INTELLIGENT AUTOMATION & SOFT COMPUTING, 2017
https://doi.org/10.1080/10798587.2017.1337673

AutoSoft®
Taylor & Francis
Taylor & Francis Group

Improving Performance Prediction on Education Data with Noise and Class Imbalance

Akram M. Radwan[a,b] and Zehra Cataltepe[a,c]

[a]Computer Engineering Department, Istanbul Technical University, Istanbul, Turkey; [b]Department of Information Technology, University College of Applied Sciences, Gaza, Palestine; [c]tazi.io Machine Learning Solutions, Istanbul, Turkey

ABSTRACT

This paper proposes to apply machine learning techniques to predict students' performance on two real-world educational data-sets. The first data-set is used to predict the response of students with autism while they learn a specific task, whereas the second one is used to predict students' failure at a secondary school. The two data-sets suffer from two major problems that can negatively impact the ability of classification models to predict the correct label; class imbalance and class noise. A series of experiments have been carried out to improve the quality of training data, and hence improve prediction results. In this paper, we propose two noise filter methods to eliminate the noisy instances from the majority class located inside the borderline area. Our methods combine the over-sampling SMOTE technique with the thresholding technique to balance the training data and choose the best boundary between classes. Then we apply a noise detection approach to identify the noisy instances. We have used the two data-sets to assess the efficacy of class-imbalance approaches as well as both proposed methods. Results for different classifiers show that, the AUC scores significantly improved when the two proposed methods combined with existing class-imbalance techniques.

KEYWORDS

Student performance prediction; imbalanced classification; SMOTE; noise filters

1. Introduction

Real-world data for many applications, including education, is never perfect. The data used to make predictions are often imbalanced and suffer from noise. Class imbalance is considered to be one of the ten challenging research problems in data mining (Yang & Wu, 2006). A data-set is said to be (class) imbalanced when one class (minority class) has much fewer instances than the remaining classes (majority classes). The minority class is usually the most interesting with respect to the domain of study; hence high prediction of minority class is our target. The traditional classification algorithms are designed to maximize the overall accuracy, which is independent of class distribution. Thus, when learning from imbalanced data, they are usually overwhelmed by the majority class instances and ignore the minority class (Thai-Nghe, Gantner, & Schmidt-Thieme, 2010). Consequently, the instances that belong to the minority class are misclassified more often than those belonging to the majority class. A number of approaches have been proposed to handle the problem of imbalanced classification, both for standard learning algorithms and for ensemble techniques (Guo, Yin, Dong, Yang, & Zhou, 2008; López, Fernández, García, Palade, & Herrera, 2013; Sáez, Luengo, Stefanowski, & Herrera, 2015; Sluban, Gamberger, & Lavrač, 2014).

The term "noise" refers to data points, which could be considered as erroneous, irrelevant or meaningless. Noise is often divided into two categories (Wu & Zhu, 2008; Zhu, Wu, & Chen, 2003); (a) attribute noise (errors or missing values in one or more attributes); and (b) class noise, which can be found in the following forms: (1) Contradictory instances; instances with the same values of attributes but with different labels. (2) Misclassifications; instances with wrong class labels (Zhu et al., 2003).

Regardless of the type of noise, existence of noisy instances in a data-set has a negative effect on the quality of information retrieved from the data, models built using this data, and decisions made based on the analysis of those models (Zhu & Wu, 2004). Noise filtering approaches are often used in machine learning (ML) to detect and eliminate noisy instances from training data, and consequently improve the classification accuracy of models induced from the clean data (Anyfantis, Karagiannopoulos, Kotsiantis, & Pintelas, 2007; Khoshgoftaar & Rebours, 2007).

Borderline instances are located in the area surrounding decision boundary separating classes. In order to achieve better prediction, most of the classification algorithms attempt to learn the borderline instances of each class during the training process (Sáez et al., 2014; Sáez et al., 2015; Seiffert, Khoshgoftaar, Van Hulse, & Folleco, 2014). Napierała, Stefanowski, and Wilk (2010) showed that a large number of borderline instances negatively affected the performance of a classifier. These instances are more likely to be misclassified than the ones far from the class boundary, and hence detecting and eliminating noisy instances within borderline area increases the chances of achieving more effective classification (Napierała et al., 2010; Sáez et al., 2015). In this study, we consider only noisy instances restricted within the area surrounding class boundaries. Thus the noisy instances, located inside the borderline area, from the majority class are removed and the minority class remains unchanged. The idea behind this is that since there are so scarce, instances from the minority class are important and should not be eliminated by the noise elimination procedures.

CONTACT Akram M. Radwan ✉ aradwan@ucas.edu.ps

The rest of the paper is organized as follows: In Section 2, we describe several approaches, which have been used to handle the class imbalance problem, and discuss the classifier evaluation metrics. In Section 3, we review the state of the art on noise filter techniques. Section 4 explores the related work and background about predicting student's performance using ML methods. Next, Section 5 proposes two empirical methods that address noise filtering and class imbalance problems simultaneously. Section 6 describes the data-sets and shows the experiments carried out and the results obtained. Finally, in Section 7, we summarize the main conclusions and future work.

2. Classification of Imbalanced Data-sets

2.1. Methods for Dealing with Class Imbalance

To deal with class imbalance problem (Guo et al., 2008; López et al., 2013), a number of approaches have been proposed. Here, we summarize some techniques that are commonly used to tackle the class imbalance problem.

