中经典的约简概念的推广. 前面提到给定决策表 $DT=(U, C, D, V, f)$, 如果 R 是 C 的约简, 则 R 与 C 必须具有完全相同的区分能力, 即 $|POS_R(D)|=|POS_C(D)|$. 根据上述严格的要求, 我们可以得到关于 C 的约简, 但是往往约简的数量不够多. 对此, 有必要对上述要求进行适当地放宽, 即要求 R 的区分能力不需要完全等于 C, 而是在一定程度上近似等于 C. 基于上述考虑, 我们可以提出近似约简的概念. 与约简不同, 近似约简的区分能力可以小于 C.

定义 5. 近似约简. 给定决策表 $DT=(U, C, D, V, f)$, 对任意 $AR\subseteq C$, 如果 $|POS_C(D)|\geqslant|POS_{AR}(D)|\geqslant\delta\times|POS_C(D)|$, 则 AR 被称为 C 相对于 D 的 δ-近似约简, 其中, $\delta\in(0, 1]$ 是一个给定的阈值.

由上述定义可以看出, 近似约简 AR 的区分能力不需要完全等于 C (即 $|POS_C(D)|\geqslant|POS_{AR}(D)|$), 而是近似等于 C (即 $|POS_{AR}(D)|\geqslant\delta\times|POS_C(D)|$), 且近似的程度由阈值 δ 来控制. δ 的值越大, 则 AR 的区分能力越接近于 C, 特别是当 $\delta=1$ 时, 近似约简 AR 就变成经典

的约简. 接下来, 我们给出算法 1, 用于计算 C 中所有的近似约简.

算法 1. 近似约简的计算

输入: 决策表 $DT=(U, C, D, V, f)$, 其中, $U=\{u_1, \cdots, u_g\}$, $C=\{a_1, \cdots, a_n\}$; 参数 S 与 MI, 其中, S 是预先给定的近似约简的数量, MI 是最大迭代次数.

输出: 近似约简集 SAR.

初始化: $Core \leftarrow \varnothing$, $SAR \leftarrow \varnothing$, 其中, $Core$ 表示所有核属性的集合.

1) 采用计数排序的方法来计算 D 的 C-正区域 $POS_C(D)$[19].

2) 对任意 $1 \leqslant j \leqslant n$, 循环执行下列语句:

3) 采用计数排序的方法来计算 $POS_{C\text{-}aj}$;

4) 如果 $POS_{C\text{-}aj}(D) \neq POS_C(D)$, 则 $Core \leftarrow Core \cup \{a_j\}$.

5) Remain $\leftarrow C - Core$.

6) 如果 Remain=Ø, 则 $SAR \leftarrow SAR \cup C$, 并跳转到步骤 17).

7) 如果 Remain≠Ø, 则对任意 $1 \leqslant i \leqslant MI$, 循环执行下列语句:

8) Temp $\leftarrow Core$;

9) 从 Remain 中随机选择一个属性 a;

10) Temp←Temp $\cup \{a\}$, Remain ← Remain $- \{a\}$;

11) 对任意属性 $b \in$ Remain, 计算 b 相对于 Temp 和 D 的重要性 $Sig(b, \text{Temp}, D)$, 其中, $Sig(b, \text{Temp}, D) = (|POS_{\text{Temp}\cup\{b\}}(D)| - |POS_{\text{Temp}}(D)|) / |U|$;

12) 选择 Remain 中重要性最大的属性 $b_{\max}$;

13) Temp ← Temp $\cup \{b_{\max}\}$, Remain ← Remain $- \{b_{\max}\}$;

14) 重复执行步骤 11)–13), 直到满足近似约简的条件, 即 $|POS_{\text{Temp}}(D) \geqslant \delta \times |POS_C(D)|$;

15) 如果 Temp$\notin SAR$, 则 $SAR \leftarrow SAR \cup \{\text{Temp}\}$;

16) 如果 $|SAR| = S$, 则结束当前循环.

17) 输出 SAR.

