A Double-Ensemble Approach for Classifying Skewed Data Streams

Chongsheng Zhang[1] and Paolo Soda[2]

[1] School of Computer and Information Engineering, HeNan University, China
chongsheng.zhang@yahoo.com
[2] Integrated Research Centre, Università Campus Bio-Medico di Roma, Italy
p.soda@unicampus.it

Abstract. Nowadays, many applications need to handle large amounts of streaming data, which often presents a skewed distribution, i.e. one or more classes are largely under-represented in comparison to the others. Unfortunately, little effort has been directed towards the classification of skewed data streams, although class-imbalance learning has already been studied in the area of pattern recognition on static data. Furthermore, while existing class-imbalance learning methods increase the recognition accuracy on minority class, they often harm the global classification accuracy. Motivated by these observations, we develop an approach suited for classifying skewed data streams, which integrates two ensembles of classifiers, each one suited for non-skewed and skewed data. This approach substantially increases the global accuracy compared to existing classification methods for skewed data. Experimental tests have been carried out on three public datasets showing interesting results. As a further contribution, we will study metrics to evaluate the performance of skewed data streams classification. We will also review the literature on class-imbalance learning, and skewed data streams classification.

1 Introduction

These days many applications deal with large amounts of transaction data, i.e. network traffic data, sensor network data and web usage data [3]. Such data, also referred to as data streams in the rest of the paper, often present skewed distributions, i.e. some classes are not sufficiently represented while instances of other classes are over-represented.

Class imbalance exists in a large number of real-world domains and, hence, learning on the static imbalanced data has received great focus [4,6]. Existing solutions can be divided into the following four categories: (i) under-sampling the majority class, so that its size matches that of the minority class(es); (ii) over-sampling the minority class so as to match the size of the other class(es); (iii) internally biasing the learning process so as to compensate for class imbalance; (iv) multi-experts systems. Despite such efforts, most of these methods, while increase the accuracy on the minority class, decrease the global accuracy in comparison with traditional learning algorithms.

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 254–265, 2012.

Turning our attention to data streams classification, recent research has been directed towards the topic of data streams classification [7,15,8,16]. Few methods, however, have been designed to classify skewed data streams [9].

Therefore, skewed data streams classification deserves more attention. In this respect, we propose here a classification method for skewed data streams, presenting the following contributions: (i) we discuss the pros and cons of metrics for performance evaluation under class skew; (ii) we present a review of the literature concerning classification methods for both static and streaming skewed datasets; (iii) we propose a new approach for skewed data streams classification. Comparing with existing methods, our proposed method improves not only the accuracy on each class but also the global recognition accuracy, as confirmed by experiments carried out on three public datasets.

The rest of the paper is organized as follows: we present background and motivations in section 2, where we also review related work. In section 3, we introduce our approach in detail. In section 4, we report the experimental results. Finally, we conclude the paper in section 5.

2 Background and Motivations

In this paper, we consider two-classes skewed data classification problems, where the minority and majority instances belong to the positive and negative classes respectively, and the positive class is largely under-represented in comparison to the negative one. The skewness of a dataset denotes the degree of data imbalance, and its value is equal to the a priori probability of an instance belonging to the majority class.

2.1 Performance Metrics

For a two-classes classification task, table 1 shows the corresponding confusion matrix which is usually used to assess the performance of a recognition system. We denote $n^- = FP + TN$ and $n^+ = TP + FN$ as the numbers of samples in the negative and positive classes, respectively.

The global recognition accuracy, referred to as *acc*, is a traditional measure for evaluating the performance of a classifier. For a two-classes classification task, $acc = (TP+TN)/(n^-+n^+)$. It is notable that such a measure is sensitive to class skew because it considers values reported in all columns of the confusion matrix. As an example, consider the Credit Card dataset with a skewness of 97.79% (see also subsection 4.1). A classification system would achieve an accuracy as high as $acc = 97.79\%$ if it arbitrarily labels all test samples as negative. However, it would fail to recognize all positive cases, so it cannot meet the need of skewed data classification applications.

