基于分层筛选和动态更新的并行选择集成算法

吴梅红[1] 郭佳盛[1] 鞠 颖[1] 林子雨[1] 邹 权[1,2]
(厦门大学计算机科学系 厦门 361005)[1] (天津大学计算机科学与技术学院 天津 300072)[2]

摘 要 提出一种选择性集成学习算法,该算法利用多线程并行优化基分类器的参数,通过多层筛选和动态更新筛选信息获取最优的候选基分类器集合,解决了以往在集成学习中选择分类器效率低下的问题。集成分类器采用分解合并的策略进行加权投票,通过使用二分法将大数据集的投票任务递归分解成多个子任务,并行运行子任务后合并投票结果以缩短集成分类器的投票运行时间。实验结果表明,相对于传统方法,所提出的算法在平均精度、F1-Measure以及AUC指标上都有着显著提升。

关键词 选择性集成学习,分治算法,并行计算,分类
中图法分类号 TB183 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.01.009

Selective Ensemble Learning Algorithm Based on Hierarchical Selection and Dynamic Updating in Parallel

WU Mei-hong[1] GUO Jia-sheng[1] JU Ying[1] LIN Zi-yu[1] ZOU Quan[1,2]
(Department of Computer Science, Xiamen University, Xiamen 361005, China)[1]
(School of Computer Science and Technology, Tianjin University, Tianjin 300072, China)[2]

Abstract In this paper, a selective ensemble learning algorithm was proposed based on hierarchical selection and dynamic updating, which can optimize the parameters of classifier with multi-thread technique and select the sub sequence set of classifiers based on hierarchical selection and dynamical information. It can solve the problem in the past for choosing classifier to ensemble learning inefficiently. In addition, divide-and-conquer strategy is employed to reduce the time cost for ensemble voting. The big voting task can be divided recursively into small child task by dichotomy, then the tasks are executed in parallel and it would conquer the voting result. Experimental results show that the selective algorithm can outperform the traditional classification algorithms on F1-Measure and AUC.

Keywords Selective ensemble learning, Divide-and-conquer, Parallel computation, Classification

集成学习因为可以显著提高一个学习系统的泛化能力和分类准确度,已经得到越来越多人的关注。目前集成学习方法已经广泛应用在图像处理、生物信息学、计算机视觉和多标记分类等众多研究领域[1-4]。集成学习通过使用多个基分类器进行学习,并使用某种集成方法把每个基分类器学习结果进行整合,从而获得比单个基分类器具有更好学习效果的一种机器学习方法[5]。基分类器的构造往往由一些已有的机器学习(如朴素贝叶斯网络、决策树、支持向量机、神经网络等)算法通过训练得到。构造一个具有差异性、泛化能力强的基分类器集合是关键,同时如何从众多分类器中选择部分互补性强的基分类器进行集成也是一个关键问题。

Bagging[6]和Boosting[7-9]是目前集成学习研究最深入的两种算法,广泛应用于现实生活中而且取得了非常好的效果。Bagging是由Breiman在1996年提出的一种集成学习方法,其基本思想是利用有放回的采样从原数据集中构造有重复、同样本个数的多个训练集实例,通过Bootstrap方法增加训练数据集的差异性生成多个具有差异性的基分类器,从而提高集成分类器的泛化能力。该方法主要用于不稳定(不稳定是指当训练集中数据有微小的变化时,会导致模型有很大的变化)的学习算法,适合并行化训练。Boosting是一类集成学习算法的总称,最开始是由Schapire提出的,它有许多变种算法,其中AdaBoost[7]算法最为流行。AdaBoost在每次的次迭代中,通过最小化训练集的加权误差来训练基分类器,然后使用基分类器的加权误差更新训练实例上的权值分布,增加错误分类的实例权值,减小正确分类实例的权值,在训练下一个分类器时,则使用更新后的实例权值分布用于基分类器的训练,并重复此过程。通过改变训练样本的权重大小,训练多个分类器,并将这些分类器进行线性组合,从而提高分类器的预测精度[7]。

由于集成学习利用多个基分类器,能够获得比单个学习器更强的泛化能力[10,11],因此直观上认为可以通过大量添加基分类器来获得更好的集成性能。但是随着个体分类器个数的不断增加,集成学习的预测速度明显下降,所需要的计算内存空间也随之上升[5]。目前对集成学习的研究已经不仅仅是

