文章编号：2095-2163(2020)11-0021-06 中图分类号：TP391 文献标志码：A

二类不均衡数据分类问题常用策略研究

杨小军，刘 志，王力猛，刘 文
（国防大学 联合勤务学院，北京 100858）

摘 要：类分布不均衡问题在现实世界中广泛存在，针对不均衡数据集的分类方法及其性能评估方法，都与传统分类算法大相径庭。本文在分析常用的二类不均衡数据分类策略的基础上，选取了十个公开的 KEEL 科研数据集，用 G-mean 值和 AUC 值分别衡量分类器的准确率和泛化性能。在 KEEL 平台上对常用的三类策略中的 12 种方法的性能进行了验证，明确了算法各自的适用情况。

关键词：二类不均衡数据分类；重采样方法；代价敏感学习算法；集成学习算法；KEEL

A comparative study of common strategies for binary classification of Imbalanced data

YANG Xiaojun, LIU Zhi, WANG Limeng, LIU Wen
(Joint logistics college , National Defense University, Beijing 100858, China)

【Abstract】 Class distribution imbalance is a widespread problem in the real world. The classification methods and performance evaluation methods for imbalanced data sets are quite different from the traditional classification algorithms. Based on the analysis of the commonly used binary imbalanced data classification strategy, selected ten the public KEEL scientific research data sets, using G-mean value and the AUC value measuring accuracy and generalization performance of the classifier. On KEEL platform, the performance of 12 methods of three commonly used strategies was experimentally verified, made clear the suitable situation of each algorithm respectively.

【Key words】 Binary classification of imbalanced data; resample method; Cost-sensitive learning method; ensemble learning method; KEEL

0 引 言

分类是数据挖掘领域的一类重要问题，现有的分类方法都很成熟，如决策树、支持向量机、朴素贝叶斯方法等，并利用这些方法成功地解决了许多实际问题。但随着应用范围的扩大和研究的深入，分类方法在使用过程中遇到了数据样本分布不均衡问题。通常称数据分布不均衡的数据集为不均衡数据集，数据分布不均衡表现为两种形式：一是类间数据分布不均衡；二是在某一类样本的内部存在着类内不均衡。在不均衡数据集中，将样本数量少的类称为少数类或正类，样本数量多的类称为多数类或负类。

对不均衡数据进行正确分类，是数据分类的一个难题。问题来源于不均衡数据集的样本分布特点，以及传统分类算法固有的局限性。传统分类算法的重要前提是：数据集中各样本比例是均衡的；以总体最大精度为目标，很容易忽略少数类；所有的分类错误代价都相同。因此，如果用传统的分类器来直接处理不均衡数据集，会造成少数类样本的分类精度较差，尤其是数据不均衡严重时更是如此。鉴于目前研究不均衡分类问题都是基于不均衡的两类问题，则本文主要研究比较二类不均衡数据分类问题的常用策略。

1 不均衡数据分类策略

常用的不均衡数据分类策略主要有如下几类：在数据层面，通过重采样来解决数据分布不均衡状况；在算法层面，通过代价敏感算法或是集成算法提升不均衡数据分类时的性能；通过数据层面与算法层面相结合的策略进行改进。

1.1 数据层面的处理方法

由于不均衡数据集是数据样本之间比例不均衡，可通过对各类别数据的增删，重新实现不同类别数据样本之间的平衡。数据重采样是最具代表性的数据层面处理办法，可将其分为欠采样、过采样，以及二者结合的混合采样方法。最简单的重采样为随机过采样(ROS)方法和随机欠采样(RUS)方法。通过简单复制/删除部分样本的方式，达到平衡二类样本比例的目的。而随机方法的缺点是增加了过学习