2.1.1. Data Re-sampling Methods

Re-sampling methods aim to balance the class distribution in the training data by either duplicating or generate new minority instances (over-sampling) or removing instances from the majority class (under-sampling) (Blagus & Lusa, 2013; Thai-Nghe et al., 2010). Several techniques for performing over-sampling and under-sampling have been proposed. Both under-sampling and over-sampling have their benefits and drawbacks (Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2010). While under-sampling causes loss of information that comes from deleting instances, over-sampling could lead to over-fitting that comes from duplicating instances or creating new ones.

Random over-sampling (ROS) (He & Garcia, 2009; Thai-Nghe et al., 2010) method is used to balance class distribution by randomly duplicating the minority class instances while training the classifier until a desired class ratio is achieved. Sample weight parameter is used in the fit method (Batista, Prati, & Monard, 2004). Although ROS is effective, it may increase the likelihood of over-fitting since it makes exact copies of the minority class instances (Guo et al., 2008).

The Synthetic Minority Oversampling Technique (SMOTE), introduced by Chawla, Bowyer, Hall, & Kegelmeyer (2002), is one of the most well-known and widely used re-sampling methods. SMOTE generates new artificial minority instances by interpolating among the existing minority instances. This method first finds the k nearest neighbors of each minority instance; next, it selects a random nearest neighbor. Then a new minority class instance is created along the line segment joining a minority class instance and its nearest neighbor. This procedure is repeated until both classes have equal number of instances (Chawla et al., 2002).

2.1.2. Cost-sensitive Learning

Standard classifiers assume that the misclassification costs (false negative and false positive cost) are the same for all classes. However, in most real-world applications, this assumption is not true. When learning from imbalanced data, the classifier tends to be biased towards the majority class. Thus we need to assign a high cost to misclassification of the minority class, and try to minimize the overall cost. In the weighting method (Ting, 1998), we assign a certain weight to each instance in terms of its class, according to the misclassification costs, with the minority class given larger weight. Classes with higher weights are given more importance while training the classifier.

2.1.3. Thresholding

Some classifiers can produce probability estimates on instances. However, when the classes in the training data are imbalanced, these predictions calculated by the classifier can be inaccurate, because many classifiers do not know how to adjust for the class imbalance. If a classifier's probabilities are accurate, the appropriate way to convert its probabilities into predictions is to cut-off at a threshold (usually 0.5) and predict the positive class if the probability is above the cut-off and otherwise the negative class. When the probabilities are inaccurate, this method does not work well. We can improve the predictions by adjusting the threshold to a value that minimizes the total misclassification cost on the training instances, and use this value to predict the class label of test instances (Seiffert et al., 2010; Sheng & Ling, 2006).

2.2. Evaluation Metrics in Imbalanced Domains

Evaluation measures are needed so that different classifiers' performances can be compared to each other. For a two-class problem, the confusion matrix, shown in Table 1, illustrates the distribution of correct and incorrect instances for the positive and negative classes. TP and TN denote the number of positive and negative instances that are classified correctly, while FN and FP denote the number of misclassified positive and negative instances respectively.

A number of widely used metrics to measure the performance of the models can be computed based on the confusion matrix. Predictive accuracy (ACC) is the performance measure generally associated with classification algorithms and is calculated as:

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

Typically, the accuracy is the most commonly used empirical measure, however, since it does not distinguish between the number of correctly classified instances of different classes, accuracy is no longer a proper measure when the data is imbalanced. The ROC (Receiver Operating Characteristics) curve is a standard technique for summarizing classifier performance on imbalanced data-sets (Fawcett, 2006). The ROC is a plot of the TP_{rate}(sensitivity) against the FP_{rate}(1-specificity), where $TP_{rate} = \frac{TP}{TP+FN}$and, $FP_{rate} = \frac{FP}{FP+TN}$for different threshold values characterizing the overall performance of a classifier. The area under the ROC curve (AUC) is the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). AUC values range from 0 to 1, with a higher AUC value indicating that the classifier has a higher discriminative capability to differentiate positive samples from negative instances. An AUC equal to 1.0 indicates a perfect classifier, whereas 0.5 indicates that a model performs like a random classifier (Berrar & Flach, 2011). AUC has been shown to be a reliable performance measure for imbalanced data-sets (García et al., 2012; López et al., 2013;

Table 1. Confusion Matrix for a Two-class Problem.

	Predicted Positive	Predicted Negative
Actual positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

Sáez et al., 2015). Thus we have used AUC metric to evaluate our experiments in Section 6.

3. Noise Filter Approaches

Noise handling is a central focus in many areas of ML. The preliminary approaches to noise treatment aim to create inductive learning algorithms that are able to resist the data-set's noise. However, the most common approaches to noise handling is to eliminate noise by filtering of noisy instances before the model is built. Noise filters are preprocessing mechanisms designed to enhance the data quality by eliminating noisy instances in the training set (Zhu et al., 2003). The separation of noise detection has the advantage that noisy instances do not influence the classifier design. Then a classifier is used on the reduced and clean training data.

There are numerous noise detection approaches in literature. The most widely used are; Classification Filter (CF) proposed by Gamberger, Boskovic, Lavrac, and Groselj (1999), Ensemble Filter (EF) (Khoshgoftaar, Joshi, & Seliya, 2006), and Iterative-Partitioning Filter (IPF) (Zhu et al., 2003). CF approach identifies the noisy instances using cross validation. The training set is partitioned into n parts, and then n different classifiers are trained using collection of any n-1 parts to identify noise from the excluded part. On the other hand, the EF approach learns a classifier from each single part and then the classifier is evaluated on the whole data-set. To identify noisy instances, CF directly adds the misclassified instances by one classifier to the noise filtered set, but EF takes the majority or consensus vote of a committee of n classifiers (Zhu et al., 2003). IPF is a preprocessing technique based on the EF. It removes noisy instances in multiple iterations until the number of identified noisy instances in each of these iterations is less than a certain percentage of the size of the original training data (Sáez et al., 2015).

