Clustering and Weighted Scoring in Geometric Space Support Vector Machine Ensemble for Highly Imbalanced Data Classification

Paweł Ksieniewicz[0000−0001−9578−8395] and Robert Burduk[0000−0002−3506−6611]

Department of Systems and Computer Networks
Wroclaw University of Science and Technology
Wroclaw, Poland
{pawel.ksieniewicz, robert.burduk}@pwr.edu.pl

Abstract. Learning from imbalanced datasets is a challenging task for standard classification algorithms. In general, there are two main approaches to solve the problem of imbalanced data: algorithm-level and data-level solutions. This paper deals with the second approach. In particular, this paper shows a new proposition for calculating the weighted score function to use in the integration phase of the multiple classification system. The presented research includes experimental evaluation over multiple, open-source, highly imbalanced datasets, presenting the results of comparing the proposed algorithm with three other approaches in the context of six performance measures. Comprehensive experimental results show that the proposed algorithm has better performance measures than the other ensemble methods for highly imbalanced datasets.

Keywords: Imbalanced Data · Ensemble of classifiers · Class imbalance · Decision boundary · Scoring function.

1 Introduction

The goal of the supervised classification is to build a mathematical model of a real-life problem using a labeled dataset. This mathematical model is used to assign the class label to each new recognized object, which, in general, does not belong to the training set. The individual classification model is called a base classifier. Ensemble methods are a vastly used approach to improve the possibilities of base classifiers by building a more stable and accurate *ensemble of classifiers* (*EoC*) [23, 28]. In general, the procedure for building *EoC* consists of three steps: generation, selection, and integration [18]. An imbalanced data problem occurs when the *prior* probability of classes in a given dataset is very diverse. There are many real-life problems in which we deal with imbalanced data [11, 25], e.g., network intrusion detection [2, 14], source code fault detection [8], or in general fraud detection [1].

There exist two main approaches to solve the problem of imbalanced data: a data-level [9, 26] and an algorithm-level solution [29]. EoC is one of the ap-

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

proaches to solve the imbalanced data classification problem which improve classification measure compared to single models and is highly competitive and robust to imbalanced data [10, 13, 16]. The use of not only voting in the *EoC* integration phase is one of the directions to solve a problem with imbalanced data [15]. Therefore, this article concerns about calculating the weighted scoring function to be applied in the weighted voting process.

In the process of *EoC* generation, we use the *K-Means* clustering algorithm[5] for each class label separately. The base linear classifier – *Support Vector Machine* [7] – is trained on cluster combination. The weighted scoring function takes into account the distance of a classified object from the decision boundary and cluster centroids used to learn the proper base classifier. Regardless of the number of learning objects in a given cluster, the cluster centroid is always determined. The proposed method for determining the scoring function is, therefore, insensitive to the number of objects defining the cluster. As shown in the article, the proposed approach is useful for imbalanced data.

The main objectives of this work are summarized as follows:

- A proposal of a new weighted scoring function that uses the location of the cluster centroids and distance to the decision boundary.
- The proposition of an algorithm that uses clustering and the proposed function.
- A new experimental setup to compare the proposed method with other algorithms on highly imbalanced datasets.

The paper is structured as follows: Section 2 introduces the base concept of *EoC* and presents the proposed algorithm. In Section 3, the experiments that were carried out are presented, while results and the discussion appear in Section 4. Finally, we conclude the paper in Section 5.