As a complementary metric for *acc* on skewed data, we introduce the geometric mean of accuracies (*gacc*) for class-imbalance learning, which is a performance measure used in the literature [12]: $gacc = \sqrt{\prod_{i=1}^{c} \frac{n_{ii}}{n_{+i}}}$, where n_{ii} is the number of elements of class i correctly labeled and n_{+i} is the number of samples

Table 1. Confusion matrix of a two-classes problem

	Actual positive	Actual negative
Hypothesis positive	True Positive (TP)	False Positive (FP)
Hypothesis negative	False Negative (FN)	True Negative (TN)

belonging to class i. Hence, n_{ii}/n_{+i} represents the accuracy for each class. It is clear that *gacc* ranges in $[0, 1]$. For two-classes skewed data classification tasks, we further introduce the following two metrics which specialize in measuring the performance of a classifier on the two different classes:

- *True Positive Rate* or *Recall*, which is defined as $TP_{rate} = acc^+ = \frac{TP}{TP+FN}$;
- *True Negative Rate*, which is defined as $TN_{rate} = acc^- = \frac{TN}{TN+FP}$;

From above definitions, for two classes recognition problem, we obtain $gacc = \sqrt{acc^+ \cdot acc^-}$. On one side, to get a large value of *gacc*, both accuracies should be large. On the other side, *gacc* will be low if either accuracy value is low. Hence, *gacc* is a balance of acc^+ and acc^-. Nevertheless, if we only use the *gacc* value to evaluate a classifier's performance, we can not distinguish its separate performance on the two different classes. As an example, consider the classifier for the Credit Card mentioned above. Its acc^- value is 100% but, since its acc^+ is 0%, the *gacc* value for this classifier is 0%. This example confirms that neither *acc* nor *gacc* on its own is enough to reflect the overall performance of the classifier on skewed data, motivating the use of acc^+ and acc^-.

As a short summary, the metrics of *acc*, *gacc*, acc^+ (or acc^-) should be used together as a joint measure to evaluate classification performance on skewed data streams. Indeed, on the one hand, *acc* measures the global recognition rate and, on the other hand, *gacc* reflects how much classifier performance is balanced. In addition, acc^+ (or acc^-) reports separate classification performance on the two different classes.

2.2 Classification Methods for Skewed Data

Researches for the learning of static imbalanced data can be classified into the following four categories:

1. Under-sampling the majority class by resizing the training sets (**TS**), makes the class distribution more balanced. The main drawback is the removal of the potentially useful samples. One-sided selection is an under-sampling method that tries to overcome tries to overcome this limitation removing borderline and redundant majority class samples, and without touching minority class samples. [2,12].
2. Over-sampling the minority class so as to match the size of the majority one. Synthetic minority over-sampling technique (SMOTE) is an over-sampling approach creating synthetic samples in the feature space along the line segments to join any/all of the k minority class nearest neighbors [5]. Depending

on the amount of needed samples, member samples from the k nearest neighbors are randomly chosen.

3. Internally biasing the discrimination-based process to compensate class imbalance without altering the class distributions. It should assign different weights to prototypes of different classes [13], or use a weighted distance function in the classification phase compensating the TS imbalance without altering the class distribution [1].
4. Multi-experts systems (**MES**). In MES, each composing classifier C_i is trained on a TS composed of a sample subset N_i of the majority class N and all instances from the minority class P. So after sampling a subset N_i from N, C_i is trained on $N_i \cup P$. Later for the test data, the outputs of C_i on the test samples are combined to make the final predication [11]. The main motivation of this approach lies in the observation that a MES generally produces better results than those provided by any of its composing classifiers.

2.3 Classification Methods for Streaming Data

Very fast decision tree learner (VFDT) is an early work for data stream classification [7]. It builds a decision tree incrementally using constant memory. It starts with a single leaf, decides which attribute is the best for splitting the tree, and selects via Hoeffding bound a small subset of examples passing through the nodes. *VFDTc* [8] is an improvement over VFDT that can handle continuous data, incorporate new information online and classify the samples with a single scan of the data.

MES is also applied to data streams building separate classifiers on sequential batches [15]. The performance of existing classifiers are tested using the new batch of data. As a constant number of classifiers is kept, the extra classifiers with worst classification accuracies will be eliminated. The final predication is made by combining the outputs of remaining classifiers through majority voting. In the following, we refer to this method as **SEA**.