到稿日期:2015-08-12 返修日期:2015-10-30 本文受国家自然科学基金(61370010,61303004,31200769)资助。
吴梅红(1982-),女,博士,副教授,主要研究领域为生物医学工程,E-mail:wmh@xmu.edu.cn;郭佳盛(1991-),男,硕士生,主要研究领域为集成学习策略与应用;鞠 颖(1977-),女,博士,副教授,主要研究领域为生物功能建模与仿真;林子雨(1978-),男,博士,助理教授,主要研究领域为云数据库;邹 权(1982-),男,博士,研究员,主要研究领域为生物信息学和数据挖掘,E-mail:zouquan@nclab.net(通信作者)。

集成方法的提出和改进，研究表明对所有的基分类器进行集成并不能获得最好的集成效果，如何从已有的基分类器集合中选取最优的基分类器子集进行集成是目前的研究重点[10]。

2002 年，周志华等人提出了“选择性集成”的概念[11]，旨在使用某种指标从已有的多个基分类器中选取一部分基分类器集合用于构建最终的集成分类器，由于舍弃了一些效果性能不好的基分类器，因此集成分类器的性能得到了提高；同时其所提出的 GASEN[11] 算法无论在回归还是分类问题上的性能都要优于 Bagging 和 Boosting，使用 GASEN 产生的基分类器个数也远少于 Bagging 和 Boosting 产生的基分类器个数。

选择性集成学习通过剔除性能不佳的分类器可以有效地提高集成学习的训练速度和泛化能力，但是由于目前处理的数据量规模越来越大、维度越来越高，导致集成学习的难度也急剧增加。一方面，基分类器在训练大规模数据时会消耗大量计算机内存；另一方面，训练如此大规模的数据以及集成这些基分类器会耗费大量的时间[12]。目前处理大规模数据集的方式主要包括减小训练数据集的大小，降低训练数据集的维度以及使用分布式方法处理大数据等方法[13,14]。本文基于集成学习的理论，提出基于分层筛选和动态更新的选择性集成算法。该算法分为两个部分，第一部分是基分类器的训练及其参数优化，通过设定指定步长对基分类器进行并行化参数优化，从而获取该基分类器的最优训练参数。第二部分是基分类器的集成，基分类器的集成是一个多层筛选模型，每一层都是一个组合优化问题，每一层的解都是在上一层解的基础上产生的，通过评价指标函数，判断是否接受某一层的解。算法通过引入模拟退火中的 Metropolis 准则可有效地解决组合优化问题中的局部最优解问题。采用分解合并的加权投票策略，以基分类器的准确率作为各自的权重大小，如果投票的数据集比较大，则将该投票任务递归地分解成多个小数据集投票子任务，然后并行运行子任务，最后合并各个子任务的预测投票结果，从而有效缩短了集成分类器的预测投票时间。通过本文的实验结果数据比较分析可以看出，相对于传统的分类器算法，该算法在多项指标上都有着显著的提升。

1　基于分层筛选和动态更新的选择性集成算法

选择性集成算法采用的是“over produce and choose”的策略[10]。选择性集成算法可以分为两个步骤。第一步是构建大量的候选基分类器集合，用于下一步的分类器集成。第二步是采取某种选择策略选取一组最优分类器进行组合，从而提高整体分类器的泛化性能和分类精度。目前，选择性集成算法主要包括聚类、排序、选择、优化以及其他方法[5,12]。对于不同的问题，这些算法有各自的优缺点，没有一种方法是通用的。

本文提出一种基于分层筛选和动态更新筛选信息的选择性集成算法，该算法由两层模型构成，如图 1 所示。第一层是采用并行的方法对每个基分类器的参数进行优化，选取基分类器的最优训练参数，这样可以有效缩短基分类器的参数优化和训练时间。第二层是基分类器的集成，是一个分层筛选最优基分类器组合的过程，每一层都会对基分类器集合的当前选择概率进行更新。通过引入模拟退火 Metropolis 准则避免经过多层筛选后的基分类器组合为局部最优解的问题。