对任意 $B \subseteq C$, 计算 D 的 B 正区域 $POS_B(D)$ 的时间复杂度通常为 $O(|U|^2)$. 在算法 1 中, 我们通过计数排序的方法[21] 来计算 $POS_B(D)$, 时间复杂度仅为 $O(|B|\times|U|)$. 在最坏的情况下, 算法 1 的时间复杂度为 $O(MI\times|C|^3\times|U|)$, 空间复杂度为 $O(|U|+|C|)$.

2.2 算法 EPA_AO

由于只采用近似约简来扰乱属性空间可能还不足以保证个体分类器的多样性, 因此, 本文将近似约简与最优采样技术结合在一起来实现一种多模态的扰乱, 即不仅对属性空间进行扰乱, 而且还利用最优采样技术来扰乱训练集. 在上述多模态扰乱策略的基础上, 我们提出了集成剪枝算法 EPA_AO. 在 EPA_AO 中, 对于每个近似约简 AR_i ($1 \leqslant i \leqslant M$), 我们首先对训练集进行多次 bootstrap 采样, 然后选择关于 AR_i 的最优采样集 O_i. 与其他采样集相比, 在 O_i 上训练的个体分类器具有最高的分类性能. 我们最终利用近似约简 AR_i 以及 AR_i 所对应的最优采样集 O_i 来训练个体分类器 IC_i ($1 \leqslant i \leqslant M$), 并对 $IC_1, \cdots, IC_M$ 进行集成.

算法 2. EPA_AO 算法

输入: 训练集 $TS=(U_1, C_1, D_1, V_1, f_1)$; 验证集 $VS=(U_2, C_2, D_2, V_2, f_2)$; 参数 H, 其中, H 表示为选择最优采样集而提前生成的 bootstrap 采样集的个数.

输出: 集成分类器 EC.

1) 利用算法 1 计算出 TS 中的所有近似约简 (令 $AR_1, \cdots, AR_M$ 分别表示这些近似约简).

2) 对任意 $1 \leqslant i \leqslant M$, 循环执行下列语句:

3) 利用 AR_i 对 TS 进行约简, 并且令 $TS_{\text{reduced}} = (U_1^*, AR_i, D_1, V_1^*, f_1^*)$ 为约简后的训练集.

4) 利用 AR_i 对 VS 进行约简, 并且令 $VS_{\text{reduced}} = (U_2^*, AR_i, D_2, V_2^*, f_2^*)$ 为约简后的验证集.

5) 对任意 $1 \leqslant j \leqslant H$, 循环执行下列语句:

6) 对 TS_{reduced} 中的 U_1^*进行 bootstrap 采样, 生成采样集 S_j;

7) 利用给定的分类算法在采样集 S_j 上训练一个个体分类器 C_j;

8) 利用 VS_{reduced} 来评价个体分类器 C_j 的性能.

9) 从 $C_1, \cdots, C_H$ 中选取分类性能最好的个体分类器 $C_{\max}$ 以及 $C_{\max}$ 所对应的采样集 $S_{\max}$, 并且令 $O_i = S_{\max}$ 表示关于近似约简 AR_i 的最优采样集, 其中, $1 \leqslant C_{\max} \leqslant H$.

10) 将 $C_{\max}$ 作为最终由近似约简 AR_i 以及最优采样集 O_i 所训练的个体分类器 IC_i.

11) 基于多数投票的策略, 将个体分类器 $IC_1, \cdots, IC_M$ 集成在一起, 从而得到集成分类器 EC.

3 实验结果及分析

为了评价 EPA_AO 的有效性, 我们开展了相关实验. 实验采用 9 个来自于 UCI 机器学习库的数据集, 并使用证据 KNN 算法来生成个体分类器. 关于这 9 个数据集的具体信息见表 1.

表 1 实验数据集

数据集	样本个数	属性个数	类别个数
Credit-g	1 000	24	2
Heart	303	13	5
Ionosphere	351	34	2
Liver	345	6	2
Pima	768	8	2
Sonar	208	60	2
Vehicle	846	18	4
Vowel	990	10	11
WDBC	569	30	2

我们基于 Java 实现了 EPA_AO. 在实验中, 对于表 1 中的任意数据集 T, 我们首先使用等宽度算法对 T 中每一个连续型属性进行离散化处理, 其中, bins=5; 其次, 我们将 T 随机划分成一个训练集 Train (占样本总数的 50%) 和一个测试集 Test (剩下 50% 的样本); 然后, 由于我们要为 EPA_AO 提供一个单独的验证集 VS 来获得当前近似约简 AR 的最优采样集, 因此我们

从 Train 中随机选择 50% 的样本来生成 *VS*.