4. Related Work

Recently, researchers in the educational community have been interested in applying ML techniques for predicting student performance. Class imbalance problem was one of the obstacles that caused unsatisfactory prediction results. Thai-Nghe, Busche, and Schmidt-Thieme (2009) proposed a model to improve the student academic performance prediction by dealing with the class imbalance problem. They first re-balanced the data-set by SMOTE and then used cost-insensitive learning to minimize the misclassification cost. The model was examined on four data-sets, and the results improved compared to the baseline classifiers. Márquez-Vera, Cano, Romero, and Ventura (2013) devised a genetic programming algorithm with a data balancing approach for solving the problem of student failure using real data about 670 first-year high school students in Mexico. In a more recent study, an ensemble filtering approach has been used in Satyanarayana and Nuckowski (2016) to enhance the quality of students' data by eliminating noisy instances. They showed that, compared to single filters, using ensemble filters gives better predictive accuracies.

With reference to the original SMOTE technique, several adaptations have been proposed in the literature, most of them aim at identifying the region in which the minority instances should be generated. Another extension of SMOTE corresponds to the Borderline-SMOTE technique (Han, Wang, & Mao, 2005), which only over-sampled the minority instances near the borderline since these are more likely to be misclassified. This method achieves better TP rate and F-value than SMOTE and random over-sampling methods. Batista et al. (2004) proposed two data cleaning methods to the over-sampled training set by SMOTE. In the first method, SMOTE-TL uses Tomek links to remove instances after applying SMOTE (Seiffert et al., 2010). An instance is removed either, because it is noisy or because it is near the border. Tomek links (Tomek, 1976) are defined on pairs of minimally distanced nearest neighbors of opposite classes. The second method SMOTE-ENN (SMOTE with Edited Nearest Neighbor) tends to remove more instances (from both classes) than the SMOTE-TL does, so it is expected to provide a more effective data cleaning. ENN removes from the training set any instance that differs from two of its three nearest neighbors (Batista et al., 2004). The two methods SMOTE-TL and SMOTE-ENN are characterized by their over-sampling followed by under-sampling.

Researchers have used ensemble methods (Dietterich, 2000) to deal with the problem of noise filtering. Consensus and majority are the two voting schemes that could be implemented to identify noisy instances. The former eliminates an instance if it is misclassified by all the classifiers, while the latter eliminates an instance if it is misclassified by more than half of the classifiers (Sáez et al., 2015). There are many ensemble-based noise filters. For example, Brodley and Friedl (1999) used consensus filters and majority vote filters to identify and eliminate mislabeled training instances, which are incorrectly classified by the multiple classifiers. Their results show that if that the training data-set is sufficiently large, then classification accuracy can be improved as more noisy instanced were removed. Sluban et al. (2014) presented an ensemble-based methodology for explicit noise detection and ranking, called NoiseRank. It can rank the detected noisy instances according to the predictions of several different noise detection algorithms and thus it provided more reliable results.

Seiffert et al. (2014) performed a comprehensive and empirical study on the effects of class imbalance and class noise on 11 different classification algorithms and data sampling techniques when they used to predict software quality. They compared the performance of seven sampling techniques using 12 data-sets derived from real world software quality data with different levels of class noise and imbalance. Later, Sáez et al. (2015) proposed and examined a new extension of SMOTE through an IPF filter, called SMOTE-IPF, which can overcome the problems introduced by noisy and borderline instances in imbalanced data-sets. The results show that SMOTE–IPF performed better than existing SMOTE generalizations with both synthetic and real-world data-sets.

5. Proposed Methods

We present two empirical methods that address noise filtering and class imbalance problems simultaneously; the first is class-Balanced by SMOTE & Thresholding combined with Classification Filter, called BST-CF, and the second is class-Balanced by SMOTE & Thresholding combined with Ensemble Filter, called BST-EF. Both methods eliminate noisy instances, located inside the borderline area, from the majority class and the minority class remains unchanged. Our methods incorporate a noise filtering into two class-imbalance techniques SMOTE and thresholding, which are used to balance the class distribution of the training data and choose the best boundary between classes. The CF approach is used to identify noisy instances in the first method while the EF approach is used in the second. The main differences between our methods and other noise filters are:

- The noisy instances are refined and restricted within borderline area of majority class.
- To the best of our knowledge, thresholding together with noise filtering approach has not been used before.

In order to perform a fair comparison, random forest (RF) (Breiman, 2001) is deployed as the base classifier in both methods, because it runs efficiently on large data, and it provides methods for balancing error in class population imbalanced data-sets (Breiman, 2001; Truong, Lin, & Beecher, 2004). Moreover, our methods need to find the probabilities that assigned to each instance in a subset of training data; RF can easily estimate probability on instances using scikit-learn.