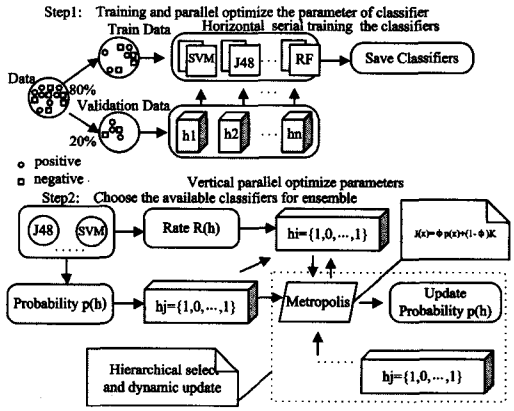

图 1　选择性集成算法框架

1.1　基分类器的训练和参数并行化

基分类器的构造是集成学习的前提，构造具有多样性的候选基分类器集合一直是研究的重点[10,15,16]。本文采用 22 个不同的基分类器（包括 SVM，J48，RF，Logistic 等其他常用分类器）进行训练。假定 $L=\{0,1\}$，$\vec{x}\in\mathcal{R}^n$ 为具有 n 个特征属性的向量，$D=\{(\vec{x}_1,l_1),(\vec{x}_2,l_2),\cdots,(\vec{x}_n,l_n)\}$，$l_i\in L$ 为给定的二分类问题数据集。对给定的训练数据集 D 进行随机扰动排序之后，通过设定优化百分比参数 *Percent*，选取部分数据作为基分类器的参数优化数据集，其余的作为训练数据集。对于没有训练参数的基分类器，则将所有的数据集用于基分类器的训练。

基分类器的建立包括两个阶段，第一阶段是建立基分类器的基本结构模型，第二阶段是对建立的结构模型所包含的训练参数进行优化。为了避免过拟合问题，这两个阶段分别使用不同的数据集，使用训练数据集训练基分类器，使用参数优化数据集对当前的基分类器模型的训练参数进行评估，找出该基分类器训练效果最好的训练参数。为了避免处理大规模数据时导致的计算内存溢出问题，本文提出的算法在水平方向上对基分类器集合进行串行训练，竖直方向上对单个基分类器进行并行化的参数优化，并将具有最优训练参数的基分类器保存到硬盘，在集成阶段重新将训练保存好的基分类器集合导入到内存进行集成，这样可以有效缓解单机内存不够用的问题。

1.2　基分类器的训练和参数并行化

选择性集成通过采用某种评价指标，从现有的基分类器集合中选取一组最优的基分类器组合用于构建集成分类器，从而提高集成学习的分类准确率和泛化能力。本文采用基分类器的多样性度量和集成分类器的准确率作为评价指标，从而选取一组具有更优泛化能力和准确率的基分类器组合。

分类器的多样性度量评价指标分为成对多样性度量和非成对多样性度量方式，本文采用非成对的相互一致性系数 k 作为基分类器的多样性评价指标[9]。设 $H=\{h_1,h_2,\cdots,h_t\}$ 为具有 t 个基分类器的候选集合，对于二分类问题，只有 0/1 两种分类结果，即分类正确为 1，分类错误为 0，则相互一致性系数 k 可表示为：

$$k=1-\frac{1}{2\bar{p}(1-\bar{p})}Dis_{av} \tag{1}$$

其中：

$$\bar{p}=\frac{1}{nt}\sum_{j=1}^{n}\sum_{i=1}^{t}h_i(x_j,l_j) \tag{2}$$

$$Dis_{av}=\frac{t}{t(t-1)}\sum_{i=1}^{t}\sum_{k=1,i\neq k}^{t}Dis_{i,k} \tag{3}$$

$$Dis_{i,k}=\frac{N^{01}+N^{10}}{N^{11}+N^{10}+N^{01}+N^{00}} \tag{4}$$

N^{ab} 的意义如下：

$$N^{ab}=\begin{cases}N^{00}, & \text{if } h_i\to F(0)\ \&\ h_k\to F(0)\\ N^{01}, & \text{if } h_i\to F(0)\ \&\ h_k\to F(1)\\ N^{10}, & \text{if } h_i\to F(1)\ \&\ h_k\to F(0)\\ N^{11}, & \text{if } h_i\to F(1)\ \&\ h_k\to F(1)\end{cases} \tag{5}$$

