Imbalanced Data Classification Based on Feature Selection Techniques

Paweł Ksieniewicz(✉) and Michał Woźniak

Department of Systems and Computer Networks,
Wrocław University of Science and Technology, Wrocław, Poland
{pawel.ksieniewicz,michal.wozniak}@pwr.edu.pl

Abstract. The difficulty of the many classification tasks lies in the analyzed data nature, as disproportionate number of examples from different class in a learning set. Ignoring this characteristics causes that canonical classifiers display strongly biased performance on imbalanced datasets. In this work a novel classifier ensemble forming technique for imbalanced datasets is presented. On the one hand it takes into consideration selected features used for training individual classifiers, on the other hand it ensures an appropriate diversity of a classifier ensemble. The proposed method was tested on the basis of the computer experiments carried out on the several benchmark datasets. Their results seem to confirm the usefulness of the proposed concept.

Keywords: Machine learning · Classification
Imbalanced data · Feature selection · Random search

1 Introduction

Most of classifier training methods assume that the numbers of objects from each classes are roughly equal in a learning set. However, in many real-life decision tasks this assumption is not fulfilled. We may deal with examples from classes being abundant and easy to collect and with the classes where number of examples is small and hard to access [1,11]. Therefore, there is a need of constructing effective predictive systems which can take into consideration imbalanced data distributions [3]. Let us present shortly the main groups of algorithms in imbalanced data classification.

Data Preprocessing. Such methods modify the learning set, before a classifier is being trained [12]. They should manipulate learning examples to obtain a balanced dataset. One may achieve this by either removing samples from the majority classes (*undersampling*), or adding new object from the minority ones (*oversampling*). One have also mention techniques of dimensionality reduction as feature selection which may be also applied to this task [4].

Algorithm-Level Methods. They modify the classifier learning procedure to take into consideration imbalanced data distributions. Usually, they use non-symmetric loss-function [7] to assign higher cost to the error committed on

H. Yin et al. (Eds.): IDEAL 2018, LNCS 11315, pp. 296–303, 2018.
https://doi.org/10.1007/978-3-030-03496-2_33

minority class objects. Another approaches employ one-class classifier learning techniques, where a given class is learned only and the objects which do not belong to it are treated as outliers.

Hybrid Solutions. They are trying to exploit the strengths of the previously discussed methods and to combine them with other techniques. Usually ensemble learning is used [13], which is able to train set of diverse individual predictors, which may take into consideration the data imbalance and propose such combination rule which can make a high quality decision on complex data.

In this paper, we introduce a novel hybrid technique that employs feature selection techniques to train a pool of individual classifiers used by a classifier ensemble. To avoid the overfitting a regularization techniques are used which on the one hand ensures that the pool of classifier is diverse, i.e., subsets of features should be different for each classifier and on the other hand the number of features used by all individuals should be as small as possible. To train the pool we use simple techniques based on random search, but experimental study carried out on a number of benchmarks prove that the proposed method is able to return satisfactory performance.

2 Proposed Algorithm

Selecting appropriate set of the feature is an important data preprocessing step in classifier learning [8] and Chawla et al. [4] underlined its crucial role when classification model is train on the basis of imbalanced data. Traditional feature selection techniques usually use criterion based on the accuracy with a factor responsible for regularization, i.e., discourages learning too complex model to avoid the overfitting. While the methods dedicated for imbalanced data [9,14] usually employs metrics related with binary problem, as *g-mean* [6].

To present the proposed solution, firstly let us formulate the classification problem.