作者简介：杨小军(1977-)，男，硕士，副教授；主要研究方向：数据挖掘与分析。

收稿日期：2020-09-25

的概率。因此目前考虑更多的是启发式方法。

Chawla 提出的 SMOTE[1]方法,是一种经典的启发式过采样方法。SMOTE 方法首先为每一个少数类样本随机地挑选出几个相邻的样本,然后在这个少数类样本和挑出的邻近样本的连接线上,以随机方式取点,生成没有重复的少数类样本。因此,在很大程度上解决了随机过采样方法产生的过拟合问题。此后,在 SMOTE 方法的基础上形成了大量的改进算法:如 D-SMOTE 过抽样算法,是采用求最近邻样本均值点的方法来生成少数类样本;N-SMOTE 算法[2],则采用了周围空间结构信息的邻居计算公式来生成少数类样本等等。

启发式欠采样方法为达到更好的分类效果,采用方法去除掉那些远离分类边界的、有数据重叠的、且对分类作用不大的多数类样本。典型的欠采样方法有 Tomek links 方法[3]和 ENN 方法等。Tomek links 方法是先判断两个不同类样本之间是否构成了 Tomek links,是则进行样本剪辑;ENN 算法的基本思想是,删除离每个多数类样本最近的 3 个近邻样本中的 2 个。在实际应用中,为了达到最佳效果,一般将各种欠采样和过采样方法混合使用。在增加少数类数据样本同时,减少了多数类数据样本,最终达到两类数据样本平衡的目的。SMOTE + Tomek links、SMOTE+ENN[4]是典型的混合采样方法。

1.2 算法层面处理方法

在算法层面,不均衡数据学习常用的方法有:代价敏感算法、集成学习方法、单类学习方法和特征选择方法。

1.2.1 代价敏感算法

传统分类器以实现样本整体误差最小为最终目标。在训练过程中,由于数量偏少的缘故,少数类样本的预测准确率很低,甚至出现被忽略的情况。为了提升少数类的重要程度,代价敏感算法给少数类样本造成的误差施加更大的惩罚。算法的中心思想是:运用该方法训练分类器的目标是最小化样本的整体误分代价,不再追求实现样本整体误差最小化。代价敏感算法的核心是代价矩阵的设计,其设计是否合理,最终决定了分类模型的性能。在二分类问题中,代价矩阵见表 1。

其中,C_{ij} 表示第 i 类样本被误分成 j 类的代价,应赋大于 0 的值。左对角线上的元素 C_{ii} 表示被正确分类的代价,其取值应为 0。重要的类别应赋更大的代价,如 $C_{ij} > C_{ji}$ 表示第 i 类样本比第 j 类更重要。在类不均衡学习中,一般更为关心少数类样本。如癌症检测中的指标异常、机器故障检测中出现的异常等。因此可将少数类视为重要类,在代价敏感学习中赋予更大的错分代价[5]。但误分代价具体取值难以确定。

表 1 二分类问题的代价矩阵

Tab. 1 The cost matrix of binary classification

	预测第 i 类	预测第 j 类
真实第 i 类	C_{ii}	C_{ij}
真实第 j 类	C_{ji}	C_{jj}

1.2.2 集成学习算法

集成算法是将多个弱分类器组合构造成一个强分类器。由于单个算法能力有限,找到的多数是局部最优解,而非全局最优解。集成学习算法对多个局部最优解进行综合,可以提升分类器的性能,已被证明是一种能有效解决不均衡问题的技术。典型的集成算法有装袋方法(Bagging)和提升方法(Boosting),其主要思想是先对训练集进行不同方式的训练,得到不同的基分类器;再对基分类器进行组合,最终达到提升集成分类器学习效果的目的。在 Bagging 算法中,为了提高集成分类器泛化能力,以有放回的方式从原始训练集中随机选取出若干样例形成训练集,多次选取不同训练集以增加基分类器差异度。AdaBoost 算法是 Boosting 方法中的代表,通过在迭代中加大被错误分类样本的权重,减少被正确分类样本的权重,由有差异的训练样本集得到不同的基分类器,最终经过加权集成为最终的分类器。在迭代过程中,Bagging 算法每个样本的权重都一样,而 Boosting 算法却能够根据样本的错误率不断调整样本的权重。因此,在处理不均衡分类问题时,基于 Boosting 的算法在一定程度上优于基于 Bagging 的算法[6]。