Gao et. al. proposed a classification method for skewed data streams [9], which is referred to as **SDM07** in the rest of the paper. To make the class distributions of the TS balanced, they (1) collect minority samples that have appeared over in the new batch and all the past batches, (2) use only the majority instances randomly sampled in the new batch. Samples from steps (1) and (2) are then merged into a new TS used to build a classifier. Moreover, to make more accurate classifications, they generate several such TSs at each new batch by running step (2) several times. The outputs of the set are then combined by majority voting.

2.4 Motivations

The review of the literature reported so far shows that recent research has focused, on the one hand, on class-imbalance learning on static data and, on the other hand, on classifying non-skewed data streams. However, conventional methods for non-skewed data streams usually do not give enough attention to

skewed streams, whereas static class-imbalance learning methods often harm the accuracy on majority class, although they increase the recognition accuracy on the minority class.

Table 2. Experimental results on Credit Card dataset using Naïve Bayes

Batch	SEA				SDM07				Our approach			
	acc	*gacc*	acc^+	acc^-	*acc*	*gacc*	acc^+	acc^-	*acc*	*gacc*	acc^+	acc^-
1	0.9084	0.5233	0.2839	0.9644	0.8422	0.6507	0.4843	0.8743	0.8613	0.6551	0.4793	0.8955
2	0.9019	0.5154	0.2773	0.9578	0.8564	0.6335	0.4495	0.8928	0.8731	0.6062	0.4015	0.9154
3	0.9029	0.5074	0.2682	0.9598	0.8634	0.6215	0.4280	0.9024	0.8704	0.6088	0.4065	0.9120
4	0.8942	0.5357	0.3030	0.9472	0.8676	0.6196	0.4230	0.9075	0.8699	0.6152	0.4156	0.9106
5	0.9142	0.4516	0.2086	0.9774	0.8773	0.6143	0.4106	0.9192	0.8803	0.5892	0.3750	0.9256

To better illustrate such motivations, columns 2-9 of Table 2 compare the classification performance achieved by two methods, namely *SDM07* [9] and *SEA* [15], on the Credit Card dataset with skewness of 97.79%. In Table 2, we observe that data streams classification method designed for training under class skew, i.e. *SDM07*, achieves more balanced performance measured in terms of *gacc* than a conventional classifier adopting learning method tailored for non-skewed data, i.e. *SEA*, due to the fact that the minority class classification accuracy (acc^+) of *SDM07* is larger than *SEA*. However, although acc^+ is improved, we observe that *acc* values returned by *SDM07* is always smaller than those provided by *SEA*, confirming our observation that global accuracy decreases for existing methods handling skewed data streams.

Thus, we are motivated to develop a new skewed data streams classification method that can increase both the global recognition accuracy and the accuracy on minority class.

3 Proposed Method

As reported in section 2.4, we have noticed that balancing the accuracies for each class has the side effect of decreasing the global recognition accuracy (*acc*). Therefore, we present in the following a method aiming at achieving larger *acc* while still improving acc^+ or *gacc*.

3.1 Framework of the Method

Since it is very difficult for a class-imbalance learner to achieve high performance on both *acc* and *gacc*, we decide not to pursue perfect performance on both measures separately, but to develop a classification method that can balance them simultaneously, harming the global accuracy less than previous methods.

So, how can we balance *acc* with *gacc*? We utilize a multi-objective optimization technique selecting the final output of the classification system between the output of a classifier trained according to a learning method addressing the

course of class imbalance, and the output of a classifier adopting a training method for non-skewed data [14]. This choice is driven by a parameter, referred to as threshold t^* in the following, whose value maximizes two objective functions, i.e. the global accuracy (*acc*) and the geometric mean accuracies (*gacc*), on a validation set.

