Optical Music Recognition: Standard and Cost-Sensitive Learning with Imbalanced Data

Wojciech Lesinski, Agnieszka Jastrzebska

Optical Music Recognition: Standard and Cost-Sensitive Learning with Imbalanced Data

Wojciech Lesinski[1] and Agnieszka Jastrzebska[2]

[1]Faculty of Mathematics and Computer Science, University of Bialystok
ul. Konstantego Ciolkowskiego 1M, 15-245 Bialystok
and
[2]Faculty of Mathematics and Information Science, Warsaw University of Technology
ul. Koszykowa 75, 00-662 Warsaw, Poland

Abstract. The article is focused on a particular aspect of classification, namely the issue of class imbalance. Imbalanced data adversely affects the recognition ability and requires proper classifier's construction. In this work we present a case of music notation as an example of imbalanced data. Three classification algorithms - random forest, standard SVM and cost-sensitive SVM are described and tested. Feature selection based on random forest feature importance was used. Also, feature dimension reduction using PCA was studied.
Keywords: cost-sensitive learning, SVM, random forest, feature selection, optical music recognition, imbalanced data

1 Introduction

The problem of pattern recognition is an important area of data mining, which has been studied and developed for many years now. In a number of its applications, satisfying results have already been achieved. However, in many fields it is still possible to obtain better results. Among many other important issues of this domain, studies on class imbalance have gained popularity.

Recently, the class imbalance problem has been recognized as a crucial obstacle in machine learning and data mining. It occurs when the training data is not evenly distributed among classes. Class imbalance is also especially critical in many real applications, such as credit card fraud detection when fraudulent cases are rare or medical diagnoses where normal cases are the majority. In these cases, standard classifiers generally perform poorly. Classifiers usually tend to be biased towards the majority class and ignore the minority class samples. Most classifiers assume an even distribution of examples among classes and an equal misclassification cost. Moreover, classifiers are typically designed to maximize accuracy, which is not a good metric to evaluate effectiveness in the case of imbalanced training data. Therefore, we need to improve traditional algorithms so as to handle imbalanced data and choose other metrics to measure performance instead of accuracy.

Most of the publications concerning imbalanced data focus on binary classification problems, for example [4], [5]. Far less articles (inter alia [1], [15]) address multiclass problems.

Automatic recognition and classification of music notation is a case of optical character recognition. It may have many applications. First and foremost, digital versions of musical scores popularise and simplify access to art. Availability of digital music notation and dedicated processing tools widens possibilities not only, but also for learning and appreciation of music. It also contributes to limitation of barriers for those, who experience difficulties in accessing standard, printed music notation, for example those, who are visually impaired. With electronic record of music notation we can make an attempt to computerize musical synthesis, we can also, by using the voice synthesizer, read music scores. Electronic music notation could also be used to verify the performance correctness of the musical composition, and to detect potential plagiarism. These applications lead to the conclusion that the optical recognition of music notation is an interesting and worthy research topic. General methodology of optical music recognition has been already researched and described in [6] and [13]. We would like to highlight that studied problem of imbalance of classes is an original contribution of this paper to the field of music symbol classification. In particular, application of cost-sensitive learning is a new element, not yet present in the literature devoted to optical music recognition.

The paper is organized as follows. Section 2 lists basic information about applied classification technique. Section 3 describes our data set, feature vector and empirical tests. Section 4 concludes the paper and indicates future research directions.

2 Preliminaries

2.1 Support Vector Machine

Support Vector Machine [14] (SVM) is a statistical classification method that forms a separation hyperplane with maximized margin between classes. The optimal margin is necessary to support generalization ability. Because some classification problems are too complicated to be solved by a linear separation, kernel functions are used to transform data set into a new one where such separation is possible. Classification decision is given by the function:

$$f(X) = \sum_{i=1}^{M} \alpha_i y_i K(X_i, X) + b \tag{1}$$

where M is the number of learning samples, y_i is the class label assigned to features vector X_i. Parameters α_i and b are calculated in learning process. Coefficients α_i are calculated as the solution of QP problem limited by the hypercube $[0, C]^{dimension}$. C is the capacity constant that controls error. The larger the C, the more the error is penalized. Function K is a kernel function.

In our work we use cost-sensitive SVM. The main approach to cost-sensitive learning is rescaling. Rescaling [3], is a general approach that can be used to make cost-blind learning algorithms cost-sensitive. The principle is to enable

influences of the higher - cost classes to be larger than that of the lower - cost classes. The rescaling approach can be realized in many ways, for example by assigning to training elements of different classes different weights, sampling the classes according to their costs or moving the decision threshold.