在实际处理不均衡数据分类时,通常将数据层面的方法与算法层面的方法相结合,解决不均衡分类问题。如,将采样技术和集成算法结合。其中最典型的是 Nitesh V.Chawla 提出的 SMOTEBoost[7]方法。该方法通过结合 SMOTE 过采样技术和 AdaBoost 提升方法,来解决不均衡数据分类问题。SMOTEBoost 算法在训练开始前,先使用 SMOTE 方法生成少数类样本,再使用 Adaboost 方法对样本分类,提升了少数类样本的分类准确率,避免了过拟合。此外,将采样和代价敏感算法相结合,也是不均衡数据学习的一类重要方法。

对常用的集成算法进一步集成就形成了混合集成算法。为防止采用降采样技术后,造成多数类样

本信息丢失的情况，Liu 等提出 EasyEnsemble 和 BalanceCascade 算法[3]。EasyEnsemble 算法首先利用 Bagging 技术对多数类样本进行多次有放回随机采样，形成多个与少数类样本数量相同的多数类样本子集；接着将每个多数类样本子集与少数类样本组合，用 AdaBoost 方法训练分类器；最后将所有的多数类子集所形成的分类器再组合。BalanceCascade 算法与 EasyEnsemble 算法的原理类似，区别之处在于每一次形成多数类样本子集时，已正确分类的多数类样本将被从多数类样本集中去掉。

此外，单类学习方法是在分类时，只识别样本中的少数类，主要应用于异常检测领域。特征选择从已知的特征集合中选择出代表性特征子集，从而保留原数据的主要信息，其目的是去除冗余特征。在不均衡数据集中选出关键的区分特征，将会增强少数类和多数类的区分度，提升分类器中少数类和整体的正确率。

2　不均衡数据分类器评估指标

评价分类器性能的指标有查准率、召回率（查全率）、$F-measure$、AUC 等。对于传统分类器来说，数据集中多数类和少数类的分布大致保持均衡，分类准确率是最常用的性能评价指标。对不均衡数据集，则不能用准确率去评价一个分类器的好坏了，而常用 $G-mean$ 和 $F-measure$、AUC 作为分类器性能的评估指标。

表 2　分类结果的混淆矩阵

Tab. 2　Confusion matrix of classification results

	预测正类	预测负类
真实正类	TP（真正）	FN（假负）
真实负类	FP（假正）	TN（真负）

表 2 表达的是二类分类结果的混淆矩阵。表中 TP 和 TN 分别表示被正确预测的正类、负类样本数，FP 和 FN 则分别表示被错误预测为正类的负类样本数和被错误预测为负类的正类样本数。因 $TP+TN$ 是分类器正确预测的样本数，$FP+FN$ 则是分类错误的样本数量，$TP+TN+FP+FN$ 是所有数据样本数量。则分类准确率 Acc 可以由式（1）得出：

$$Acc=\frac{TP+TN}{TP+TN+FP+FN}. \tag{1}$$

查准率 $Precision$、召回率（查全率）$Recall$、真正率 TPR、真负率 TNR 等指标，也可由这 4 个变量，通过以下各式得到：

$$Precision=\frac{TP}{TP+FP}, \tag{2}$$

$$Recall=\frac{TP}{TP+FN}, \tag{3}$$

$$TPR=\frac{TP}{TP+FN}, \tag{4}$$

$$TNR=\frac{TN}{TN+FP}. \tag{5}$$

其中，查准率和召回率是一对矛盾的度量指标，一个指标高时，另一指标往往偏低。为实现两者之间的平衡，将其合并为一个 $F-measure$ 度量。只有当查准率和召回率都高时，$F-measure$ 的值才会大，其计算公式如下：

$$F-measure=\frac{2\times Precision\times Recall}{Precision+Recall}. \tag{6}$$

此外，采用 $G-mean$ 来衡量真正率 TPR 和真负率 TNR 之间的关系。只有当正类和负类的准确率同时都高时，$G-mean$ 值才会高，$G-mean$ 值可用来衡量不均衡分类器的整体准确率，其计算公式如下：

$$G-mean=\sqrt{TPR\times TNR}. \tag{7}$$

在不均衡数据学习中，还有一种常用的性能评价标准：受控者操纵特征曲线下面积（AUC）。受控者操纵特征曲线（ROC）显示了分类模型真正率和假正率之间的关系，是对各样本的决策输出值排序而形成的。ROC 曲线下的面积就是 AUC 测度，AUC 能很好地评价不均衡分类器的泛化性能。