相互一致性系数 k 作为一种相似度度量方式，k 值越高，集成分类器集合中的基分类器之间的相似度越高，差异性也就越低。

准确率(Accuracy)是一个被广泛使用的评价指标，集成学习通过多个基分类器的学习，可以提高整体分类器的准确率。

本文将分类器的差异性和准确率这两项指标进行综合考虑，提出了一种新的评价度量方式 F_2：

$$F_2=\alpha\times Acc+(1-\alpha)\times k,0\leqslant\alpha\leqslant 1 \tag{6}$$

其中，α 为两个评价指标的调节因子，α 越大则准确率所起到的作用就越大。通过调节 α 因子，得到一个泛化能力强且准确率高的分类器。

1.3 基分类器的训练和参数并行化

假设 $H=\{h_1,h_2,\cdots,h_n\}$ 为经过参数优化后的基分类器集合，按照顺序使用布尔值 0,1 表示是否选择了某个基分类器用于集成学习，则 $\vec{x}=\{1,1,0,1,\cdots,1\}$ 表示集成学习的基分类器的一个组合。基分类器的集成是一个多层选择模型，每一层都是一个组合优化问题，组合优化的评价指标函数为 F_2。每一层结束之后，依据 Metropolis 准则决定是否接受新解，如果接受，则动态增加在该新解序列集合中的候选基分类器的选择概率大小，同时减小不在该序列集合中的分类器的选择概率大小。这样通过多层的序列选择，性能较好的基分类器有更大的机会作为集成的候选弱分类器，同时也可以使得每个基分类器均有机会作为候选分类器进行集成。

具体算法设计流程如图 2 所示。

Input: parameter φ, T, diversity K, threshold θ, avaliable classifiers $H=\{h_1,h_2,h_3,\cdots,h_n\}$, probability $p(x_r)(r\in 1,\cdots,n)$ and the rate of classifers $rate=\{rate_1,rate_2,rate_3,\cdots,rate_n\}$

Output: h^r as the best sulution

1. $p(x_r)$ =rate, get h^x base on $p(x_r)$;
2. while i$\leqslant\theta$ do
3. get h^r base on $p(x_r)$;
4. $J(h^r)=\varphi * p(x_r)+(1-\varphi) * K_r$;
5. compare $J(h^r)$ and $J(h^x)$;
6. accept h_r as h^x by Metropolis;
7. update $p(x^r)$ in h^r;
8. end while

图 2 分层筛选和动态更新的选择性集成算法

如图 2 算法所示，算法依据 Metropolis 准则接受 h^r，即如果 $\min\{1,\exp(\nabla J/T)\}>\beta$，则 $h^r=h^r$。其中 $\Delta J=J(h^r)-J(h^r)$，J 为集成分类器的 F_2 度量评价指标函数，β 为在[0,1]区间内的均匀分布随机数。通过引入 Metropolis 准则可以有效避免求得的组合优化解为局部最优解。

1.4 基于分解合并的加权集成投票策略

选择性集成的关键在于构建出具有多样性的基分类器集合，通过剔除性能差劲和冗余的基分类器来提高集成的效率和性能。下一步便是集成分类器的投票预测，对于集成学习使用加权投票策略会优于简单平均投票，因此本文采用加权投票策略对集成分类器进行投票预测[17]。加权投票定义如下：

$$F(X)=\sum_{i=1}^{N}w_i f(\vec{x_i},l_i),w_i\geqslant 0,\sum_{i=1}^{N}w_i=1 \tag{7}$$

其中，N 表示用于集成的基分类器个数，w_i 为各个基分类器的权重大小。本文采用基分类器分类准确度作为基分类器的权值大小，权向量 $w=[w_1,w_2,\cdots,w_N]^T$，假定第 i 个基分类器的准确度为 p_i，则经过归一化后该基分类器的权值大小为：

$$w_i=p_i/\sum_{i=1}^{N}p_i \tag{8}$$

处理的数据集比较大的时候，如果采用以往传统的单任务单线程的集成投票方法，势必会花费大量时间。

具体算法设计流程如图 3 所示。

Input: avaliable classifiers $H=\{h_1,h_2,h_3,\cdots,h_n\}$, predict Data D_x and threshold θ;