2.2 Feature selection using Random Forest

In machine learning feature selection, also known as variable selection, attribute selection or variable subset selection, is a process of selecting a subset of relevant features for use in model construction. The central assumption when using any feature selection technique is that the data contains redundant or irrelevant features. Redundant features are those, which provide no more information than the currently selected feature set, and irrelevant features provide no useful information in any context.

In the case of feature selection a criterion function J(.) is optimized:

$$\begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_d \end{bmatrix} \rightarrow \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_d} \end{bmatrix} = argmax[J(\{x_i | i = 1, \ldots, D\})] \tag{2}$$

Where D is the dimension of the whole feature set and d is the dimension of given feature subset. Unfortunately, comprehensive search of all possible subsets of D is usually impossible due to computational complexity of such endavour. Therefore, algorithms that approximate function J(.) are applied. We can distinguish two groups of such algorithms: filter and wrapper methods.

Wrapper methods use a predictive model to score feature subsets. Each new subset is used to train a model, which is tested on a hold-out set. Counting the number of mistakes made on that hold-out set (the error rate of the model) gives the score for that subset. As wrapper methods train a new model for each subset, they are very computationally intensive, but usually provide the best performing feature set for that particular type of model. An example of wrapper methods is random forest feature selection.

Filter methods use a proxy measure instead of an error rate to score a feature subset. This measure is chosen to be fast to compute, whilst still capturing the usefulness of the feature set. Common measures include the Mutual Information, Pearson product-moment correlation coefficient, and the inter/intra class distance. Filters are usually less computationally intensive than wrappers, but they produce a feature set which is not tuned to a specific type of predictive model. Many filters provide a feature ranking rather than an explicit best feature subset selection, and the cut off point in the ranking is chosen via cross-validation.

The Random Forest [2] method uses a collection of decision tree classifiers, where each tree in the forest has been trained using a bootstrap sample of individuals from the data, and each split attribute in the tree is chosen from

among a random subset of attributes. Classification of individuals is based upon aggregate voting over all trees in the forest. Each tree in the random forest is built as follow:

- Let the number of training objects be N, and the number of features in features vector be M.
- Training set for each tree is built by choosing N times with replacement from all N available training objects.
- Number $m << M$ is an amount of features on which to base the decision at that node. This features are randomly chosen for each node.
- Each tree is built to the largest extent possible. There is no pruning.

Repetition of this algorithm yields a forest of trees, which all have been trained on bootstrap samples from training set. Thus, for a given tree, certain elements of training set will have been left out during training.

Prediction error and attribute importance is estimated from these "out-of-bag" elements. This part of training set is used to estimate the importance of particular attributes according to the following logic: if randomly permuting values of a particular attribute does not affect the predictive ability of trees on out-of-bag samples, that attribute is assigned a low importance score. If, however, randomly permuting values of a particular attribute drastically impairs the ability of trees to correctly predict the class of out-of-bag samples, then the importance score of that attribute will be high. By running out-of-bag samples down entire trees during the permutation procedure, attribute interactions are taken into account when calculating importance scores, since class is assigned in the context of other attribute nodes in the tree.

2.3 Reduction of problem dimensionality - Principal Component Analysis

One of dimension reduction algorithms is called Principal Component Analysis (PCA). PCA is a statistical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or in the worst-case-scenario equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of original variables.

2.4 Evaluation of solution

Evaluation of classification methods applied to imbalanced pattern recognition problem is the principal goal of this research. First of all, classification quality

from a perspective of single classes is considered. We adopt parameters of binary classification evaluation and parameters and quality measures used in signal detection theory. Since these parameters are widely utilized, we do not refer to original sources, but of course we do not claim to be their authors. In this point we recall employment of these factors in an imbalanced two-class problem, c.f. [4].

Two-class problem We share an opinion that evaluation of just a single factor cannot truly express classification quality. This is valid in general, as well as in the two-class problem. For instance, it is not only important to account the proportion of the number of correctly recognized symbols of a class to the number of all symbols of this class. Let us point out that, for example, the number of symbols falsely accounted to this class affects intuitive meaning of quality. Especially, when we consider a class of small number of elements, falsely classified symbols significantly decrease intuitive evaluation of quality. Therefore, we should look for formal evaluations compatible with intuition. Let us recall that in the case of imbalanced two-class problem, the minority class is often called positive one while majority class - negative one.