The framework of our method, shown in Fig. 1 (left), is based on two ensembles of classifiers. They are referred to as non-skewed ensemble of classifiers (**NEC**), and skewed ensemble of classifiers (**SEC**). The former is trained on the original skewed distribution, whereas the latter is trained on an artificially balanced training set, applying MES scheme suited for imbalanced data. In our implementation, we use *SEA* [15] for *NEC*, and *SDM07* [9] for *SEC*. This framework is adapted to data streams by means of dividing the training streams into batches, and each batch is further divided into training and validation sets.

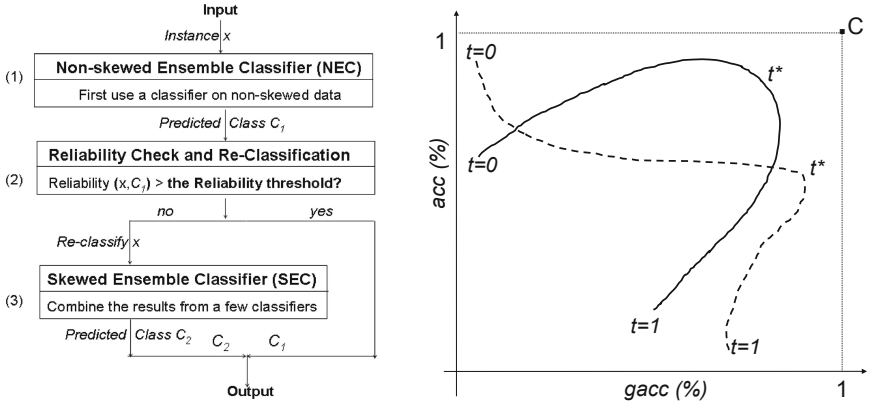

Fig. 1. Framework of Proposed Method (left) and Example of Acc-Gacc curves (right)

Given an instance x belonging to test set, the final label $O(x)$ is determined as follows:

$$O(x) = \begin{cases} O_{NEC}(x) & \text{if } \phi(x) \geq t^* \\ O_{SEC}(x) & \text{otherwise} \end{cases} \quad (1)$$

When the reliability $\phi(x)$ provided by *NEC* is larger than the threshold t^*, the final label corresponds to the label returned by *NEC* because it is reasonable to assume that *NEC* is likely to provide a correct classification. But, when $\phi(x)$ is below t^*, $O(x)$ is equal to the label assigned by *SEC*, i.e. a classification method tailored specially for skewed data. Indeed, in this case, the value of the reliability suggests that the decision returned by NEC may not be safe. We will explain the rationale of reliability estimation in subsection 3.3.

3.2 Multi-objective Optimization

According to our proposal, we will first train NEC and SEC on the training set of a given batch. Next, both of them are used to classify instances belonging to the validation set of the batch to determine the best value of t^* to be used with the test data. Finally for test data, we apply equation 1 to set the final classification. As reported above, the choice between the outputs of NEC and SEC is driven by t^*. Since t^* is an important threshold parameter, how do we set this parameter? In order to answer this question, recall that *gacc* measures how much the accuracies on two classes are balanced, whereas *acc* estimates the global performance of the classification system. Let us represent *gacc* and *acc* on the X and Y axes respectively, and vary a threshold t to generate a set of points that can be used to plot a curve using samples belonging to validation set. The curve extrema at $t = 0$ and $t = 1$ correspond to *NEC* and *SEC* performance, respectively. In this plot, the ideal point is $\mathbf{C} = (1, 1)$; hence, the nearer the curve to this point, the better the performance obtained. Therefore, the value t^* is given by $\arg\min_t(||\mathbf{p}(t) - \mathbf{C}||)$, where $\mathbf{p}(t)$ is the pair of $gacc(t)$ and $acc(t)$ values measured on the the validation set when the threshold t is used.

Fig. 1 (right) shows two examples of this curve, corresponding to two different situations that may occur. The first situation is represented by the continuous line in the figure. In this case, the proposed method selects a value of t^* that permits to improve both *gacc* and *acc* in comparison to individual performance of *NEC* and *SEC*, i.e. points marked with $t = 0$ and $t = 1$. The second situation is represented by the dashed curve. In this case, the proposed method selects a value of t^* that improves *gacc* with respect to both *NEC* and *SEC*, while it reduces slightly the value of *acc* in comparison to *NEC*. We deem that such a reduction can be accepted since final performance are more balanced than individual ones returned by *NEC* and *SEC*.