$F-measure$、$G-mean$ 与 AUC 的取值范围均为 $[0,1]$，分类器性能与其值成正比，即指标值越大，分类器性能越好。

3　各种策略分析实验

各种处理不均衡数据集的方法各有优劣。在不同的应用场景下，对各种不同的数据类型，需要采用不同的处理方法。下面对常用的 3 种类不均衡分类策略：重采样方法、代价敏感学习、集成学习及其组合方法进行实验分析比较。本文试验数据来自于 KEEL 数据集（http://www.keel.es/），本文从中选取了 10 个样本数据集，见表 3。使用基于 Java 语言的开源软件 KEEL 实现了不均衡数据集的分类学习。KEEL 软件有专门的不均衡数据学习模块，集成了大部分主流的不均衡数据处理方法。实验采用 $G-mean$ 和 AUC 值作为评价不均衡分类学习能力的指标，用 $G-mean$ 值衡量分类器的准确率，AUC 值衡量分类器的泛化性能，取值越大，性能越优。

实验采用 5 折交叉验证法。实验环境具体配置为:处理器为 Intel i7-4720 2.60GHz;8G 内存;64 位 windows 操作系统。

3.1 实验方法与结果

(1)重采样方法在不均衡数据集上的分类性能比较。实验选用了过采样方法 SMOTE、欠采样方法 Tomek links 方法、混合采样方法 SMOTE_Tomek links 和 SMOTE_ENN 方法。通过重采样方法实现了数据集的再平衡之后,选用常用的决策树算法 C4.5 进行分类。各种重采样方法与 C4.5 算法的结合在不同数据集上的性能见表 4。表中的 TL 表示 Tomek links 欠采样方法,SMOTE_TL 表示 SMOTE_Tomek links 混合采样方法。

表 3 不均衡数据集基本信息

Tab. 3 Basic information about imbalanced data sets

样本集	记录个数	属性个数	不均衡比率
wisconsin	683	9	1.86
Haberman	306	3	2.78
vehicle0	846	18	3.25
yeast3	1484	8	8.1
ecoli4	336	7	15.8
glass-0-1-6_vs_5	184	9	19.44
yeast-1-2-8-9_vs_7	947	8	30.57
ecoli-0-1-3-7_vs_2-6	281	7	39.14
yeast6	1484	8	41.4
abalone19	4174	8	129.44

表 4 重采样方法性能比较

Tab. 4 Performance comparison of resample method

样本集	*G - mean* 值				*AUC* 值			
	SMOTE+C4.5	TL+C4.5	SMOTE_TL + C4.5	SMOTE_ENN + C4.5	SMOTE+C4.5	TL+C4.5	SMOTE_TL+C4.5	SMOTE_ENN+C4.5
wisconsin	0.96	0.950 9	0.949	0.945 6	0.960 1	0.951 2	0.949 1	0.945 9
Haberman	0.618 6	0.606	0.600 6	0.576 7	0.636 4	0.632 8	0.611 1	0.589 1
vehicle0	0.935 9	0.927 7	0.921 9	0.927 8	0.936 0	0.928 2	0.922 4	0.930 3
yeast3	0.903 3	0.887 4	0.89	0.912 8	0.904 4	0.890 8	0.891 0	0.913 3
ecoli4	0.950 2	0.777 7	0.861 7	0.883 5	0.951 3	0.813 9	0.869 9	0.887 0
glass-0-1-6_vs_5	0.956	0.794 2	0.851 3	0.958 9	0.957 1	0.891 4	0.865 7	0.96
yeast-1-2-8-9_vs_7	0.635 7	0.243 4	0.527	0.565 4	0.674 4	0.544 0	0.607 8	0.618 2
ecoli-0-1-3-7_vs_2-6	0.323 7	0.740 1	0.715 8	0.770 2	0.619 0	0.848 1	0.822 7	0.869 0
yeast6	0.813 4	0.757 7	0.813 6	0.822 1	0.830 6	0.795 5	0.830 6	0.836 9
abalone19	0.338 9	0	0.368 1	0.491 6	0.555 8	0.5	0.566 1	0.605 3