Output: result of D;

1. if Instance(D_x)$\geqslant\theta$ then
2. Divide D_x into D_1 and D_2;
3. $D_1\to D_x$ and go to step1;
4. $D_2\to D_x$ and go to stpe2;
5. end if
6. get a set data $D=\{D_1,D_2,\cdots,D_n\}$,
7. parallel predict D;
8. for i=1 to n do
9. combine the result of D;
10. end for

图 3 基于分解合并的集成投票策略

为了提高集成学习的投票预测速度，本文采用分解合并的策略，如图 3 所示，通过设置集成投票处理数据集任务的阈值大小，采用二分法将大规模数据集投票任务递归分解成多份小数据集投票任务，这样就将一个大问题递归分解成多个子问题，最后再将子问题并行运行后合并投票结果，以此提高集成学习对实验数据集的投票运行速度。

2 实验分析

根据问题的类型、样本大小等因素，本文采用 UCI 机器学习数据库[18]中的 5 组数据集进行实验，以测试本文提出的选择性集成算法的分类性能，表 1 列出了这些数据集的基本信息。

Liver，Ionosphere，Prima-diabetes 和 Adult 数据集都是二分类问题，Letter 数据集为多分类问题，共有 26 个类别。由于本文所使用的许多基分类器更适合用于解决二分类问题，因此对于多类别的数据集 LETTER，通过将类别为 0－13 的标记为 0、将类别为 14－26 的标记为 1 并将其转换为二分类问题，从而本文最后所实验的数据集都是二分类问题。

表 1 实验所用 UCI 数据集

Datasets	No of attribute	No of classes	No of instances
Liver	7	2	345
Ionosphere	34	2	351
Prima-diabetes	9	2	768
Letter	17	2	20000
Adult	15	2	32561

2.1 算法性能指标分析

本文采用 4 个评价指标用于测试基于分层筛选和动态更新的选择性集成算法（命名为 HSDU）的性能，包括准确度（Acc）、F1 值（F1-Measure）、AUC 以及 3 种指标的平均值（MEAN）。其中 $F1$ 值为召回率和准确率的调和平均数，Acc 和 $F1$ 值的定义如下：

$$Acc=\frac{TP+TN}{TP+TN+FP+TN} \tag{9}$$

$$F1=2\times\frac{precision\times recall}{precision+recall} \tag{10}$$

$$precision=\frac{TP}{TP+FP} \tag{11}$$

$$recall=\frac{TP}{TP+FN} \tag{12}$$

其中，TP 为正类判定为正类的样本个数，FP 为负类判定为正类的样本个数，FN 为正类判定为负类的样本个数，TN 为负类判定为负类的样本个数。本文对所有的数据集都进行了 10 次的十折交叉验证分析，通过计算统计得出每种方法在不同数据集上对于各项评价指标的均值和方差。表 2－表 6 分别列出数据集 Liver，Ionosphere，Prima-diabetes，Letter 以及 Adult在不同方法下的评价指标结果。

表 2　不同方法在数据集 Liver 上的性能指标

Method	Performance metrics(%)			
	Acc	F1	AUC	MEAN
Bagging	**69.97±2.0**	69.14±3.2	**73.54±1.7**	**70.88±2.3**
AdaBoost	66.09±0.0	63.37±0.0	68.41±0.0	66.23±0.0
RF	66.52±1.8	68.32±2.6	73.41±2.1	69.42±2.2
SMO	58.23±2.0	37.45±0.0	50.35±0.0	48.68±0.6
NaiveBayes	55.36±0.0	54.99±0.0	64.01±0.0	58.12±0.0
J48	68.70±0.0	66.77±0.0	66.50±0.0	67.32±0.0
HSDU	69.69±0.5	**69.22±0.5**	72.53±1.3	70.48±0.8