2 Clustering and Weighted Scoring

2.1 Basic notation

From a probabilistic point of view the recognition algorithm Ψ maps the feature space $\mathcal{X} \subseteq \Re^d$ to the set of class labels $\Omega = \{\omega_1, \omega_2, \ldots, \omega_C\}$ according to the general formula:

$$\Psi : \mathcal{X} \to \Omega. \tag{1}$$

For the feature vector $x \in \mathcal{X}$, that represents the recognized object the Equation (1) can be expressed as:

$$\Psi(x) = \omega_c. \tag{2}$$

Let us assume that L different base classifiers $\Psi_1, \Psi_2, \ldots, \Psi_L$, are employed to solve the classification task. This set of classifiers defines *EoC*. One of the most popular methods to integrate outputs of the base classifiers set L is the majority vote rule. In this method, each base model has the same impact on the final decision of *EoC*. This method allows counting base classifiers outputs as

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

a vote for a class and assigns the input pattern to the class with the greatest count of votes. It is defined as follows:

$$\Psi_{MV}(x) = \arg\max_{\omega_c} \sum_{k=1}^{L} I(\Psi_k(x) = \omega_c), \tag{3}$$

where $I(\cdot)$ is the indicator function.

In the weighted majority voting rule, the integration phase includes probability estimators or other factors of base models to the final decision of EoC [19], like in Equation 4:

$$\Psi_{MV}(x) = \arg\max_{\omega_c} \sum_{k=1}^{L} w_k I(\Psi_k(x) = \omega_c), \tag{4}$$

where w_k is the weight assigned to the classifier Ψ_k.

Over the last years, the issue of calculating the weights in the voting rule has been considered many times. The article [30] presents an approach in which the weights are combining with local confidence. The classifier trained on a subset of training data should be limited to the area it spans in an impact on the resulting classifier. The problem of generalization of majority voting was studied in [12]. The authors are using a probability estimate calculated as percentage of properly classified validation objects over geometric constraints. Separately, regions that are functionally independent are considered. A significant improvement in the classification quality was observed when using the proposed algorithm, although knowledge of the domain is needed to provide a proper division. The authors are using a retinal image and classify over anatomic regions.

The weights of the base classifier are also considered in the context of the interval-valued fuzzy sets [6]. The upper weight of base the classifier refers to the situation in which the definite base classifier was correct, while the other classifiers proved the correct prediction. The lower weight describes the situation in which the definite base classifier made errors, while the other classifiers didn't make any errors.

In the paper [22] weights are determined for each label separately over the entire validation dataset. This can lead to the improvement of the performance of the resulting integrated classifier.

The following article is a proposition of an algorithm assigning weights not to base classifiers, but recognized objects. The weight for each object depends on its location in the feature space. Therefore, the weight of an object is determined by the score function calculated in the geometric space.

2.2 Proposed Method

We propose that the score function of the object x depends on its position in the geometric space. In particular, the distance from the decision boundary of the

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

base classifier Ψ_l from EoC and the cluster centroids used to learn this classifier are used to calculated the unweighted score function :

$$sf_l(x) = \|\psi_l(x)\| + \sum_{c=1}^{C} d_c, \tag{5}$$

where d_c is the distance from x to cluster centroid in *Manhattan* metric. This metric was chosen because of the lowest calculation cost among all the considered alternatives. The calculation of the distance occurs between the centroids of all clusters and all tested patterns, so the computational complexity of the prediction procedure is very much dependent on the chosen metric, which is the reason for minimizing its impact.

We propose the following scoring weighting method:

$$wsf_l(x) = 1 - \frac{sf_l(x)}{\sum_{l=1}^{L} sf_l(x)}, \tag{6}$$

which includes all scoring functions obtained for each classifier from EoC.

Figure 1 shows how to calculate the object's score function for a linear dichotomic classifier and two cluster centroids. A solid red line marks the decision boundary of the base linear classifier Ψ_l constructed for the selected class cluster combination – C^{ω_1} and C^{ω_2}. Blue points are cluster centroids. The sum of the dashed sections indicates the value of the score function for the tested object x. The red dashed line – the distance to decision boundary $\|\psi_l(x)\|$, the blue dashed line – the distance to cluster centroids determined by the *Manhattan* metric d_1 and d_2.

Algorithm (1) presents the pseudocode of the proposed approach to EoC with clustering and weighted scoring in the geometric space. In addition, Algorithm (1) concerns the dichotomous division of the learning set into class labels. These types of highly imbalanced datasets were used in the experimental research.