(2)3 种代价敏感算法在不同数据集上的性能比较。实验结果见表 5。C4.5CS 表示代价敏感决策树算法,SVMCS 表示代价敏感支持向量机算法,NNCS 表示代价敏感神经网络算法。

表 5 代价敏感方法性能比较

Tab. 5 Performance comparison of cost-sensitive learning method

样本集	*G - mean* 值			*AUC* 值		
	C4.5CS	SVMCS	NNCS	C4.5CS	SVMCS	NNCS
wisconsin	0.963 5	0.970 7	0.962 2	0.963 6	0.970 7	0.962 4
Haberman	0.500 4	0.569 8	0.557 1	0.575 2	0.619 7	0.576 5
vehicle0	0.928 5	0.962 9	0.769 8	0.928 9	0.963	0.786 2
yeast3	0.911 2	0.894 8	0.716 1	0.911 7	0.895 1	0.742 9
ecoli4	0.853 4	0.951 2	0.508 4	0.863 6	0.952 9	0.66
glass-0-1-6_vs_5	0.988 5	0.973 9	0.870 4	0.988 6	0.974 3	0.88
yeast-1-2-8-9_vs_7	0.620 6	0.707 4	0.453 3	0.676 9	0.713 1	0.507 8
ecoli-0-1-3-7_vs_2-6	0.734 6	0.768 3	0.670 9	0.828 1	0.869	0.787 3
yeast6	0.784 5	0.873	0.463 3	0.808 2	0.875 8	0.562 4
abalone19	0.307 1	0.755 7	0.304 4	0.57	0.761 5	0.410 8

(3)重采样方法 SMOTE 和 SMOTE_ENN 方法性能比较。选用经典的重采样方法 SMOTE 和 SMOTE_ENN 方法，将其与代价敏感决策树算法 C4.5CS进行组合，观察其是否比与普通决策树算法 C4.5 结合性能提升更大，结果见表 6。

表 6　重采样与代价敏感集成方法性能比较

Tab. 6　Performance comparison of ensemble learning method about resample and cost-sensitive learning

样本集	*G - mean* 值		*AUC* 值	
	SMOTE+C4.5CS	SMOTE_ENN +C4.5CS	SMOTE+C4.5CS	SMOTE_ENN +C4.5CS
wisconsin	0.957 8	0.954 0	0.957 9	0.954 2
Haberman	0.433 5	0.647 9	0.576 6	0.651 8
vehicle0	0.927 6	0.921 6	0.927 7	0.922 9
yeast3	0.903 5	0.911 8	0.904 2	0.913
ecoli4	0.945 3	0.886 9	0.946 5	0.894 9
glass-0-1-6_vs_5	0.901 2	0.849 2	0.91	0.862 9
yeast-1-2-8-9_vs_7	0.445 8	0.544 8	0.550 5	0.577 5
ecoli-0-1-3-7_vs_2-6	0.527 6	0.747 1	0.731 8	0.820 8
yeast6	0.818 3	0.822 2	0.834 2	0.832 9
abalone19	0.494 4	0.390 9	0.596 3	0.559 7

(4)经典集成方法性能比较。比较 3 种经典集成方法 SMOTEBoost、EasyEnsemble、BalanceCascade 在不同数据集上的性能，这 3 种集成方法均以 C4.5 决策树算法作为弱分类器，结果见表 7。