Algorithm 1 shows the algorithm implementing our proposal presented so far. The training stream is divided into sequential batches (line 1), and each batch is further divided into training and validation sets (line 3-(a)). Using the training set, we train *NEC* and *SEC* (line 3-(b)). Next we compute t^* using a validation set and applying the method given in subsection 3.2, (line3-(c)). As *NEC* and *SEC* are both ensemble of classifiers, we collect the member classifiers in line 3-(d). To classify test instances, we apply step 3-(e) according to equation 1.

3.3 Reliability Estimation

This subsection answers to the following question: what is the reliability and why is it useful in the proposed method?

Utilizing information derived from classifier outputs allows for estimating the reliability of each classification act. Reliability takes into account many issues that influence the achievement of a correct classification, such as the noise affecting the samples domain, and the differences between the objects to be recognized and those used for training the classifiers.

Let $\phi(x)$ denote the reliability of a classification act on any instance x, and the value range within $[0, 1]$. For two-class classifiers, $\phi(x)$ is computed using

Algorithm 1. Algorithm of the proposed method

1. Divide the labeled dataset $\mathbf{Z}$ into n batches $D_1, D_2, \ldots, D_n$.
2. Let $\mathbf{Z}_{Te}$ be the test set.
3. For each batch:
   (a) Divide the samples into training and validation sets, denoted by $\mathbf{Z}_{Tr}$, $\mathbf{Z}_{Va}$.
   (b) Train the non-skewed ensemble classifier (NEC) and a skewed ensemble classifier (*SEC*) on $\mathbf{Z}_{Tr}$.
   (c) Find the best threshold t^* s.t. the system achieve the largest values of both *acc* and *gacc*.
   (d) Collect trained NEC_i and SEC_i, with $i = 1, 2, \ldots, n$.
   (e) Apply NEC_i and SEC_i to $\mathbf{Z}_{Te}$, using the following classification rule

$$O(x) = \begin{cases} O_{NEC}(x) & \text{if } \phi(x) \geq t^* \\ O_{SEC}(x) & \text{otherwise} \end{cases}$$

   where x is a sample, $O_{NEC}(x)$ and $O_{SEC}(x)$ are the outputs provided by *NEC* and *SEC*, $\phi(x)$ is the reliability of *NEC*, and $O(x)$ is the final label.

the difference of predictions on the two different classes. A low value of $\phi(x)$ will suggest that the classification decision made on instance x is not safe since, for instance, it may be a borderline instance or it can be affected by noise in the feature space; while a large value of $\phi(x)$ would suggest that the classifier is more likely to have provided a correct classification [10].

In order to explain the rationale of using the reliability for skewed data classification, let us consider Fig. 2, where we report the experimentally measured distributions of reliability values for test samples labeled by a classifier (i.e. *NEC*) trained on a skewed distribution. On the one hand, when we apply *NEC*, the minority class samples are more likely to receive low reliability values (see the left part of Fig. 2). On the other hand, although low reliability values can also be found for true negative instances, instances with high reliability values are more likely to belong to the majority (negative) class (see the right part of Fig. 2).

In short, there are two main reasons for using reliability estimations on skewed data stream classifiers: (1) applying *NEC*, samples with high reliability values are more likely to belong to negative (majority) class. Hence, we can use reliability values to distinguish between positive and negative instances; (2) *SEC* is trained on artificially balanced training sets, so it should recognize not only positive instances, but also negative ones. Therefore, although instances with low reliability values can contain negative (majority) instances, *SEC* should be able to correctly classify most of them.

4 Experimental Evaluation

In this section, we first describe the datasets used for the experiments. Second, we introduce the experimental protocol and, third, we report the experimental results.