3 Experiments set-up

The experimental evaluation conducted for the needs of verification of the method proposed in the following work was based of 30 highly imbalanced datasets contained in the KEEL repository [3]. Datasets selected for the study are characterized by an *imbalance ratio* (the proportion between minority and majority class) ranging from 1:9 up to 1:40. Besides, due to the preliminary nature of the conducted research, the pool of datasets includes only binary classification problems.

The basis of the used division methodology was *Stratified K-Fold Crossvalidation* with $k = 5$, necessary to ensure the presence of minority class patterns in each of the analyzed training subsets. Statistical tests, for both pair and rank tests, were carried out using the *Wilcoxon test* with the significance level $\alpha = 0.05$ [4].

The analysis was conducted following the four classification approaches:

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

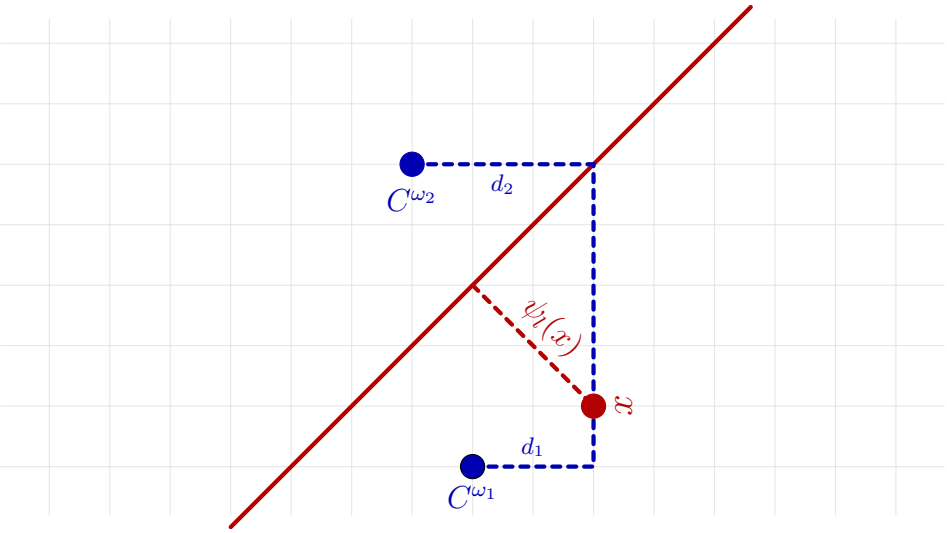

Fig. 1. Schema for calculating the of score function for the object x.

- (SVC) *Support Vector Machine* — the base experimental model with the scaled gamma and linear kernel [21].
- (CWS) *Clustering and Weighted Scoring* — *EoC* with the pool diversified by pairs of clusters and integrated geometrically by the rules introduced in Section 2.
- (CMV) *Clustering and Majority Vote* — *EoC* identical with CWS but integrated using the majority vote [24].
- (CSA) *Clustering and Support Accumulation* — *EoC* identical with CWS and CMV but integrated using the support accumulation rule [27].

In the construction of each *EoC*, in order to limit the number of the presented tables and readability of the analysis, each time we construct the ensemble by dividing classes into two clusters, thus building a pool of four members. In the case of data as strongly imbalanced as those from the selected databases, often only a few (literally four or five) minority class objects remain in a single fold so that a more substantial number would treat almost every minority object as a separate cluster.

The whole experimental evaluation was performed in Python, using the *scikit-learn* API [20] to implement the CWS method and is publicly available on the GIT repository[1]. As metrics for the conducted analysis, due to the imbalanced nature of the classification problem, three aggregate measures (*balanced accuracy score*, *F1-score*, and *G-mean*) and three base measures constituting their calculation (*precision*, *recall*, and *specificity*) were applied, using their implementation included in the *stream-learn* package [17].