2.1 Problem Formulation

Classifier Ψ makes a decision by assigning an observed object into one of predefined classes derived from the set of possible labels $\mathcal{M} = \{1, 2, ..., M\}$ [7]. Each object is described by the set of attributes (features) gathered in the feature vector x belonging to d dimensional feature space $\mathcal{X}$

$$x = [x^1, x^2, \ldots, x^d]^T \in \mathcal{X} \subseteq \mathcal{R}^d, \tag{1}$$

The aim of a feature selection algorithm is to choose k valuable features only to avoid so-called *course of dimensionality* [5].

$$x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \ldots \\ x^{(d)} \end{bmatrix} \rightarrow \overline{x} = \begin{bmatrix} \overline{x}^{(1)} \\ \overline{x}^{(2)} \\ \ldots \\ \overline{x}^{(k)} \end{bmatrix}, \quad k < d \tag{2}$$

Let us also define a pool of individual classifiers

$$\Pi = \{\Psi_1, \Psi_2, \ldots, \Psi_K\} \tag{3}$$

where Ψ_k denotes the k-th elementary classifier. To ensure diversity of the pool we will train the individuals on the basis of the different set of features. Let's propose the following representation of the classifier pool Π as word of bits:

$$\Pi = \left[\left[b_1^1, b_1^2, ..., b_1^d\right] \left[b_2^1, b_2^2, ..., b_2^d\right] ... \left[b_K^1, b_K^2, ..., b_K^d\right]\right] \tag{4}$$

where b_i^j denotes if the jth feature is used by the ith classifier.

2.2 Criterion

In our algorithm we use the following optimization criterion based on *Balanced Accuracy*

$$Q(\Pi) = BAC(\Pi) - \alpha * \frac{no - features(\Pi)}{d} + \beta * \frac{av - Hamming(\Pi)}{d} \tag{5}$$

where $BAC(\Pi)$ denotes balanced accuracy of the classifier ensemble based on the ensemble represented by Π. The first regularization factor $no-features(\Pi)$ is responsible for the number of the selected features, i.e., number of features used by all individual in the ensemble, while the second regularization factor $av-Hamming(\Pi)$ is the average Hamming distance between the words represented individuals in Π. It is a kind of a diversity measure [13], which encourages to select different features by different individuals. α and β are the parameters of the algorithm, which should be set experimentally.

2.3 Algorithm Description

Firstly, the algorithm randomly generates the population of ensembles

$$Population = \{\Pi_1, \Pi_2, ..., \Pi_S\} \tag{6}$$

A size of the *Population* is an input parameter. Its value S is set arbitrary, but we have to take into consideration, that on the one hand the larger S guarantees the more comprehensive optimization, but on the other hand, the larger S requires the higher computational effort.

Individuals in the population are evaluated by criterion Eq. 5 calculated on the basis of Algorithm 1 using samples stored in the training set. We decide to select the best evaluated ensemble only.

The *combination rule* of chosen ensemble is carried out by *averaging* the *support vectors* received from the members of a pool. It is important, that for such approach, it is necessary to use a *probabilistic classification model*. Three combination rules are proposed for further analysis:

1. **R**—basic accumulation of support without weighing the committee members.

Algorithm 1. Criterion count

Input: pool of individual classifiers Π, training set $\mathcal{TS}$
Parameters: α, *beta*
Output: value of criterion 5 for Π

$counter \leftarrow 0$
$nobits \leftarrow 0$
$word \leftarrow [00..0]$
for $i \leftarrow 1$ **to** $K-1$ **do**
  for $j \leftarrow i$ **to** K **do**
    $counter \leftarrow counter + 1$
    $nobits \leftarrow$ nobits + number of bits of $\left[b_i^1, b_i^2, ..., b_i^d\right] XOR \left[b_j^1, b_j^2, ..., b_j^d\right]$
  end for
  $word \leftarrow wordOR \left[b_i^1, b_i^2, ..., b_i^d\right]$
end for
$word \leftarrow wordOR \left[b_K^1, b_K^2, ..., b_K^d\right]$
$no - features \leftarrow$ number of bits in $word * \frac{1}{d}$
$av - Hamming \leftarrow \frac{nobits}{counter*d}$
$BAC \leftarrow$ balanced accuracy of Π calculated on $\mathcal{TS}$
$criterion \leftarrow BAC - \alpha * no - used - features + \beta * av - Hamming - dist$
return *criterion*

2. **W**— weighted aggregation, where weights are proportional to balanced accuracy values achieved by individual classifiers.
3. **N**—weighted aggregation, where weights are proportional to balanced accuracy values achieved by individual classifiers and additionally weights are subjected to *MinMax* scaling.