表 3　不同方法在数据集 Ionosphere 上的性能指标

Method	Performance metrics(%)			
	Acc	F1	AUC	MEAN
Bagging	91.57±0.6	90.85±0.4	95.30±0.2	92.57±0.4
AdaBoost	90.88±0.0	89.56±2.0	94.40±0.0	91.61±0.6
RF	92.54±0.8	91.99±0.7	96.94±0.2	93.82±0.5
SMO	88.60±2.0	86.95±0.0	85.35±0.0	86.30±0.6
NaiveBayes	82.62±0.0	81.86±0.0	93.48±0.0	85.97±0.0
J48	91.45±0.0	90.46±0.0	89.23±0.0	90.38±0.0
HSDU	**93.00±1.1**	**92.87±0.5**	96.40±0.7	**94.09±0.7**

表 4　不同方法在数据集 Pima-diabetes 上的性能指标

Method	Performance metrics(%)			
	Acc	F1	AUC	MEAN
Bagging	75.94±0.8	72.27±0.4	81.59±0.4	76.60±1.7
AdaBoost	74.35±0.0	70.58±0.0	80.08±0.0	75.00±0.0
RF	74.22±1.5	70.77±1.0	78.69±1.5	74.56±1.3
SMO	77.34±0.0	73.13±0.0	71.95±0.0	74.14±0.0
NaiveBayes	**76.30±0.0**	73.29±0.0	**81.86±0.0**	**77.15±0.0**
J48	73.83±0.0	70.81±0.0	75.14±0.0	73.26±0.0
HSDU	75.63±1.1	**74.6±0.42**	81.00±0.1	77.07±0.5

表 5　不同方法在数据集 Letter 上的性能指标

Method	Performance metrics(%)			
	Acc	F1	AUC	MEAN
Bagging	94.81±0.0	94.73±0.0	98.99±0.0	96.18±0.0
AdaBoost	69.37±0.0	68.75±2.0	78.86±2.0	72.33±1.3
RF	96.00±0.0	96.06±0.0	99.40±4.1	97.15±1.4
SMO	73.19±1.2	73.18±0.0	73.21±0.0	73.19±0.0
NaiveBayes	70.65±0.0	70.65±0.0	79.84±0.0	73.71±0.0
J48	92.92±0.0	92.92±0.0	94.23±0.0	93.36±0.0
HSDU	**96.20±0.0**	**96.17±0.0**	**99.36±0.0**	**97.22±0.0**

表 6　不同方法在数据集 Adult 上的性能指标

Method	Performance metrics(%)			
	Acc	F1	AUC	MEAN
Bagging	85.17±0.0	78.48±0.0	90.17±0.0	84.61±0.0
AdaBoost	83.97±2.0	74.49±0.0	87.14±2.0	81.87±1.3
RF	84.49±0.0	77.33±0.0	87.98±0.0	83.27±0.0
SMO	84.91±0.0	84.20±0.0	75.30±0.0	81.47±0.0
NaiveBayes	83.43±0.0	74.80±0.0	89.21±0.0	82.48±0.0
J48	86.23±0.0	80.00±0.0	89.16±0.0	85.13±0.0
HSDU	**85.18±0.0**	**84.23±0.0**	90.10±0.0	**86.50±0.0**

从以上 5 个表格结果可以看出，集成分类器 Bagging，AdaBoost，RF（Random Forest）的性能效果从整体上看明显比单个基分类器要好。对于 Liver 数据集，Bagging 分类器的性能效果是最好的，其次是本文提出的 HSDU 算法。HSDU 在数据集 Ionosphere 上取得了最好的效果。而对于 Prima-diabetes 数据集，基分类器 NaiveBayes 取得了最好的分类效果，可见分类器的分类性能和训练数据集有着很大的关系。对于数据集 Letter，除了 AdaBoost 方法外，集成分类器的性能依然要优于单个基分类器。对于 Adult 数据集，HSDU 在 Acc，F1，MEAN 指标上都明显优于其他分类器。可以看出，本文提出的 HSDU 的选择性集成算法相对于传统的单分类器或者集成分类器都有着显著的性能提升。

2.2 分类器参数的并行化优化分析

分类器集成之前，首先需要构造多个基分类器集合，不同的基分类器有不同的训练参数，参数的选择对基分类器的性能有着重要的影响。当前计算机往往都是多核的，因此通过多核多线程并行可以大大减少基分类器的参数优化时间。对于基分类器的参数优化，本文通过设定指定步长，每次定量增加参数值，采用多核多线程并行的方式对这些参数值进行训练，从而选取出该基分类器的最优训练参数值。