Our proposed method BST-CF is described as follows:

(1) Grid search with cross validation are used to determine the optimal hyper-parameters for RF model in terms of AUC.
(2) Split the whole data-set X into five subsets, four of them are selected as training data X_{train}, and the fifth subset is used as the test data X_{test} to evaluate the performance of RF model on the clean data.
(3) Initially, the method starts with a set of noisy instances $S = \varnothing$.
(4) Use stratified 10-fold cross validation to divide X_{train} into fit set X_{fit} (aggregation of any 9 subsets) and validation set X_{val} (excluded subset) such that they are randomly sampled, and classes are equally balanced in both.
(5) Firstly, the over-sampling technique SMOTE is applied to balance X_{fit}, then the RF is built on this set.
(6) Investigate the threshold value θ^*_{fit} in range from 0.40 to 0.80 that gives the best average AUC score when converting probabilities on X_{fit} into classes by the RF classifier using cross validation.
(7) Generate the predicted probabilities that the RF classifier assigned to each instance in validation set X_{val}.
(8) Find out the borderline instances in X_{val}, i.e., instances around the chosen threshold θ^*_{fit} with a certain β width.
(9) Add to set S the noise, which are incorrectly classified instances within the borderline area and belonging to the majority class.
(10) Repeat steps 4–9 on the other folds; and then eliminate the noisy set S.
(11) Re-fit the RF model on the filtered fit set X'_{train}.
(12) Find the optimal threshold value θ^*_{val} by taking the mean of θ^*_{fit} of the ten runs.
(13) The final RF model generated in step 11 is then used to predict the class label of test data X_{test} using threshold θ^*_{val}.

Evaluation measures are estimated in 10-fold cross validation repeated 5 times and the results are obtained by averaging scores of the 5 runs. Figure 1 illustrates the overall design of our methods.

The second method BST-EF differs from the BST-CF method in the following:

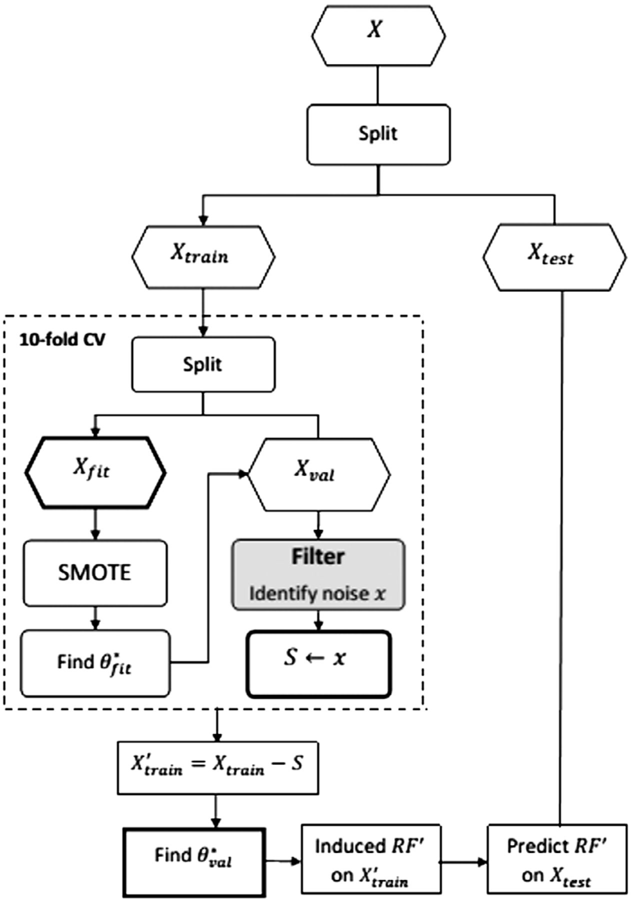

Figure 1. The Overall Design of BST-CF and BST-EF. The Steps that Differ between Two Methods are Shown using a Thick Border.

(1) The fit set X_{fit} consists of only one subset while the validation set X_{val} is a union of the remaining 9 subsets (step 4 above).
(2) When identifying a noisy instance, BST-CF adds it directly to S set at each run, but BST-EF computes the number of times that an instance is identified as a noise, then eliminates the noisy instances with high scores; such that the total number of eliminated noisy instances in S equals the average number of noisy instances identified at each of the ten runs (step 9).
(3) The optimal threshold θ^*_{val} is found by cross validation on the set X'_{fit} (step 12).

6. Experiments and Results

In this section, we first present the two real-world data-sets, then we describe a series of experiments that handle class imbalance and class noise problems. We then present the test results of our methods on the two data-sets, and finally discuss the results. All experiments have been conducted using scikit-learn package. Scikit-learn (Pedregosa et al., 2011) is a general purpose ML library written in Python, which provides efficient implementations of many ML algorithms. All classifier' performances were evaluated through stratified 10-fold cross validation method.

6.1. Performance of Students with Autism Data-set

The Autism data-set has been gathered from the web application during learning sessions of five students with autism whilst

Table 2. Description of the Autism Data-set.

Attribute	Description	Values	Type
student_id	the student ID	numeric: from 1 to 5	predictor
object_id	the object ID used in a trial	numeric: from 1 to 30	predictor
category_id	the category ID of object	numeric: from 1 to 5	predictor
level	difficulty level of the object	numeric: from 1 to 4	predictor
RT	the response time	numeric: real	predictor
RC	the response by the student	binary: 1 or 0	output

they have been teaching object recognition over the period from April to June 2015 (Radwan, Birkan, Hania, & Cataltepe, 2017). Students' ages ranged from 5 to 9 years. Each student performed 20 sessions; each contains 30 to 40 assessment trials. The total number of instances (rows) used in this study is 3090. Each instance represents the results of an assessment trial for a particular student during the conduction of the investigation. The attributes/variables of the Autism data-set are shown in Table 2. Since they contain discrete and unordered values, 1-of-K representation was used for the following attributes: student_id, object_id, and category_id. After applying this representation, the total number of features was 42.