3 Experimental Study

Experimental investigations, backed up with statistical analysis of the results, were conducted to evaluate the practical usefulness of the proposed strategy. In the remainder of this section we describe set-up of the study, present obtained results and discuss achieved outcomes.

3.1 Set-Up

For the experimental evaluation of the proposed method, a series of benchmark datasets available on the KEEL repository [2] were used. Selection was made to ensure wide scope of 35 binary problems with *Imbalance Ratio* IR varying from 1 to around 40. The overview of chosen datasets, informing about their IR and number of features, was included in Table 1.

To allow a reliable comparison of literature methods, datasets from KEEL repository are pre-divided into folds. It led to employ *k-fold cross-validation* with 5 folds in the experimental procedure. Due to strong bias of regular classification metrics towards majority class, to ensure reliable results, all scores are presented

as *balanced accuracy*, according to its implementation from the development version (0.20.dev0) of the *scikit-learn* library [10].

Implementation of the experimental procedure, as well as the implementation of the method itself, has been prepared according to the *scikit-learn* library API, using *Gaussian Naive Bayes* as a base classifier. Besides the variations of a method, to provide a comparative result, each problem was also evaluated on a full-featured representation of a dataset. To analyze a paired dependency between the classifiers outputs, the signed-rank *Wilcoxon* test was employed.

The implementation of the method proposed in following paper, as well as the script allowing to reconstruct conducted research can be found in repository[1].

3.2 Results

First step of experimental evaluation was optimization procedure to obtain the best α and β values in the context of *balanced accuracy*. It has been conducted with a *Grid Search* approach, analyzing 7 values evenly dividing the range from 0 to 1. Example visualizations of results for three datasets are presented on Fig. 1. Presentation for all datasets is available at website[2].

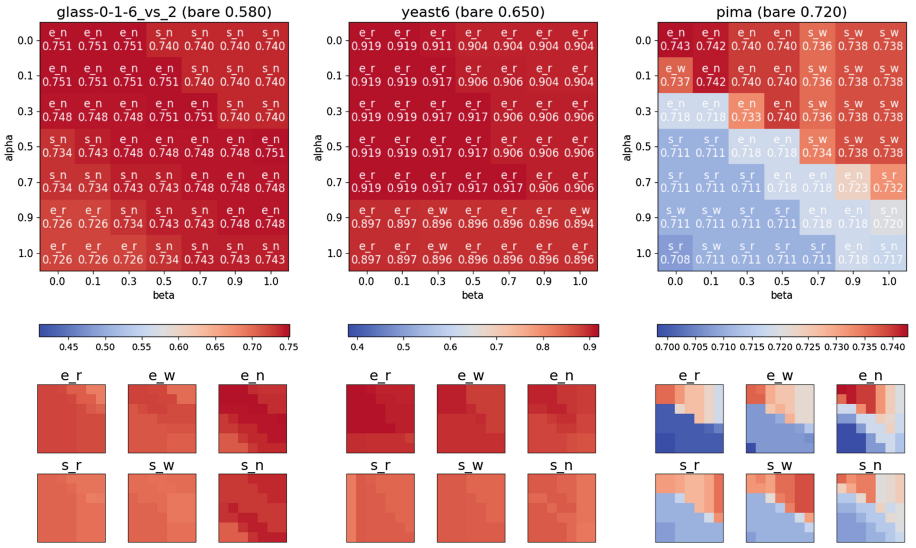

Fig. 1. Examples of α and β influence on classification quality for best (top) and every (bottom) approach. Blue indicates result worse than full-featured classification, red – a better result. (Color figure online)

[1] https://github.com/w4k2/ideal2018.
[2] http://w4k2.github.io/ideal2018.

Table 1. Balanced accuracy scores obtained with the optimized hyperparameters α and β on datasets selected to experimental evaluation. **Full** stands for the results of the classifier using all features, **Ensemble** stands for results of classifier ensemble using different combination rules described in Sect. 2.3, and **Best in ensemble** stands for balance accuracy of the best individual in the ensemble.