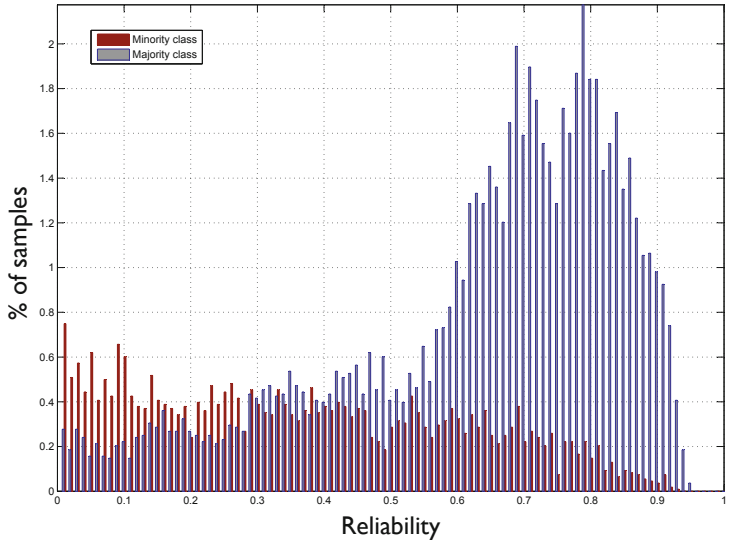

Fig. 2. Examples of reliability distributions for majority and minority class samples

4.1 Datasets

We use the three datasets shown in table 3. These datasets vary in both number of features and skewness. The prediction task for the Adult dataset is to determine whether a person makes over 50K income a year. We only use two classes of the Forest Cover dataset: Ponderosa and Lodgepole Pine. The task of the Forest Cover dataset is to predict the forest cover type. The Credit Card dataset was provided by the 2009 UCSD/FICO data mining contest[1] and used for predicting whether a transaction is an anomaly or not.

Table 3. Datasets description

Datasets	Number of instances	Skewness	Number of features	Source
Adult	44848	70.70%	14	UCI
Forest Cover	319055	88.79%	54	UCI
Credit Card	94682	97.79%	19	UCSD

4.2 Experimental Protocol

We test our approach on the above mentioned datasets. For each dataset, we divide the data into batches, and the last two batches are left for testing only. We vary the size of the batches in our tests: each batch in the Credit Card data contains 10,000 transactions, the size of a batch in the Forest Cover dataset is 20,000, and it is 5,000 for the Adult data.

[1] *http://mill.ucsd.edu*

Table 4. Experimental results on Credit Card dataset using Logistic Regression

Batch	SEA				SDM07				Our approach			
	acc	*gacc*	acc^+	acc^-	*acc*	*gacc*	acc^+	acc^-	*acc*	*gacc*	acc^+	acc^-
1	0.9095	0.5161	0.2757	0.9663	0.7886	0.6490	0.5182	0.8128	0.8692	0.5896	0.3808	0.9130
2	0.9029	0.5469	0.3129	0.9558	0.8027	0.6721	0.5472	0.8256	0.7989	0.6652	0.5381	0.8223
3	0.9082	0.4861	0.2442	0.9677	0.8084	0.6804	0.5571	0.8309	0.8113	0.6725	0.5414	0.8355
4	0.9151	0.5253	0.2839	0.9717	0.8200	0.6729	0.5356	0.8455	0.8740	0.5819	0.3684	0.9193
5	0.9139	0.4941	0.2508	0.9734	0.8084	0.6804	0.5571	0.8309	0.8158	0.6735	0.5397	0.8405
6	0.9084	0.4764	0.2343	0.9688	0.8109	0.6777	0.5505	0.8343	0.8138	0.6658	0.5281	0.8394
7	0.9034	0.4767	0.2359	0.9633	0.8207	0.6844	0.5546	0.8445	0.8664	0.6097	0.4098	0.9073
8	0.9130	0.4625	0.2194	0.9752	0.8147	0.6860	0.5621	0.8373	0.8792	0.5729	0.3543	0.9263

As there are few methods for skewed data stream classification, we implement *SDM07* [9] for *SEC* and we apply *SEA* [15] for *NEC*. These two methods were chosen because both of them are well recognized methods for classifying skewed or non-skewed data streams. Since both *NEC* and *SEC* are classifier ensembles, we use C4.5, Naïve Bayes and Logistic Regression as the base learners in our experiments. Performance are estimated measuring *acc*, *gacc*, acc^+, and acc^-.