Since we want to predict the RC, we have a binary classification problem. As 85.6% instances are labeled with 1 (correct) and the remaining 14.4% of instances are labeled with 0 (incorrect), this is an imbalanced data-set and the imbalance ratio (IR) of 1 to 0 instances is 5.94. In this paper (and without loss of generality), the minority class (class 0) is regarded as the positive class, and the majority class (class 1) is regarded as the negative class.

6.1.1. Experiment 1—Class Imbalance

In the first set of experiments, we analyzed the behavior of class imbalance approaches. Our aim is to investigate the improvement of classifier's result when using five different approaches: ROS, SMOTE, SMOTE-TL, SMOTE-ENN and weighting. Four base ML classification algorithms were used: Logistic Regression (LR), Random Forest classifier, linear Support vector machine (SVM) and AdaBoost. The RF involves an ensemble of decision trees grown based a randomly selected subset of samples and features. The prediction is made by aggregating majority vote of the ensemble. For the RF model, the number of trees in the forest was empirically chosen. We performed a grid search of the n_estimators model parameter, evaluating a series of values from 20 to 320 with a step size of 30. The best number of trees was n_estimators = 200 that can be used to build more accurate RF classifiers. AdaBoost is an ensemble classifier (Dietterich, 2000), which follows a boosting technique that combines multiple weak classifiers into a strong one. AdaBoost was used with LR model.

Table 3 presents the overall test AUC results (± standard deviation) obtained by different classifiers using class-imbalance approaches considered in this paper. The row denoted by "None" corresponds to the case in which no class-imbalance approach is performed. From results of Table 3, the following main points should be stressed:

(1) All classifiers in "None" case have the test AUC score of about 0.5, which means that they perform similar to a random classifier.
(2) Classification with balanced data-set results in effective performance, regardless of any used classifier.
(3) All approaches have been able to improve the AUC results. However, SMOTE-ENN and weighting are quite robust on average among those approaches for all classifiers.

6.1.2. Experiment 2—Class Imbalance and Noise Filter

The goal of experiment 2 is to investigate the performance of the two proposed methods for different classifiers. The methods BST-CF and BST-EF were used to produce the optimal noise set to be removed from the Autism data-set at different widths β (from 0 to 0.6). The number of noisy instances detected by BST-CF method was 350 (11.3%), and by BST-EF method was 320 (10.3%). The experimental results shown in Figure 2 are presented in terms of the average AUC via 10 runs for different four classifiers after removing noisy instances from the Autism data-set using BST-CF and BST-EF methods. Figure 2 reveals the following conclusions. First, the AUC scores obtained by classifier using class-imbalance approaches after removing noise from the Autism data-set are better than the AUC results obtained by a classifier with only class-imbalance approaches. For instance, the results for RF with SMOTE increase from 0.593 to 0.673 when using our methods. Secondly, the best AUC's scores (0.749 for BST-CF and 0.74 for BST-EF) are achieved by RF classifier. Third, the AUC results of BST-CF are slightly better than BST-EF. The reason may be due to BST-CF eliminates more noisy instances.

6.2. UCI Portuguese Student's Failure Prediction Data-set

This UCI repository[1] data-set called Portuguese is based on a study of data collected during the 2005–2006 school year from two public schools at Alentejo region in Portugal (Cortez & Silva, 2008). The data-set is the Portuguese language class performance of secondary school students. The data-set contains 649 instances; each representing a secondary school student. Each instance has 32 attributes including student grades, demographic, social and school related features. Attributes of the data-set are shown in Table 4. During the school year, students are evaluated in two periods (G1, G2) and the last evaluation

Table 3. Comparison of Different Classifiers on a Autism Data-set in Terms of AUC.

Algorithm	LR	RF	SVM	AdaBoost
None	0.514 ± .015	0.509 ± .010	0.5 ± 0.0	0.536 ± .024
ROS	0.643 ± 0.067	0.631 ± 0.042	0.649 ± 0.054	0.638 ± 0.047
SMOTE	0.682 ± 0.038	0.593 ± 0.030	0.679 ± 0.047	0.678 ± 0.034
SMOTE-TL	0.687 ± 0.042	0.602 ± 0.039	0.684 ± 0.044	0.684 ± 0.039
SMOTE-ENN	0.690 ± 0.027	0.677 ± 0.044	0.695 ± 0.035	0.686 ± 0.026
Weighting	0.690 ± 0.039	0.682 ± 0.043	0.681 ± 0.041	0.678 ± 0.028