[1] https://github.com/w4k2/geometric-integration

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

Algorithm 1: Clustering and Weighted Scoring in Geometric Space algorithm – for binary problem

Input: Learning set D, Number of clusters K – equal in each class label, object x

Output: The ensemble classifier decision

1 Cluster D into K clusters label using the K-means clustering procedure separately for each class. The final cluster combination equals $L = 2 * K$ for the binary problem.

2 Find the cluster centroids $C_1^{\omega_1}, \ldots, C_K^{\omega_1}, C_1^{\omega_2}, \ldots, C_K^{\omega_2}$ as the means of the points in the respective clusters.

3 Train base classifier $\Psi_1, \ldots, \Psi_L$ using each combination of clusters from different class labels.

4 Calculate weighted scoring functions for the object x:

$$wsf_l(x) = 1 - \frac{sf_l(x)}{\sum_{l=1}^{L} sf_l(x)},$$

where

$$sf_l(x) = \|\psi_l(x)\| + \sum_{c=1}^{2} d_c.$$

5 The ensemble classifier decision:

$$\Psi_{CWS}(x) = sign\left(\sum_{l=1}^{L} wsf_l(x)\Psi_l(x)\right),$$

where $\Psi(x)$ is the prediction returned by base classifier $\Psi(x) \in \{-1, 1\}$.

4 Experimental Evaluation

For the readability of the analysis, the full results of the experiment, along with the presentation of the relation between the algorithms resulting from the conducted paired tests, are presented only for the *balanced accuracy score* (Table 1) and *recall* (Table 2) metrics.

As may be observed, for aggregate metrics (results are consistent for both *balanced accuracy* and *G-mean*, only in *F1-score* presenting a slightly smaller scale of differences) the use of majority voting (CMV) for *EoC* diversified with clustering, often leads to deterioration of the classification quality even concerning a single base classifier. Integration by support accumulation (CSA) performs slightly better, due to taking into consideration the certainty (*support*) of the decisions of each classifier but ignoring their areas of competence. The use of areas of competence present in the CWS method allows for substantial improvement in classification results, often leading to a statistically significant advantage. The primary source of advantage in results is a significant improvement in the *recall* metric in this approach.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

Table 1. Results achieved by the analyzed method for the *balanced accuracy score* metric.