4.3 Results

Tables 2, 4 and 5 report the results of the tests we performed on the Credit Card dataset. Tables 6 and 7 show a portion of the test results on both Adult and Forest Cover datasets. It is worth noting that *SEA* usually achieves the largest *acc* value but has the smallest *gacc* value. The case is reversed for *SDM07*, with the largest value for *gacc* but the smallest value for *acc*. Our proposed method, however, achieves a balanced performance between the two above methods. As discussed in section 2, this occurs because *SEA* is a learning method that usually ignores the minority class in skewed data. *SDM07*, on the other hand, is biased toward the minority class but harms the recognition accuracy on majority class. Unlike the other two methods, our proposed approach balances *acc* and *gacc* simultaneously.

We now provide a deeper analysis of the results achieved on the Credit Card dataset (Tables 2, 4 and 5). We notice that: (i) *SEA* usually achieves the best values of *acc*, while *SDM07* often has the best values of both acc^+ and *gacc*; (ii) Sometimes the *gacc* values of our method are as large as or even larger than *SDM07*; (iii) Our method increases the values of the acc^+ of *SEA* by up to 70%. Our method also outperforms *SEA* in terms of *gacc* by 25%; (iv) Our method does not outperform *SEA* in terms of *acc*. With respect to *SEA*, our method decreases *acc* by approximately 4%, but the decrease in *acc* is usually 13% in the case of *SDM07*.

In summary, the above observations show that the proposed method takes into account minority class instances without harming the global accuracy as much as existing methods. We owe this fact to both the double-ensemble framework and the multi-objective optimization technique embedded in the learning algorithm, which dynamically adapts its threshold to variation in data distribution.

Table 5. Experimental results on Credit Card dataset using C4.5

Batch	SDM07				Our approach			
	acc	$gacc$	acc^+	acc^-	acc	$gacc$	acc^+	acc^-
1	0.6908	0.7315	0.7839	0.6825	0.7143	0.7251	0.7384	0.7121
2	0.7261	0.7460	0.7707	0.7221	0.7531	0.7383	0.7210	0.7560
3	0.6952	0.7284	0.7707	0.6884	0.8000	0.6974	0.5944	0.8144
4	0.7153	0.7330	0.7641	0.7109	0.7326	0.7269	0.7202	0.7337
5	0.7101	0.7415	0.7815	0.7307	0.8090	0.6970	0.5681	0.8290
6	0.7163	0.7469	0.7856	0.7100	0.7086	0.7357	0.7699	0.7031
7	0.7124	0.7468	0.7906	0.7054	0.7880	0.7271	0.6614	0.7993
8	0.7100	0.7387	0.7748	0.7042	0.7410	0.7387	0.7359	0.7415

Table 6. Experimental results on Adult dataset using C4.5

Batch	SDM07				Our approach			
	acc	$gacc$	acc^+	acc^-	acc	$gacc$	acc^+	acc^-
1	0.7867	0.8051	0.8438	0.7681	0.7941	0.8092	0.8406	0.7790
2	0.8012	0.8181	0.8535	0.7842	0.8026	0.8136	0.8363	0.7916
3	0.8158	0.8242	0.8411	0.8075	0.8339	0.8021	0.7458	0.8606
4	0.7987	0.8181	0.8589	0.7791	0.8120	0.8087	0.8024	0.8151
5	0.8105	0.8137	0.8201	0.8074	0.8334	0.7841	0.7017	0.8762

Similar results were also found in the experiments with the other two datasets. The results are shown in Table 6 and Table 7.