实验采用的数据集如表 2－表 6 所列。实验在 Windows 服务器上运行，采用 8 线程对每个基分类器的参数进行优化。表 7 列出了并行化参数优化和串行化参数优化的时间效率结果。

表 7　基分类器并行参数优化效率结果

Method	Sequential(s)	Parallel(s)	Reduce time(s)
Bupa	2.437	2.165	0.272
Ionosphere	2.947	2.776	0.171
Prima-diab	3.536	3.168	0.368
Letter	176.881	142.767	34.114
Adult	431.417	349.472	81.945

可以看出，通过多核多线程并行可以大大减少基分类器的参数优化时间，并且随着数据集大小的增加，这种加速效果更加明显。原因在于，基分类器的并行化参数优化能够充分利用计算机资源，基分类器的优化时间是这些参数集合中训练时间最长的，而采用单线程所耗费的时间则是这些参数训练时间的总和。

2.3 基于分解合并的集成投票运行时间分析

本文采用分解合并的集成投票策略，通过设置集成分类器的投票任务大小，使用二分法递归地将集成分类器的投票任务分解为多个子任务，然后并行运行子任务，最后合并投票结果。表 8 比较了集成分类器在处理 ADULT 数据集时采用分解合并策略和传统的单任务单线程的方法进行集成投票的运行时间。表 9 比较了集成分类器在基分类器个数一致的情况下对不同大小的 ADULT 测试集上采用分解合并和传统单任务单线程方法的运行时间结果。

表 8 集成分类器个数不同时的运行时间比较

分类器个数	单线程执行时间(ms)	分解合并执行时间(ms)	加速(ms)
2	128.4	116.4	12.0
3	185.6	134.4	51.2
4	234.4	188.2	46.2
5	249.8	196.8	53.0
6	383.8	284.4	99.4
7	405.4	277.2	128.2

表 9 数据集大小不一样时的运行时间比较

数据集大小	单线程执行时间(ms)	单线程执行时间(ms)	加速(ms)
4000	160.8	136.6	24.2
8000	226.2	183.4	42.8
12000	288.6	243.2	45.4
16000	303.6	214.6	89.0
20000	339.4	248.4	91.0
24000	404.2	275.0	129.2
28000	435.8	297.6	138.2

从图 4 可以分析看出，采用分解合并的策略可以有效减少集成投票的运行时间，并且随着集成分类器的基分类器个数的不断增加，这种提升效果也更加明显，加速的时间随着基分类器个数的增加呈线性增加。同样，可以从图 5 看出，随着测试数据集的不断增加，采用分解合并策略所带来的时间加速效果也越来越明显。因此，通过使用基于分解合并的集成投票策略可以有效减少集成分类器用于投票的运行时间。

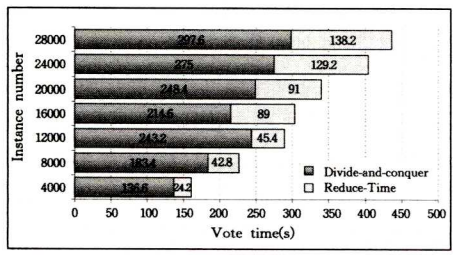

图 4 采用分解合并和传统方法的集成投票的时间比较

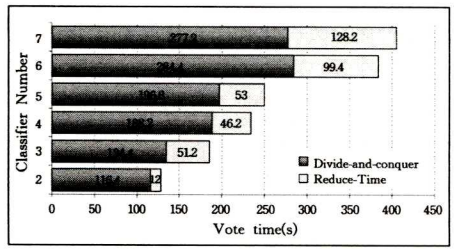

图 5 采用分解合并和传统方法的集成投票的时间比较

结束语 本文提出基于分层筛选和动态更新的选择性集成算法，该算法包括基分类器集合的构造和集成方式。基分类器的集成是一个多层筛选模型，每一层通过提出的评价指标函数判断是否为当前可接受解，同时对当前的每个候选基分类器的选择概率信息进行实时更新。每一层都是在上一层的基础上产生新的候选分类器集合，通过多层筛选之后可以得到一个近似的最优候选分类器集合用于集成学习。对多个来自 UCI 的数据集进行测试，结果表明基于 HSDU 的选择性集成算法在多项评价指标上都取得了良好的效果。但是由于部分基分类器更适合用于解决二分类问题，因此本文实验的研究对象都是基于二分类问题，多分类、多标记问题是今后的一个研究工作。

参 考 文 献

[1] ZHANG Min-ling, ZHOU Zhi-hua. A review on multi-label learning algorithms [J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(8): 1819-1837.