Dataset	IR	1 SVC	2 CWS	3 CMV	4 CSA
glass-0-4-vs-5	1:9	0.738 ± 0.160	0.938 ± 0.125	0.696 ± 0.247	0.856 ± 0.174
ecoli-0-1-4-7-vs-5-6	1:12	0.867 ± 0.076	0.839 ± 0.060	0.713 ± 0.063	0.856 ± 0.025
ecoli-0-6-7-vs-5	1:10	0.890 ± 0.103	0.915 ± 0.061	0.710 ± 0.056	0.882 ± 0.108
ecoli-0-1-vs-2-3-5	1:9	0.880 ± 0.083	0.871 ± 0.115	0.692 ± 0.086	0.780 ± 0.142
ecoli-0-3-4-6-vs-5	1:9	0.845 ± 0.118	0.890 ± 0.085	0.720 ± 0.081	0.879 ± 0.092
yeast-0-3-5-9-vs-7-8	1:9	0.537 ± 0.081	0.597 ± 0.129	0.549 ± 0.078	0.539 ± 0.080
ecoli4	1:16	0.570 ± 0.140	0.753 ± 0.088	0.619 ± 0.149	0.500 ± 0.000
ecoli-0-1-4-7-vs-2-3-5-6	1:11	0.796 ± 0.120	0.790 ± 0.098	0.666 ± 0.130	0.756 ± 0.085
ecoli-0-3-4-7-vs-5-6	1:9	0.766 ± 0.097	0.744 ± 0.114	0.829 ± 0.057	0.779 ± 0.139
shuttle-c2-vs-c4	1:20	1.000 ± 0.000	1.000 ± 0.000	0.756 ± 0.026	1.000 ± 0.000
yeast-2-vs-8	1:23	0.774 ± 0.120	0.774 ± 0.120	0.600 ± 0.093	0.650 ± 0.093
ecoli-0-4-6-vs-5	1:9	0.839 ± 0.118	0.867 ± 0.070	0.611 ± 0.103	0.875 ± 0.093
yeast-2-vs-4	1:9	0.667 ± 0.093	0.693 ± 0.070	0.634 ± 0.068	0.627 ± 0.039
ecoli-0-6-7-vs-3-5	1:9	0.853 ± 0.077	0.835 ± 0.101	0.728 ± 0.185	0.807 ± 0.102
ecoli-0-1-4-6-vs-5	1:13	0.829 ± 0.125	0.863 ± 0.077	0.631 ± 0.143	0.829 ± 0.063
ecoli-0-2-3-4-vs-5	1:9	0.875 ± 0.075	0.859 ± 0.119	0.745 ± 0.100	0.804 ± 0.129
glass-0-6-vs-5	1:11	0.639 ± 0.195	0.924 ± 0.103	0.744 ± 0.186	0.766 ± 0.113
ecoli-0-2-6-7-vs-3-5	1:9	0.860 ± 0.115	0.833 ± 0.092	0.628 ± 0.079	0.785 ± 0.123
ecoli-0-3-4-vs-5	1:9	0.822 ± 0.123	0.886 ± 0.090	0.761 ± 0.143	0.911 ± 0.055
glass4	1:15	0.554 ± 0.093	0.914 ± 0.071	0.640 ± 0.124	0.568 ± 0.139
glass5	1:23	0.544 ± 0.087	0.882 ± 0.121	0.704 ± 0.188	0.737 ± 0.162
glass-0-1-5-vs-2	1:9	0.500 ± 0.000	0.500 ± 0.000	0.601 ± 0.159	0.507 ± 0.086
yeast-0-2-5-6-vs-3-7-8-9	1:9	0.509 ± 0.021	0.581 ± 0.101	0.532 ± 0.066	0.504 ± 0.010
yeast3	1:8	0.630 ± 0.035	0.500 ± 0.042	0.632 ± 0.050	0.701 ± 0.045
ecoli-0-1-vs-5	1:11	0.880 ± 0.093	0.932 ± 0.061	0.895 ± 0.123	0.864 ± 0.090
shuttle-c0-vs-c4	1:14	1.000 ± 0.000	1.000 ± 0.000	0.785 ± 0.030	0.992 ± 0.009
yeast6	1:41	0.500 ± 0.000	0.528 ± 0.055	0.500 ± 0.000	0.500 ± 0.000
yeast4	1:28	0.500 ± 0.000	0.510 ± 0.020	0.500 ± 0.000	0.500 ± 0.000
yeast-0-2-5-7-9-vs-3-6-8	1:9	0.704 ± 0.099	0.840 ± 0.158	0.669 ± 0.072	0.578 ± 0.067
vowel0	1:10	0.767 ± 0.119	0.719 ± 0.129	0.786 ± 0.079	0.787 ± 0.079

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

Table 2. Results achieved by the analyzed method for the *recall* metric.