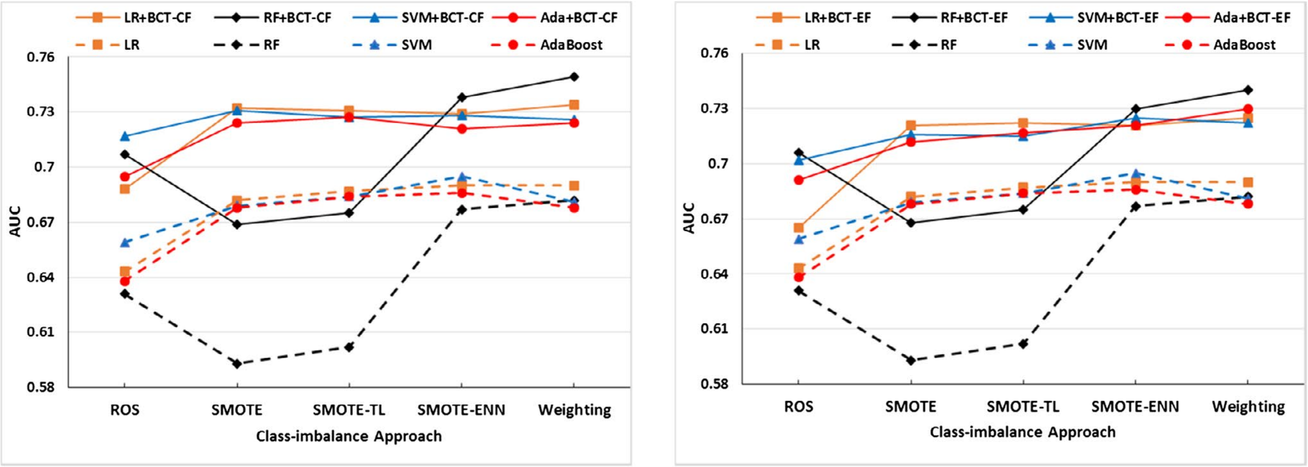

Figure 2. Average AUC Obtained by Different Classifiers using only Class-imbalance Approaches (Dashed Line) and Class-imbalance Approaches with One of Proposed Methods (Solid Line) on Autism Data-set. (Left) Using BST-CF Method. (Right) Using BST-EF Method.

Table 4. Attributes of the Portuguese Data-set.

Attribute	Description	Values
school	student's school	binary: GP or MS
sex	student's sex	binary: female or male
age	student's age	numeric: from 15 to 22
address	student's home address type	binary: urban or rural
famsize	family size	binary: ≤ 3 or > 3
Pstatus	parent's cohabitation status	binary: living together or apart
Medu	mother's education	numeric: from 0 to 4
Fedu	father's education	numeric: from 0 to 4
Mjob	mother's job	nominal: teacher, health, services, at_home or other
Fjob	father's job	nominal: teacher, health, services, at_home or other
reason	reason to choose this school	nominal: close to home, school reputation, course preference or other
guardian	student's guardian	nominal: mother, father or other
traveltime	home to school travel time	< 15 min, 15 to 30 min, 30 min to 1 h, or >1 h
studytime	weekly study time	< 2 h, 2 to 5 h, 5 to 10 h, or >10 h
failures	number of past class failures	numeric: n if 1<=n<3, else 4
schoolsup	extra educational support	binary: yes or no
famsup	family educational support	binary: yes or no
paid	extra paid classes within the course	binary: yes or no
activities	extra-curricular activities	binary: yes or no
nursery	attended nursery school	binary: yes or no
higher	wants to take higher education	binary: yes or no
internet	Internet access at home	binary: yes or no
romantic	with a romantic relationship	binary: yes or no
famrel	quality of family relationships	numeric: from 1 - very bad to 5 - excellent
freetime	free time after school	numeric: from 1 - very low to 5 - very high
goout	going out with friends	numeric: from 1 - very low to 5 - very high
Dalc	workday alcohol consumption	numeric: from 1 - very low to 5 - very high
Walc	weekend alcohol consumption	numeric: from 1 - very low to 5 - very high
health	current health status	numeric: from 1 - very bad to 5 - very good
absences	number of school absences	numeric: from 0 to 93
G1	first period grade	numeric: from 0 to 20
G2	second period grade	numeric: from 0 to 20
G3	final grade (output target)	numeric: from 0 to 20

(G3) corresponds to the final grade. In this work, the grade G3 has been modelled using supervised approach as binary classification where a student passes if G3 ≥ 10 else fails. The majority of students (549) passed and a minority (100) failed thus the data-set is imbalanced and the IR ration of pass to fail instances is 5.49. The nominal attributes were transformed into a 1-of-K encoding. The G3 attribute was transformed into 1 (pass) and 0 (fail).

We replicated the above experiment 1 on the Portuguese data-set. The same classifiers were used here. Since AdaBoost with LR using different class-imbalance approaches achieved test AUC scores of about 0.53, which is very low, so we changed Adaboost to be with decision tree (DT). Table 5 presents the average test AUC obtained by different classifiers using five class-imbalance approaches. When class-imbalance approaches except SMOTE-ENN are used, AUC scores improve compared with "None" case regardless of any used classifier. In most cases, SVM outperforms the others classifiers. The results indicate that the weighting is robust on average among the approaches, and the worst results are obtained by SMOTE-ENN.

We also replicated the above experiment 2 on the Portuguese data-set. The two methods BST-CF and BST-EF were used to produce the best noise set to be removed from Portuguese data-set. The number of noisy instances detected by BST-CF method was 54 (8.32%), and by BST-EF method was 49 (7.55%). The experimental results, shown in Figure 3,

Table 5. Comparison of Different Classifiers on a Portuguese Data-set in Terms of AUC.