在运行证据 KNN 算法之前, 我们要对 β_0 与 γ_s 这两个参数的取值进行设置. 在我们的实验中, β_0 被设置为 0.95, 而 γ_s 则被设置为相应类别训练样本的平均距离的倒数[12], 在经过多次取值实验后, 发现 β_0 为 0.95 时证据 KNN 算法训练出的个体分类器精度最好. 对于 EPA_AO, 我们需要对 S 和 MI 这两个参数以及阈值 δ 的取值进行设置. 在我们的实验中, S 和 MI 分别被设置为 10 和 25, 而 δ 则被设置为 0.9 (实验发现, S 设为 10 时既能避免约简个数太少而影响集成分类器的分类性能, 也不会因约简太多而影响训练的效率; MI 设为 25 时实验效果最佳; δ 设为 0.9 时能够保证约简后的个体分类器的分类能力与约简前比较接近, 不会因为分类能力下降太多而影响集成分类器的整体性能).

下面, 我们首先比较 EPA_AO 与文献 [12] 中所采用的 RSM 方法的性能. 在文献 [12] 中, RSM 也通过证据 KNN 算法来生成个体分类器. 证据 KNN 的性能依赖于 k 的值, 因此, 对于每个数据集, 我们为 k 赋予不同的取值 (即 $k=3$、$k=5$ 和 $k=7$), 从而得到不同 k 值下的实验结果.

表 2 给出了 EPA_AO 和 RSM 这两个算法在不同 k 值下的分类性能 (为了避免偶然性, 我们将全部实验都重复执行 10 遍, 表 2 中所列出的实验结果都是这 10 次实验的平均值).

表 2 EPA_AO 和 RSM 算法在不同 k 值下的准确率

数据集	$k=3$		$k=5$		$k=7$		不同k值下的平均	
	RSM	EPA_AO	RSM	EPA_AO	RSM	EPA_AO	RSM	EPA_AO
Credit-g	0.7140	0.8228	0.7170	0.8254	0.7148	0.8270	0.7153	0.8251
Heart	0.6762	0.7902	0.6881	0.7922	0.6987	0.7974	0.6877	0.7933
Ionosphere	0.8920	0.8869	0.8874	0.9085	0.8840	0.9264	0.8878	0.9073
Liver	0.6384	0.7677	0.6500	0.8520	0.6581	0.9057	0.6488	0.8418
Pima	0.7352	0.8785	0.7383	0.8669	0.7422	0.8815	0.7386	0.8756
Sonar	0.7922	0.7533	0.7621	0.7876	0.7408	0.7926	0.7650	0.7778
Vehicle	0.6787	0.7688	0.6635	08354	0.6517	0.8669	0.6646	0.8237
Vowel	0.9271	0.8582	0.9044	08824	0.8893	0.8927	0.9069	0.8778
WDBC	0.9243	0.9227	0.9239	0.9315	0.9243	0.9445	0.9241	0.9329

根据表 2, 我们可以得出这样的结论: EPA_AO 的分类准确率在大部分数据集上都比 RSM 更高. 具体而言, 如果将 k 设置为 7, 则 EPA_AO 的准确率在所有数据集上均要优于 RSM; 如果将 k 设置为 5, 则 EPA_AO 的准确率在除了 Vowel 之外的其余 8 个数据集上均要优于 RSM; 如果将 k 设置为 3, 则 EPA_AO 的准确率在其中 5 个数据集上要优于 RSM, 而在另外 4 个数据集上 (即 Ionosphere、Sonar、Vowel 和 WDBC) 要低于 RSM. 特别是, 如果我们统计多个 k 值下的平均准确率, 则可以发现 EPA_AO 的性能在总共 7 个数据集上 (除了 Vowel 和 WDBC) 均要优于 GAv1 与 GAv2. 由上述实验结果可知, 本文提出的 EPA_AO 算法其分类性能明显优于现有的集成学习算法.