Dataset	IR	1 SVC	2 CWS	3 CMV	4 CSA
glass-0-4-vs-5	1:9	0.600 ± 0.374	1.000 ± 0.000	0.500 ± 0.447	0.800 ± 0.244
ecoli-0-1-4-7-vs-5-6	1:12	0.800 ± 0.178	0.760 ± 0.080	0.680 ± 0.097	0.800 ± 0.000
ecoli-0-6-7-vs-5	1:10	0.800 ± 0.187	0.900 ± 0.122	0.450 ± 0.100	0.800 ± 0.187
ecoli-0-1-vs-2-3-5	1:9	0.800 ± 0.178	0.760 ± 0.233 [3]	0.430 ± 0.218	0.620 ± 0.271
ecoli-0-3-4-6-vs-5	1:9	0.750 ± 0.273	0.850 ± 0.200	0.500 ± 0.158	0.850 ± 0.200
yeast-0-3-5-9-vs-7-8	1:9	0.080 ± 0.160	0.200 ± 0.252	0.100 ± 0.154	0.080 ± 0.160
ecoli4	1:16	0.150 ± 0.300	0.550 ± 0.244	0.250 ± 0.316	0.000 ± 0.000
ecoli-0-1-4-7-vs-2-3-5-6	1:11	0.687 ± 0.289	0.720 ± 0.231 [1,4]	0.540 ± 0.257	0.680 ± 0.263
ecoli-0-3-4-7-vs-5-6	1:9	0.640 ± 0.265	0.600 ± 0.219	0.840 ± 0.079	0.680 ± 0.271
shuttle-c2-vs-c4	1:20	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
yeast-2-vs-8	1:23	0.550 ± 0.244	0.550 ± 0.244	0.200 ± 0.187	0.300 ± 0.187
ecoli-0-4-6-vs-5	1:9	0.750 ± 0.273	0.800 ± 0.187	0.300 ± 0.187	0.850 ± 0.200
yeast-2-vs-4	1:9	0.335 ± 0.186	0.396 ± 0.146 [3]	0.276 ± 0.149	0.256 ± 0.083 [3]
ecoli-0-6-7-vs-3-5	1:9	0.740 ± 0.146	0.780 ± 0.203	0.490 ± 0.361	0.640 ± 0.185
ecoli-0-1-4-6-vs-5	1:13	0.700 ± 0.244	0.850 ± 0.200	0.350 ± 0.339	0.850 ± 0.200
ecoli-0-2-3-4-vs-5	1:9	0.800 ± 0.187	0.900 ± 0.122	0.600 ± 0.200	0.850 ± 0.122
glass-0-6-vs-5	1:11	0.300 ± 0.399 [3]	0.900 ± 0.200	0.600 ± 0.374	0.800 ± 0.244
ecoli-0-2-6-7-vs-3-5	1:9	0.760 ± 0.224	0.770 ± 0.203	0.310 ± 0.215	0.610 ± 0.261
ecoli-0-3-4-vs-5	1:9	0.800 ± 0.187 [3]	0.900 ± 0.122	0.650 ± 0.254	0.900 ± 0.122
glass4	1:15	0.233 ± 0.200	0.933 ± 0.133	0.300 ± 0.266	0.200 ± 0.266
glass5	1:23	0.200 ± 0.400	0.900 ± 0.200 [all]	0.500 ± 0.316	0.600 ± 0.374
glass-0-1-5-vs-2	1:9	0.000 ± 0.000	0.000 ± 0.000	0.633 ± 0.335	0.633 ± 0.335
yeast-0-2-5-6-vs-3-7-8-9	1:9	0.031 ± 0.040	0.197 ± 0.252	0.074 ± 0.147	0.011 ± 0.021
yeast3	1:8	0.264 ± 0.069	0.190 ± 0.069	0.412 ± 0.104	0.430 ± 0.085
ecoli-0-1-vs-5	1:11	0.800 ± 0.187	0.900 ± 0.122	0.800 ± 0.244 [1]	0.750 ± 0.158 [1,2]
shuttle-c0-vs-c4	1:14	1.000 ± 0.000	1.000 ± 0.000	0.661 ± 0.179	0.984 ± 0.019
yeast6	1:41	0.000 ± 0.000	0.057 ± 0.114	0.000 ± 0.000	0.000 ± 0.000
yeast4	1:28	0.000 ± 0.000	0.020 ± 0.040	0.000 ± 0.000	0.000 ± 0.000
yeast-0-2-5-7-9-vs-3-6-8	1:9	0.421 ± 0.189	0.711 ± 0.310	0.351 ± 0.135	0.160 ± 0.135
vowel0	1:10	0.589 ± 0.284 [4]	0.489 ± 0.288 [4]	0.611 ± 0.185 [4]	0.600 ± 0.183

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

Table 3. Results for mean ranks achieved by analyzed methods with all metrics included in the evaluation.