Algorithm	LR	RF	SVM	AdaBoost
None	0.774 ± 0.059	0.805 ± 0.048	0.824 ± 0.069	0.818 ± 0.086
ROS	0.793 ± 0.055	0.868 ± 0.039	0.893 ± 0.029	0.831 ± 0.075
SMOTE	0.779 ± 0.061	0.801 ± 0.047	0.830 ± 0.054	0.820 ± 0.031
SMOTE-TL	0.802 ± 0.062	0.815 ± 0.042	0.828 ± 0.071	0.815 ± 0.093
SMOTE-ENN	0.712 ± 0.075	0.729 ± 0.070	0.702 ± 0.075	0.747 ± 0.089
Weighting	0.865 ± 0.052	0.889 ± 0.036	0.902 ± 0.026	0.835 ± 0.045

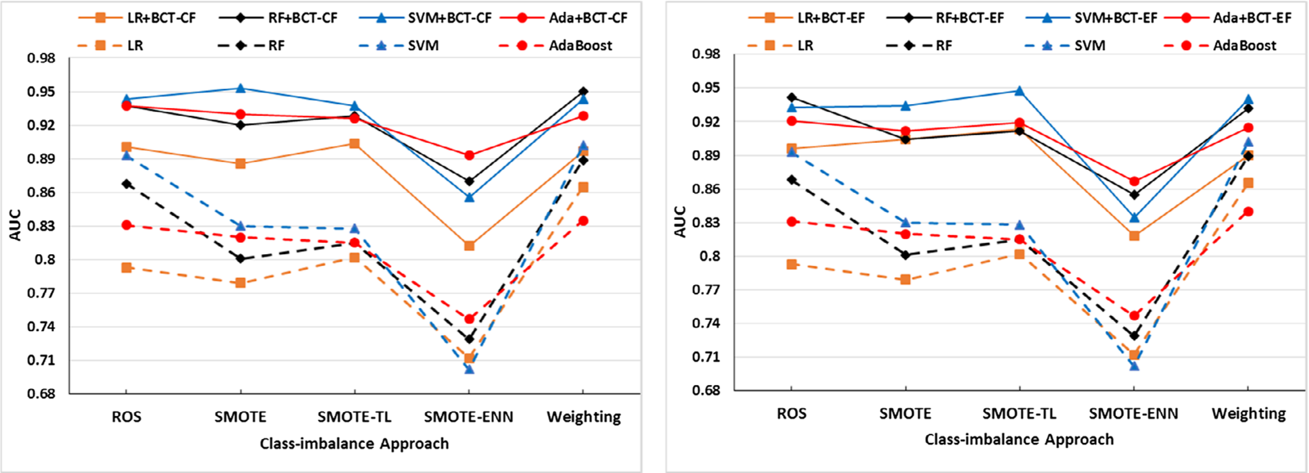

Figure 3. Average AUC Obtained by Different Classifiers using only Class-imbalance Approaches (Dashed Line) and Class-imbalance Approaches with One of Proposed Methods (Solid Line) on Portuguese Data-set. (Left) Using BST-CF Method. (Right) Using BST-EF Method.

are presented in terms of the average AUC for different four classifiers after removing noisy instances from Portuguese data-set using BST-CF and BST-EF methods (solid line) compared with results obtained without noise removal (dashed line). The classifiers' performance always improves with noise elimination. With all five class-imbalance approaches, BST-CF and BST-EF methods can contribute from 3% to over 15% to the AUC performance improvement, varying from classifiers and approaches. Comparing the results in Table 5, the AUC scores for RF and SVM with SMOTE increase from 0.801 and 0.83 to 0.92 and 0.953 respectively when using the BST-CF method. In terms of AUC, the SVM is the best choice in 8 cases (out of 12), followed by the RF, which obtains 4 best results. One can also find that the least AUC values obtained with SMOTE-ENN using two methods.

7. Conclusion and Future Work

Despite the increasing number of class noise filtering techniques, these have been slightly used in the context of learning from the imbalanced data (Sáez et al., 2015). According to our best knowledge, class noises together with class imbalance have not yet been used on an educational domain data-set.

This paper proposes two noise filter methods, BST-CF and BST-EF that deal with class imbalance and noise filtering problems simultaneously. Our experimental results on the two imbalanced Autism and Portuguese data-sets show that the four classification algorithms have a notably better performance when combining our methods with class-imbalance approaches. The most accurate classifier tested over our methods is the random forest on Autism data-set and SVM on the Portuguese data-set.

In the future, we aim to carry out the proposed methods on data-sets outside the education domain, and examine the effectiveness of them. We will investigate the use of other noise filter approaches such as IPF (Zhu et al., 2003), saturation filter or NoiseRank (Sluban et al., 2014) using the same procedures employed in our methods.

Note

1. http://archive.ics.uci.edu/ml

Notes on contributors

Akram Radwan received a Ph.D. in Computer Engineering from Istanbul Technical University, Turkey in 2016. He is currently a lecturer at University College of Applied Sciences (UCAS) in Gaza, Palestine. His research interests include machine learning, feature selection, noise filter, imbalanced data classification and machine learning for human learning.

Zehra Cataltepe is a professor at the Istanbul Technical University and also the co-founder of tazi.io, which is a startup company with the purpose of developing online machine learning products. She received her M.Sc. and Ph.D. degrees from Caltech in Computer Science. Her research interests include machine learning algorithms and software, feature selection, anomaly detection, online machine learning, machine learning for human learning, learning on heterogeneous, networked and time series data, and big data. She has both industry and academia experience and has taken part in EU and Tubitak projects as a referee, researcher and principal investigator. She has 14 patents and over 80 publications.

References

Anyfantis, D., Karagiannopoulos, M., Kotsiantis, S., & Pintelas, P. (2007). Robustness of learning techniques in handling class noise in imbalanced datasets. In *Artificial intelligence and innovations 2007: From theory to applications* Proc. IFIP Int. Federation Inform. Process., vol. 247. (pp. 21–28). Springer.

Batista, G.E.A.P.A., Prati, R.C., & Monard, M.C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations Newsletter, 6*, 20–29.