表 7　经典集成方法性能比较

Tab. 7　Performance comparison of classical ensemble learning method

样本集	*G - mean* 值			*AUC* 值		
	SMOTEBoost	EasyEnsemble	BalanceCascade	SMOTEBoost	EasyEnsemble	BalanceCascade
wisconsin	0.965 1	0.965 8	0.961 0	0.965 2	0.965 9	0.961 1
Haberman	0.609 7	0.657 4	0.633 1	0.614 6	0.661	0.636 8
vehicle0	0.975 9	0.951 5	0.928 2	0.976 1	0.951 8	0.928 9
yeast3	0.885 2	0.910 9	0.910 9	0.887 8	0.910 9	0.911 6
ecoli4	0.873 8	0.889 8	0.912 2	0.885 7	0.893 1	0.914 8
glass-0-1-6_vs_5	0.864 3	0.866 5	0.964 1	0.88	0.872 9	0.965 7
yeast-1-2-8-9_vs_7	0.560 2	0.659 2	0.670 2	0.651 4	0.678 4	0.672 7
ecoli-0-1-3-7_vs_2-6	0.726 3	0.682 7	0.702 7	0.831 7	0.792 2	0.799 7
yeast6	0.725 2	0.824 7	0.813 6	0.774 3	0.829 3	0.816 4
abalone19	0.236 1	0.660 2	0.615 3	0.536 6	0.671 7	0.623 7

3.2　实验结果分析

(1)重采样方法分析。根据表 4 的实验结果，过采样方法 SMOTE 在大部分数据集上的 G-mean 值和 AUC 值都高于欠采样方法 Tomek links，只有在一个数据集“ecoli-0-1-3-7_vs_2-6”上出现例外，而且随着不平衡率的增加，二者之间的差值有逐渐增大的趋势，这说明 SMOTE 方法的性能全面优于 Tomek links 方法。混合采样方面，当不均衡率小于 3 时，SMOTE_TL 采样方法的 G-mean 值和 AUC 值都高于 SMOTE_ENN 方法。不均衡率大于 3 后，SMOTE_ENN 方法的 G-mean 值和 AUC 值普遍高于 SMOTE_TL 方法，说明 SMOTE_ENN 的准确率和泛化性能优于 SMOTE_TL 方法。比较 SMOTE 和 SMOTE_ENN 这两种相对更好的方法，当不平衡率在 30 以内时，SMOTE 方法的 G-mean 值和 AUC 值高于 SMOTE_ENN 方法或是与其接近。当不平衡率超过 30 时，SMOTE_ENN 方法的 G-mean 值和 AUC 值才会高于 SMOTE 方法。

(2)代价敏感学习方法分析。根据表 5 的实验结果，代价敏感支持向量机算法 SVMCS 的 G-mean

值和 AUC 值,大多数情况下都高于另外两种代价敏感算法。在不均衡比例较高时,代价敏感决策树方法 C4.5CS 的性能与代价敏感支持向量机算法 SVMCS 的性能相差不大,在两个数据集中 C4.5CS 的准确率与泛化性能甚至超过了 SVMCS 方法。当不均衡比例超过 100 时,如在“abalone19”数据集中,SVMCS 的性能比另外两种代价敏感方法要高出很多。相比较而言,代价敏感神经网络算法的性能比另外两种算法差。

(3)重采样方法与代价敏感方法集成分析。根据表 6 的实验结果,当不平衡率小于 10 时,二种集成方法在不同数据集上所表现出的性能没有明显的规律可循。不平衡率在 10~20 时,SMOTE+C4.5CS 集成方法的性能要强于 SMOTE_ENN+C4.5CS集成方法。当不平衡率在 20~100 时则相反,SMOTE_ENN+C4.5CS 方法的性能要强于 SMOTE+C4.5CS 方法。当数据分布严重不均衡时,SMOTE+C4.5CS 方法的性能又超过了 SMOTE_ENN+C4.5CS 方法。总体而言,重采样方法与代价敏感方法的集成方法其性能表现出的规律性不强。