Metric	1 SVC	2 CWS	3 CMV	4 CSA
balanced accuracy	2.500	3.250	1.867	2.383
	–	*all*	–	–
F1-score	2.733	3.217	1.767	2.283
	3	3,4	–	–
G-mean	2.467	3.283	1.900	2.350
	–	*all*	–	–
precision	3.000	2.783	1.900	2.317
	3,4	3	–	–
recall	2.333	3.350	1.917	2.400
	–	*all*	–	–
specificity	2.817	2.017	2.533	2.633
	2	–	–	–

These observations become even clearer when we look at the results of the ranking tests presented in Table 3. In the case of *balanced accuracy* and *G-mean*, the CWS method is statistically significantly better than in all other cases. For the *F1-score* metric, despite the numerical advantage, the statistical significance of the base method disappears, which is due to the symmetry of the *F1-score* metric relative to problem classes, which accepts the equal cost of a wrong decision to the minority and majority class.

The design of the recognition algorithm dedicated to imbalanced data is almost always based on the calibration of factors measurable by the base classification metrics. As in the case of the CWS method, we try to increase the *recall* so that the inevitable reduction of *precision* or *specificity* will further give us a significant statistical advantage in the chosen aggregate metric, selected to define the cost of the incorrect classification that is relevant to us. In this case, the CWS method turns out to be much better than the other methods of *EoC* integration with a pool diversified by clusters and allows a statistically significant improvement of the base method in the case of highly imbalanced data.

5 Conclusions

This paper presented a new clustering and weighted scoring algorithm dedicated to constructing *EoC*. We proposed that the scoring function should take into account the distance from the decision boundary of each base classifier and the cluster centroids necessary to learn this classifier. In the proposed weighting scoring function the distance to the decision boundary and sum of the distances to the centroids have the same weight. The proposed approach applies to imbalanced datasets because each cluster centroid can be calculated regardless of the number of objects in this cluster.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

Comprehensive experiments are presented on thirty examples of highly imbalanced datasets. The obtained results show that the proposed algorithm is better than other algorithms in the context of statistical tests and some performance measures. In particular, in the case of the *balanced accuracy* and *G-mean* classification measures,the proposed in this paper method is statistically significantly better than all others methods used in the experiments.

A possible future work is to be considered: other distance measures to calculate the distance to cluster centroids, the impact of the use of heterogeneous base classifiers in the proposed method of building *EoC* or another scoring weighting method. In particular, we suggest that the distance from the decision boundary can be scaled or weighting factors regarding the distance of the object from the decision boundary and the distance from the cluster centroids can be introduced. Additionally, we can assign weights for cluster centroids depending on the number of objects that were used to determine these centroids.

Acknowledgements

This work was supported by the Polish National Science Centre under the grant No. 2017/25/B/ST6/01750 as well as by the statutory funds of the Department of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Science and Technology.

References

1. Abdallah, A., Maarof, M.A., Zainal, A.: Fraud detection system: A survey. Journal of Network and Computer Applications **68**, 90–113 (2016)
2. Abdulhammed, R., Faezipour, M., Abuzneid, A., AbuMallouh, A.: Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. IEEE sensors letters **3**(1), 1–4 (2018)
3. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing **17** (2011)
4. Alpaydin, E.: Introduction to machine learning. MIT press (2014)
5. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: In Proceedings of 19th International Conference on Machine Learning (ICML-2002. Citeseer (2002)
6. Burduk, R.: The proposal of calculation classifier weights for an assembly of classifiers. Journal of Medical Informatics & Technologies **23** (2014)
7. Cao, X., Wu, C., Yan, P., Li, X.: Linear svm classification using boosting hog features for vehicle detection in low-altitude airborne videos. In: Image Processing (ICIP), 2011 18th IEEE International Conference on. pp. 2421–2424. IEEE (2011)
8. Chorań, M., Pawlicki, M., Kozik, R.: Recognizing faults in software related difficult data. In: International Conference on Computational Science. pp. 263–272. Springer (2019)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