Table 7. Experimental results on Forest Cover dataset using Naïve Bayes

Batch	SEA				SDM07				Our approach			
	acc	$gacc$	acc^+	acc^-	acc	$gacc$	acc^+	acc^-	acc	$gacc$	acc^+	acc^-
1	0.9430	0.9274	0.9077	0.9475	0.9272	0.9332	0.9411	0.9254	0.9440	0.9262	0.9038	0.9491
2	0.9403	0.9326	0.9228	0.9425	0.9244	0.9360	0.9511	0.9211	0.9393	0.9330	0.9262	0.9398
3	0.9385	0.9308	0.9209	0.9408	0.9249	0.9348	0.9477	0.9220	0.9363	0.9313	0.9250	0.9377
4	0.9403	0.9319	0.9211	0.9428	0.9253	0.9345	0.9465	0.9226	0.9371	0.9289	0.9185	0.9395
5	0.9413	0.9333	0.9232	0.9436	0.9264	0.9354	0.9472	0.9237	0.9403	0.9336	0.9252	0.9422
6	0.9421	0.9313	0.9176	0.9452	0.9262	0.9358	0.9485	0.9233	0.9390	0.9326	0.9244	0.9408
7	0.9379	0.9328	0.9264	0.9393	0.9246	0.9366	0.9525	0.9210	0.9396	0.9318	0.9218	0.9418
8	0.9412	0.9334	0.9235	0.9434	0.9253	0.9363	0.9508	0.9221	0.9369	0.9353	0.9331	0.9374
9	0.9390	0.9319	0.9229	0.9411	0.9239	0.9367	0.9534	0.9202	0.9378	0.9316	0.9237	0.9395
10	0.9396	0.9313	0.9206	0.9420	0.9254	0.9363	0.9506	0.9223	0.9406	0.9303	0.9172	0.9436

Finally, we report the elapsed time during training and test phases of each method. The running time increases with the batch size. Using C4.5 as the base learner on Credit Card data, the proposed method takes 353 seconds, whereas *SDM07* spends 280 seconds. In the case of the Adult data, the proposed method and *SDM07* use 274 and 234 seconds, respectively. These results are reasonable, because the proposed method trains two ensembles of classifiers. Hence, the training time is slight longer than that of *SDM07*.

5 Conclusions

In this paper, we have presented a classification method for skewed data streams. This method is based on two classifier ensembles suited for learning with and without class skew. While still improving the accuracy on each class, the proposed method does not decrease the global recognition accuracy as much as existing methods. Future work will be directed towards extending our study to multi-class data streams.

References

1. Barandela, R., Valdovinos, R.M., Sánchez, J.S.: New applications of ensembles of classifiers. Pattern Analysis & Applications 6(3), 245–256 (2003)
2. Batista, G.E., Carvalho, A.C., Monard, M.C.: Applying One-sided Selection to Unbalanced Datasets. In: Cairó, O., Cantú, F.J. (eds.) MICAI 2000. LNCS, vol. 1793, pp. 315–325. Springer, Heidelberg (2000)
3. Bay, S.D., Kibler, D., Pazzani, M.J., Smyth, P.: The uci kdd archive of large data sets for data mining research and experimentation. SIGKDD Explorations, 81–85 (2000)
4. Chan, P.K., Fan, W., Prodromidis, A.: Distributed data mining in credit card fraud detection. IEEE Intelligent Systems 14, 67–74 (1999)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
6. Chawla, N.V., Japkowicz, N.: Editorial: Special issue on learning from imbalanced data sets. SIGKDD Explorations 6 (2004)
7. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proc. SIGKDD, pp. 71–80 (2000)
8. Gama, J., Rocha, R., Medas, P.: Accurate decision trees for mining high-speed data streams. In: Proc. SIGKDD, pp. 523–528 (2003)
9. Gao, J., Fan, W., Han, J., Yu, P.S.: A general framework for mining concept-drifting data streams with skewed distributions. In: Proc. SIAM SDM 2007, pp. 3–14 (2007)
10. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions On Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
11. Kotsiantis, S., Pintelas, P.: Mixture of expert agents for handling imbalanced data sets. Ann. of Mathematics, Computing and Teleinformatics 1(1), 46–55 (2003)
12. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: Proc. ICML 1997, pp. 179–186 (1997)
13. Pazzani, M., Merz, C., Murphy, P., Ali, K.: Reducing misclassification costs. In: Proc. ICML 1994, pp. 217–225 (1994)
14. Soda, P.: A multi-objective optimisation approach for class imbalance learning. Pattern Recognition 44(8), 1801–1810 (2011)
15. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proc. SIGKDD, pp. 377–382 (2001)
16. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: SIGKDD, pp. 226–235 (2003)