Dataset	IR	F.	Params.		Balanced Accuracy Scores						
					Full	Ensemble			Best in ensemble		
			α	β		E_R	E_W	E_N	S_R	S_W	S_N
australian	1	14	.0	.0	0.777	0.852	0.855	0.877	0.878	0.876	0.861
heart	1	13	.0	.0	0.838	0.870	0.878	0.874	0.847	0.847	0.868
glass0	2	9	.3	.5	0.700	0.746	0.746	0.749	0.750	0.750	0.763
glass1	2	9	.1	.0	0.671	0.721	0.719	0.710	0.738	0.723	0.732
pima	2	8	.0	.0	0.720	0.736	0.737	0.743	0.731	0.732	0.736
wisconsin	2	9	.0	.0	0.969	0.976	0.976	0.976	0.967	0.967	0.974
yeast1	2	8	.0	.0	0.519	0.695	0.699	0.680	0.654	0.659	0.660
glass0123vs456	3	9	.0	.0	0.869	0.891	0.898	0.910	0.900	0.910	0.910
hepatitis	5	19	.0	.0	0.687	0.880	0.903	0.877	0.872	0.872	0.881
glass6	6	9	.0	.0	0.891	0.939	0.942	0.945	0.945	0.959	0.959
yeast3	8	8	.0	.0	0.605	0.904	0.895	0.915	0.841	0.840	0.813
glass015vs2	9	9	.0	.0	0.519	0.711	0.728	0.765	0.691	0.696	0.710
glass04vs5	9	9	.0	.0	0.994	0.994	0.994	0.994	0.994	0.994	0.994
yeast0256vs3789	9	8	.0	.3	0.670	0.689	0.689	0.737	0.753	0.748	0.771
yeast02579vs368	9	8	.0	.0	0.577	0.912	0.911	0.900	0.878	0.878	0.894
yeast0359vs78	9	8	.0	.0	0.557	0.668	0.662	0.621	0.607	0.600	0.607
yeast05679vs4	9	8	.0	.0	0.504	0.780	0.763	0.720	0.706	0.702	0.710
yeast2vs4	9	8	.3	.0	0.561	0.897	0.887	0.892	0.838	0.904	0.885
glass016vs2	10	9	.0	.0	0.580	0.726	0.726	0.751	0.705	0.700	0.731
vowel0	10	13	.5	.3	0.917	0.898	0.914	0.911	0.924	0.929	0.933
glass0146vs2	11	9	.1	.0	0.577	0.747	0.761	0.739	0.724	0.746	0.773
glass06vs5	11	9	.0	.0	0.945	0.995	0.995	0.995	0.960	0.960	0.995
glass2	12	9	.0	.0	0.591	0.767	0.775	0.747	0.718	0.721	0.721
shuttlec0vsc4	14	9	.0	.0	0.991	1.000	1.000	1.000	1.000	1.000	1.000
glass4	15	9	.1	.0	0.587	0.609	0.609	0.718	0.768	0.766	0.753
pageblocks13vs4	16	10	.0	.0	0.763	0.786	0.866	0.949	0.867	0.879	0.928
glass016vs5	19	9	.0	.0	0.941	0.991	0.991	0.989	0.989	0.989	0.989
shuttlec2vsc4	20	9	.0	.0	0.996	1.000	1.000	1.000	0.996	0.996	1.000
yeast1458vs7	22	8	.0	.0	0.547	0.592	0.588	0.574	0.556	0.556	0.569
glass5	23	9	.0	.0	0.938	0.988	0.988	0.988	0.988	0.988	0.988
yeast2vs8	23	8	.0	.0	0.657	0.799	0.810	0.774	0.774	0.774	0.774
yeast4	28	8	.0	.0	0.551	0.817	0.783	0.797	0.679	0.670	0.651
yeast1289vs7	31	8	.0	.0	0.544	0.683	0.701	0.706	0.629	0.628	0.606
yeast5	33	8	.0	.0	0.831	0.963	0.964	0.973	0.954	0.947	0.945
yeast6	41	8	.0	.0	0.650	0.919	0.905	0.903	0.821	0.857	0.858

The results of the evaluation after optimization procedure are presented in Table 1, which has been divided to present a *balanced accuracy* obtained on different variations of the method, using whole ensemble or just its best member,

according to the different fusers (R – regular, W – weighted and N – normalized weights). Such division led to the number of 6 analyzed approaches.