(4)C4.5 为基分类器的 3 种经典集成方法比较分析。根据表 7 的实验结果,当不均衡率小于 3 时,EasyEnsemble 方法的性能优于其它二种方法。不平衡率在 8~30 之间时,BalanceCascade 的性能要强于 SMOTEBoost 方法和 EasyEnsemble 方法。当不均衡率超过 40 后,EasyEnsemble 较另外两种集成方法重新取得了性能优势。当不均衡率超过 100 时,SMOTEBoost 方法的 G-mean 值明显下降,AUC 值也不如另外两种算法。

4 结束语

迄今为止,对于不均衡数据分类的理论成果非常少,本文所作的研究也只是在实验数据的基础上,总结出一些经验性的结果,迫切需要进行更深入的理论分析和研究。另外,目前研究不均衡分类问题都是基于不均衡的二分类问题,即使是不均衡的多类问题,也是通过将原问题分解成二类问题的方法去解决,并没有针对多类不均衡问题公认的评价指标。因此,需要进一步的深入研究,提出针对多类不均衡分类问题的评价指标和相应的学习算法。

参考文献

[1] CHAWLA N V, BOWYER K W, HALL L O, et al. Smote: Synthetic Minority Over-sampling Technique[J]. Journal of Artificial Intelligence Research(JAR),2002,(1):321-3357.

[2] 陶新民,郝思媛,张冬雪. 不均衡数据分类算法的综述[J]. 重庆邮电大学学报,2013, 25(1):101-110.

[3] LIU X Y, WU J, ZHOU Z H. Exploratory Undersampling for Class-Imbalance Learning[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2009, 39(2): 539-550.

[4] BATISTA G E, PRATI R C, MONARD M C. A study of the behavior of several methods for balancing machine learning training data[J]. ACM Sigkdd Explorations Newsletter, 2004.6(1):20-29.

[5] 席晓燕. 极限学习机在类不平衡学习中的应用研究[D]. 江苏:江苏科技大学,2018.

[6] 马杰. 基于 boosting 的不平衡数据分类算法研究[D]. 重庆:重庆大学,201.

[7] CHAWLA N, LAZAREVIC A, HALL L, et al. SMOTEBoost: Improving prediction of the minority class in boosting[J]. Knowledge Discovery in Databases:PKDD 2003,2003:107-119.

[8]韩家炜,Micheline Kamber,裴健. 数据挖掘概念与技术[M]. 北京:机械工业出版社,2012.250-251.

[9]孙宽宏. 不平衡数据分类方法研究[D]. 西安:西安电子科技大学,2015.

[10]曹鹏. 不平衡数据分类方法的研究[D]. 沈阳:东北大学,2014.

[11]李勇,刘战东,张海军. 不平衡数据的集成分类算法综述[J]. 计算机应用研究,2014,31(5):1287-1291.

[12]赵楠,张小芳,张利军. 不平衡数据分类研究综述[J]. 计算机科学,2018,45(6A):22-27,57.

[13]瞿云,杨炳儒,曲武. 不平衡类数据挖掘研究综述[J]. 计算机科学,2010,37(10):27-32.

[14]向鸿鑫,杨云. 不平衡数据挖掘方法综述[J]. 计算机工程与应用,2019,55(4):1-16.

[15]于文莉.面向非平衡类数据的分类器性能比较研究与方法改进[D]. 大连:大连海事大学,2017.

[16]HAIXIANG G, YIJING L, SHANG J, et al. Learning from class-imbalanced data:Review of methods and applications[J]. Expert Systems with Applications,2017,73:220-239.

[17]张立旺. 基于不平衡数据的分类方法研究[D]. 太原:中北大学,2016.

[18] LÖPEZ V. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics[J]. Information Sciences,2013.250:113-141.

[19]龙浩. 用于不平衡分类问题的自适应加权极限学习机研究[D]. 深圳:深圳大学,2017.

[20]叶志飞,文益民,吕宝粮. 不平衡分类问题研究综述[J]. 智能系统学报,2009,4(2):148-156.