[2] WEI Le-yi, LIAO Ming-hong, GAO Yue, et al. Improved and Promising Identification of Human MicroRNAs by Incorporating a High-quality Negative Set [J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2014, 11(1): 192-201.

[3] HU Yao, JIN Zhong-ming, SHI Yi, et al. Large scale multi-class classification with truncated nuclear norm regularization [J]. Neurocomputing, 2015, 148: 310-317.

[4] DENG Chao, GUO Mao-zu. Tri-training and data editing based semi-supervised clustering algorithm [J]. Journal of Software, 2008, 19(3): 663-673. (in Chinese)
邓超，郭茂祖. 基于 Tri-Training 和数据剪辑的半监督聚类算法[J]. 软件学报，2008，19(3)：663-673 .

[5] ZHANG Chun-xia. A Survey of Selective Ensemble Learning Algorithms [J]. Chinese Journal of Computers, 2011, 34(8): 1399-1410. (in Chinese)
张春霞. 选择性集成学习算法综述. 计算机学报[J]，2011，34(8)：1399-1410.

[6] LEOoB. Bagging predictors [J]. Machine learning, 1996, 24(2): 123-140.

[7] RÄTSCH, GUNNAR, ONODA T, et al. Soft margins for AdaBoost [J]. Machine learning, 2001, 42(3): 287-320.

[8] ROBERT S. The strength of weak learnability [J]. Machine Learning, 1990, 5(2): 197-227.

[9] YOAV F. Boosting a weak learning algorithm by majority [J]. Information and Computation, 1995, 121(2): 256-285.

[10] LIN Chen, CHEN Wen-qiang, QIU Cheng, et al. LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy [J]. Neurocomputing, 2014, 123: 424-435.

[11] ZHOU Zhi-hua, WU Jian-xin, TANG Wei. Ensembling neural networks: many could be better than all [J]. Artificial intelligence, 2002, 137(1): 239-263.

[12] HAO Hong-wei, WANG Zhi-bin, YIN Xu-cheng, et al. Dynamic selection and circulating combination for multipleclassifier systems[J]. Acta Automatica Sinica, 2011, 37(11): 1290-1295. (in Chinese)
郝红卫，王志彬，殷绪成，等. 分类器的动态选择与循环集成方法[J]. 自动化学报，2011，37(11)：1290-1295.

[13] CAI Deng, ZHANG Chi-yuan, HE Xiao-fei. Unsupervised feature selection for multi-cluster data[C] // Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2010: 333-342.

[14] ZOU Quan, LI Xu-bin, JIANG Wen-rui, et al. Survey of MapReduce Frame Operation in Bioinformatics[J]. Briefings in Bioinformatics, 2014, 15(4): 637-647.

[15] ZOU Quan, GUO Jia-sheng, JU Ying, et al. Improving tRNAscan-SE annotation results via ensemble classifiers [J]. Molecular Informatics, 2015, 34(11/12): 761-770.

[16] YANG Chun, YIN Xu-cheng, HAO Hong-wei. Classifier Ensemble with Diversity: Effectiveness Analysis and Ensemble Optimization [J]. Acta Automatica Sinica, 2014, 40(4): 660-674. (in Chinese)
杨春，殷绪成，郝红卫，基于差异性的分类器集成：有效性分析及优化集成[J]. 自动化学报，2014，40(4)：660-674.

[17] LIN Chen, ZOU Ying, QIN Ji, et al. Hierarchical Classification of Protein Folds Using a Novel Ensemble Classifier [J]. PLoS One, 2013, 8(2): e56499 .

[18] MOSHE L. UCI machine learning repository [OL]. http://archive.ics.uci.edu/ml.