Scores for the method were supplemented by the quality of a single model trained on a whole possible feature space. The green color in table indicates the statistical dependency to the best result and underline – the highest *balanced accuracy* obtained on a given dataset.

The presented results clearly showed that feature selection plays important role for imbalanced data classification. Our proposition usually outperforms the results obtained by the classifier using the whole set of features. It also behaves better (22 out of 35 datasets) than the best classifier in the pool. It has been probably caused by very naive optimization method (random search) used in this work.

4 Conclusions and Future Directions

The novel hybrid classification method for imbalanced data classification was presented. It employs ensemble learning to increase performance (*balanced accuracy*) of the combined classifier. To ensure the appropriate level of diversity, each individual is trained on the basis of selected features. Nevertheless, in contrast with well-known methods based on randomly chosen features (as *Random Subspaces*), the choice of the features is the results of the optimization procedure. The optimization criterion takes into consideration not only the performance of the classifier, but to protect against *overfitting* it encourages to build the ensemble of diverse individuals which do not use too many features. Additionally, we observed that the proposed method can significantly outperform the classifier based on whole set of features.

As the future works we are going to use more sophisticated optimization procedure based on genetic approach, but we also realize that it will negatively impact computational complexity, therefore we will focus on the method which can be run in distributed computing systems as GPU or SPARK. Additionally, we plan to definitely extend the scope of experiments to compare our methods with other methods as *Random Subspaces* or *Decision Forrest*.

Acknowledgments. This work was supported by the Polish National Science Center under the grant no. UMO-2015/19/B/ST6/01597 as well as Statutory Found of the Faculty of Electronics, Wroclaw University of Science and Technology.

References

1. Ahmed, F., Samorani, M., Bellinger, C., Zaïane, O.R.: Advantage of integration in big data: feature generation in multi-relational databases for imbalanced learning. In: 2016 IEEE International Conference on Big Data, BigData 2016, Washington DC, USA, 5–8 December 2016, pp. 532–539 (2016)
2. 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. J. Multiple-Valued Logic Soft Comput. **17** (2011)

3. Branco, P., Torgo, L., Ribeiro, R.P.: A survey of predictive modeling on imbalanced domains. ACM Comput. Surv. **49**(2), 1–50 (2016)
4. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor. Newsl. **6**(1), 1–6 (2004)
5. Domingos, P.: A few useful things to know about machine learning. Commun. ACM **55**(10), 78–87 (2012)
6. Du, L.M., Xu, Y., Zhu, H.: Feature selection for multi-class imbalanced data sets based on genetic algorithm. Ann. Data Sci. **2**(3), 293–300 (2015)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)
8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
9. Maldonado, S., Weber, R., Famili, F.: Feature selection for high-dimensional class-imbalanced data sets using support vector machines. Inf. Sci. **286**, 228–246 (2014)
10. 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. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
11. Porwik, P., Doroz, R., Orczyk, T.: Signatures verification based on PNN classifier optimised by PSO algorithm. Pattern Recogn. **60**, 998–1014 (2016)
12. Triguero, I., Galar, M., Merino, D., Maillo, J., Bustince, H., Herrera, F.: Evolutionary undersampling for extremely imbalanced big data classification under apache spark. In: IEEE Congress on Evolutionary Computation, CEC 2016, Vancouver, BC, Canada, 24–29 July 2016, pp. 640–647 (2016)
13. Wozniak, M., Graña, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. Inf. Fusion **16**, 3–17 (2014)
14. Yin, L., Ge, Y., Xiao, K., Wang, X., Quan, X.: Feature selection for high-dimensional imbalanced data. Neurocomputing **105**, 3–11 (2013)