接下来, 我们将比较 EPA_AO 与两种基于 GA (遗传算法) 的集成剪枝算法的性能. 这两种基于 GA 的集成剪枝算法是由 Altinçay 所提出[12], 分别称为: GAv1 (GA version 1) 和 GAv2 (GA version 2). 与本文所提出的 EPA_AO 类似, GAv1 和 GAv2 也采用多模态扰乱策略来训练个体分类器.

表 3 给出了 EPA_AO、GAv1 与 GAv2 这 3 个算法的分类准确率 (因为 GAv1 与 GAv2 能够自动选取最优的 k 值以得到最高的准确率[12], 所以这里我们只给出了 EPA_AO 的最优结果, 即 $k=7$ 时的准确率. 此外, 我们还统计了 EPA_AO 在不同 k 值下的平均准确率).

根据表 3, 我们可以得出这样的结论: EPA_AO 的分类准确率在大部分数据集上都比 GAv1 与 GAv2 这两个算法更高. 具体而言, 如果将 k 设置为 7, 则 EPA_AO 的准确率在总共 8 个数据集上均要优于 GAv1 与 GAv2, 只是在 Vowel 上表现差一些. 尤其是在以下 4 个数据集上: Credit-g、Liver、Pima 与 Vehicle, EPA_AO 的分类准确率相对于 GAv1 与 GAv2 这两个算法有明显的提升 (提升了 10% 以上). 此外, 如果我们统计 EPA_AO 在多个 k 值下的平均准确率, 则可以发现 EPA_AO 的性能在总共 7 个数据集上 (除了 Vowel 和 WDBC) 均要优于 GAv1 与 GAv2. 由上述实验结果可知, 本文提出的 EPA_AO 算法其分类性能明显优于现有的集成学习算法.

表 3 EPA_AO、GAv1 与 GAv2 的准确率

数据集	GAv1	GAv2	EPA_AO ($k=7$)	EPA_AO (不同k值下的平均)
Credit-g	0.7236	0.7184	0.8270	0.8251
Heart	0.7775	0.7861	0.7974	0.7933
Ionosphere	0.8943	0.8943	0.9264	0.9073
Liver	0.6535	0.6581	0.9057	0.8418
Pima	0.7401	0.7424	0.8815	0.8756
Sonar	0.7757	0.7689	0.7926	0.7778
Vehicle	0.6991	0.6919	0.8669	0.8237
Vowel	0.9224	0.9164	0.8927	0.8778
WDBC	0.9317	0.9356	0.9445	0.9329

4 总结

为了增加集成学习中个体分类器的多样性, 本文提出了近似约简的概念, 并由此设计出一种新的多模态扰乱策略. 该策略通过近似约简来扰乱属性空间, 并通过最优采样技术来扰乱训练集. 近似约简是对粗糙集中经典的约简概念的扩展, 它能够解决给定决策表中约简数量可能不足的问题. 在上述多模态扰乱策略的基础上, 本文提出了集成剪枝算法 EPA_AO. 实验结果表明, EPA_AO 算法的性能要优于现有的集成算法.

由于本文在 Pawlak 的经典粗糙集模型[17] 基础上提出了近似约简的概念, 而经典粗糙集模型更适合于处理离散型属性, 所以, EPA_AO 需要通过某种离散化技术将所有的连续型属性转换为离散型属性. 然而, 属性离散化可能会导致信息的丢失问题. 因此, 在下一步工作中, 我们计划将 EPA_AO 扩展到邻域粗糙集[22] 或者模糊粗糙集[23] 等扩展的粗糙集模型中, 从而可以不经过离散化过程而直接处理连续型属性.

参考文献

1 Dietterich TG. Ensemble learning. In: Arbib MA, ed. Handbook of Brain Theory and Neural Networks. 2nd ed. Cambridge: MIT Press, 2002.

2 Li H, Wang XS, Ding SF. Research and development of neural network ensembles: A survey. Artificial Intelligence Review, 2018, 49(4): 455–479. [doi: 10.1007/s10462-016-9535-1]

3 徐森, 皋军, 花小朋, 等. 一种改进的自适应聚类集成选择方法. 自动化学报, 2018, 44(11): 2103–2112.