Berrar, D., & Flach, P. (2011). Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them). *Briefings in bioinformatics, 13*, 83–97.

Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, *14*(1): 1–16.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Brodley, C.E., & Friedl, M.A. (1999). Identifying mislabelled training data. *Journal of Artificial Intelligence Research, 11*, 131–167.

Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 341–378.

Cortez, P., & Silva, A. (2008). Using data mining to predict secondary school student performance. In *Proceedings of 5th Annual Future Business Technology Conference* (pp. 5–12). Porto, Portugal.

Dietterich, T.G. (2000). Ensemble Methods in Machine Learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems* (pp. 1–15). London, UK.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*, 861–874.

Gamberger, D., Boskovic, R., Lavrac, N., & Groselj, C. (1999). Experiments with noise filtering in a medical domain. In *Proc. of 16th ICML* (pp. 143–151), San Francisco, CA.

García, V., Sánchez, J. S., Martín-Félez, R., & Mollineda, R. A. (2012). Surrounding neighborhood-based SMOTE for learning from imbalanced data sets. *Progress in Artificial Intelligence, 1*(4), 347–362.

Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the class imbalance problem. *Fourth International Conference on Natural Computation IEEE, 4*, 192–201.

Han, H., Wang, W.Y., & Mao, B.H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In D.-S. Huang, X.P. Zhang, & G.-B. Huang (Eds.), *Advances in intelligent computing* (pp. 878–887). Springer Berlin Heidelberg.

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284.

Khoshgoftaar, T.M., & Rebours, P. (2007). Improving software quality prediction by noise filtering techniques. *Journal of Computer Science and Technology, 223*, 387–396.

Khoshgoftaar, T.M., Joshi, V., & Seliya, N. (2006). Detecting noisy instances with the ensemble filter: A study in software quality estimation. *International Journal of Software Engineering and Knowledge Engineering, 16*, 53–76.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences, 250*, 113–141.

Márquez-Vera, C., Cano, A., Romero, C., & Ventura, S. (2013). Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Applied Intelligence, 38*, 315–330.

Napierała, K., Stefanowski, J., & Wilk, S. (2010). Learning from imbalanced data in presence of noisy and borderline examples. In M. Szczuka, M. Kryszkiewicz, S. Ramanna, R. Jensen, & Q. Hu (Eds.), *Rough sets and current trends in computing* (pp. 158–167). Springer Berlin Heidelberg.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825–2830.

Radwan, A.M., Birkan, B., Hania, F., & Cataltepe, Z. (2017). Active machine learning framework for teaching object recognition skills to children with autism. *International Journal of Developmental Disabilities, 63*, 158–169. doi:10.1080/20473869.2016.1190543

Sáez, J. A., Luengo, J., Stefanowski, J., & Herrera, F. (2014). Managing borderline and noisy examples in imbalanced classification by combining SMOTE with ensemble filtering. In E. Corchado, J. A. Lozano, H. Quintián, & H. Yin (Eds.), *Intelligent Data Engineering and Automated Learning-IDEAL 2014* (pp. 61–68). Berlin, Heidelberg: Springer.

Sáez, J.A., Luengo, J., Stefanowski, J., & Herrera, F. (2015). SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences, 291*, 184–203.

Satyanarayana, A., & Nuckowski, M. (2016). Data mining using ensemble classifiers for improved prediction of student academic performance. *ASEE Mid-Atlantic Section Spring 2016 Conference*, Washington, DC.

Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., & Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 40*, 185–197.

Seiffert, C., Khoshgoftaar, T.M., Van Hulse, J., & Folleco, A. (2014). An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences, 259*, 571–595.

Sheng, V.S., & Ling, C.X. (2006). Thresholding for making classifiers cost-sensitive. In *Proceedings of the 21st National Conference on Artificial Intelligence* (pp. 476–481). Boston, MA: Massachusetts.

Sluban, B., Gamberger, D., & Lavrač, N. (2014). Ensemble-based noise detection: Noise ranking and visual performance evaluation. *Data Mining and Knowledge Discovery, 28*, 265–303.

Thai-Nghe, N., Busche, A., & Schmidt-Thieme, L. (2009). Improving academic performance prediction by dealing with class imbalance. *Ninth International Conference on Intelligent Systems Design and Applications, (ISDA 2009)* (pp. 878–883). Pisa, Italy: IEEE Computer Society.

Thai-Nghe, N., Gantner, Z., & Schmidt-Thieme, L. (2010). Cost-sensitive learning methods for imbalanced data. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8), Barcelona, Spain, 2010.

Ting, K.M. (1998). Inducing cost-sensitive trees via instance weighting. In J.M. Żytkow & M. Quafafou (Eds.), *Principles of data mining and knowledge discovery* (pp. 139–147). Heidelberg: Springer, Berlin.

Tomek, I. (1976). Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-6, 11*, 769–772.

Truong, Y., Lin, X., & Beecher, C. (2004). Learning a complex metabolomic data set using random forests and support vector machines. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp 835–840). New York, NY: ACM.

Wu, X., & Zhu, X. (2008). Mining with noise knowledge: error-aware data mining. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 38*, 917–932.

Yang, Q., & Wu, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making, 5*, 597–604.

Zhu, X., & Wu, X. (2004). Class noise vs. Attribute noise: A quantitative study. *Artificial Intelligence Review, 22*, 177–210.

Zhu, X., Wu, X., & Chen, Q. (2003). Eliminating class noise in large datasets. In *Proceedings of the 20th ICML* (pp. 920-927), Washington, DC.