9. Fotouhi, S., Asadi, S., Kattan, M.W.: A comprehensive data level analysis for cancer diagnosis on imbalanced data. Journal of biomedical informatics **90**, 103089 (2019)
10. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **42**(4), 463–484 (2011)
11. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications **73**, 220–239 (2017)
12. Hajdu, A., Hajdu, L., Jonas, A., Kovacs, L., Toman, H.: Generalizing the majority voting scheme to spatially constrained voting. IEEE Transactions on image processing **22**(11), 4182–4194 (2013)
13. Klikowski, J., Ksieniewicz, P., Woźniak, M.: A genetic-based ensemble learning applied to imbalanced data classification. In: International Conference on Intelligent Data Engineering and Automated Learning. pp. 340–352. Springer (2019)
14. Kozik, R., Choras, M., Keller, J.: Balanced efficient lifelong learning (b-ella) for cyber attack detection. J. UCS **25**(1), 2–15 (2019)
15. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence **5**(4), 221–232 (2016)
16. Krawczyk, B., Woźniak, M., Schaefer, G.: Cost-sensitive decision tree ensembles for effective imbalanced classification. Applied Soft Computing **14**, 554–562 (2014)
17. Ksieniewicz, P., Zyblewski, P.: stream-learn–open-source python library for difficult data stream batch analysis. arXiv preprint arXiv:2001.11077 (2020)
18. Kuncheva, L.I.: Combining pattern classifiers: methods and algorithms. John Wiley & Sons (2004)
19. Mao, S., Jiao, L., Xiong, L., Gou, S., Chen, B., Yeung, S.K.: Weighted classifier ensemble based on quadratic form. Pattern Recognition **48**(5), 1688–1706 (2015)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
21. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: ADVANCES IN LARGE MARGIN CLASSIFIERS. pp. 61–74. MIT Press (1999)
22. Rahman, A.F.R., Alam, H., Fairhurst, M.C.: Multiple classifier combination for character recognition: Revisiting the majority voting system and its variations. In: International Workshop on Document Analysis Systems. pp. 167–178. Springer (2002)
23. Rokach, L.: Pattern classification using ensemble methods, vol. 75. World Scientific (2010)
24. Ruta, D., Gabrys, B.: Classifier selection for majority voting. Information fusion **6**(1), 63–81 (2005)
25. Sun, Y., Wong, A.K., Kamel, M.S.: Classification of imbalanced data: A review. International journal of pattern recognition and artificial intelligence **23**(04), 687–719 (2009)
26. Szeszko, P., Topczewska, M.: Empirical assessment of performance measures for preprocessing moments in imbalanced data classification problem. In: IFIP International Conference on Computer Information Systems and Industrial Management. pp. 183–194. Springer (2016)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10

27. Wozniak, M.: Hybrid classifiers: methods of data, knowledge, and classifier combination, vol. 519. Springer (2013)
28. Woźniak, M., Graña, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Information Fusion **16**, 3–17 (2014)
29. Zhang, C., Bi, J., Xu, S., Ramentol, E., Fan, G., Qiao, B., Fujita, H.: Multi-imbalance: An open-source software for multi-class imbalance learning. Knowledge-Based Systems **174**, 137–143 (2019)
30. Zia, M.S., Hussain, M., Jaffar, M.A.: A novel spontaneous facial expression recognition using dynamically weighted majority voting based ensemble classifier. Multimedia Tools and Applications **77**(19), 25537–25567 (2018)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50423-6_10