4 朱旭辉, 倪志伟, 程美英, 等. 融合改进二元萤火虫算法和边界最小化测度的集成剪枝方法. 计算机学报, 2019, 42(6): 1252–1273. [doi: 10.11897/SP.J.1016.2019.01252]

5 Jiang F, Yu X, Zhao HB, *et al*. Ensemble learning based on random super-reduct and resampling. Artificial Intelligence Review, 2021, 54(4): 3115–3140. [doi: 10.1007/s10462-020-09922-6]

6 Buhlmann P, Yu B. Analyzing bagging. The Annals of Statistics, 2002, 30(4): 927–961.

7 Schapire RE. The boosting approach to machine learning: An overview. In: Denison DD, Hansen MH, Holmes CC, *et al*., eds. Nonlinear Estimation and Classification. New York: Springer, 2002. 149–171.

8 Ho TK. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 832–844. [doi: 10.1109/34.709601]

9 Tian Y, Feng Y. RaSE: Random subspace ensemble classification. Journal of Machine Learning Research, 2021, 22(45): 1–93.

10 Breiman L. Random forests. Machine Learning, 2001, 45(1): 5–32. [doi: 10.1023/A:1010933404324]

11 Zhou ZH, Yu Y. Ensembling local learners through multimodal perturbation. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2005, 35(4): 725–735. [doi: 10.1109/TSMCB.2005.845396]

12 Altınçay H. Ensembling evidential *k*-nearest neighbor classifiers through multi-modal perturbation. Applied Soft Computing, 2007, 7(3): 1072–1083. [doi: 10.1016/j.asoc.2006.10.002]

13 Gao J, Liu KH, Wang BZ, *et al*. Improving deep forest by ensemble pruning based on feature vectorization and quantum walks. Soft Computing, 2021, 25(3): 2057–2068. [doi: 10.1007/s00500-020-05274-z]

14 Zhang S, Chen Y, Zhang WY, *et al*. A novel ensemble deep learning model with dynamic error correction and multi-objective ensemble pruning for time series forecasting. Information Sciences, 2021, 544: 427–445. [doi: 10.1016/j.ins.2020.08.053]

15 Latinne P, Debeir O, Decaestecker C. Different ways of weakening decision trees and their impact on classification accuracy of DT combination. Proceedings of the 1st International Workshop on Multiple Classifier Systems. Cagliari: Springer, 2000. 200–209.

16 Altınçay H. Optimal resampling and classifier prototype selection in classifier ensembles using genetic algorithms. Pattern Analysis and Applications, 2004, 7(3): 285–295. [doi: 10.1007/s10044-004-0225-2]

17 Pawlak Z. Rough Sets: Theoretical Aspects of Reasoning About Data. Dordrecht: Springer Science & Business Media, 1991.

18 Denoeux T. A *k*-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man, and Cybernetics, 1995, 25(5): 804–813. [doi: 10.1109/21.376493]

19 王国胤. Rough 集理论与知识获取. 西安: 西安交通大学出版社, 2001.

20 Wu QX, Bell D, McGinnity M. Multiknowledge for decision making. Knowledge and Information Systems, 2005, 7(2): 246–266. [doi: 10.1007/s10115-004-0150-0]

21 徐章艳, 刘作鹏, 杨炳儒, 等. 一个复杂度为 max(O(|*C*||*U*|), O(|*C*|^2|*U*/*C*|)) 的快速属性约简算法. 计算机学报, 2006, 29(3): 391–399. [doi: 10.3321/j.issn:0254-4164.2006.03.006]

22 Wang Q, Qian YH, Liang XY, *et al*. Local neighborhood rough set. Knowledge-Based Systems, 2018, 153: 53–64. [doi: 10.1016/j.knosys.2018.04.023]

23 Dai JH, Hu H, Wu WZ, *et al*. Maximal-discernibility-pair-based approach to attribute reduction in fuzzy rough sets. IEEE Transactions on Fuzzy Systems, 2018, 26(4): 2174–2187. [doi: 10.1109/TFUZZ.2017.2768044]

(校对责编: 孙君艳)