Such intuitive measures, as indicated above, provide a simple way of describing classifier's performance on a given data set. However, they can be deceiving in certain situations and are highly sensitive to changes in data. For example, consider a problem where only 1% of the instances are positive. In such situation, a simple strategy of labelling all new objects as members of other classes would give a predictive accuracy of 99%, but failing on all positive cases. In [4] the following confusion matrix was used in evaluating classification quality of a two-class problem, c.f. Table 1. Parameters presented in this table were then used to define several factors, which outline classification quality.

	Positive prediction	Negative prediction
Positive class	True Positives (TP)	False Negatives (FN)
Negative class	False Positives (FP)	True Negatives (TN)

Table 1. Confusion matrix for *two-class* problem

Multiclass problem For a better classification quality measuring, let us first consider the following parameters of a *multiclass problem*. The parameters given in Table 2 are numbers of elements of a testing set, which have the following meaning:

- TP - the number of elements of the considered class correctly classified to the considered class,
- FN - the number of elements of the considered class incorrectly classified to other classes,
- FP - the number of elements of other classes incorrectly classified to the considered class,
- TN - the number of elements of other classes correctly classified to other classes (no matter, if correctly, or not).

	Classification to the considered class	Classification to other classes
The class	True Positives (TP)	False Negatives (FN)
Other classes	False Positives (FP)	True Negatives (TN)

Table 2. Confusion matrix for a *multiclass* problem

Using parameters showed in Table 1 and Table 2 we can calculate some measures valid to evaluate performance even when we deal with imbalanced data:

$$\begin{aligned} Sensitivity &= \frac{TP}{TP+FN} \\ Miss\ Rate &= \frac{FN}{TP+FN} = 1 - Sensitivity \end{aligned} \tag{3}$$

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+FN+FP+TN} \\ Error &= \frac{FP+FN}{TP+FN+FP+TN} = 1 - Accuracy \end{aligned} \tag{4}$$

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ False\ Discovery\ Rate &= \frac{FP}{TP+FP} = 1 - Precision \end{aligned} \tag{5}$$

3 Experiment and Results

3.1 The data set

The recognized set of music notation symbols had about 27.000 objects in 20 classes. There were 12 classes defined as numerous and each of them had

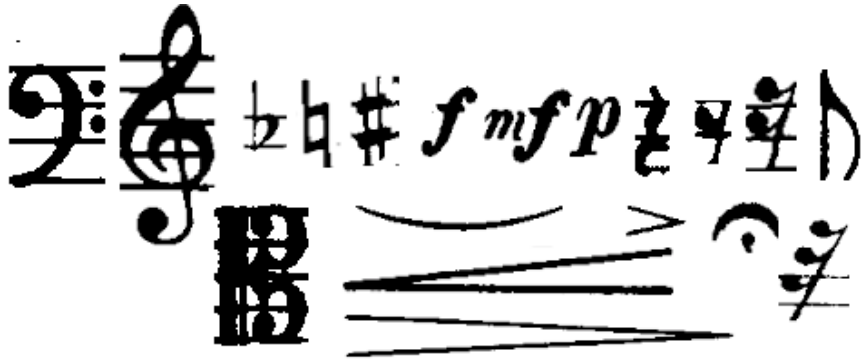

Fig. 1. Symbols being recognized: 1) numerous classes in the upper row, left to right: clefs (F and G), chromatic symbols (flats, naturals and sharps), dynamic markings (forte, mezzo forte and piano), rests (quarter, eight, sixteenth), flagged stem, 2) rare symbols in the bottom row, left to right, top to down: clef C, tie (arc), crescendo, diminuendo, accent fermata, 32nd rest.

about 2.000 representatives. Cardinality of the other eight classes was much lower and various in each of them. Part of the examined symbols was cut out from chosen Fryderyk Chopin's compositions. Other part of the symbols' library comes from our team former research projects [6]. The following elements form regular classes: flat, sharp, natural, G clef, F clef, forte, piano, mezzo forte, quarter rest, eight rest, sixteenth rest, and flagged stem. The set of irregular classes includes breve note, accent, crescendo, diminuendo, tie, fermata, C clef and thirty-second rest were included. Each regular class in training sets consisted of 400 elements. Cardinality of irregular classes is shown in Table 3. Recognized symbols are illustrated in Figure 1.

class	learning set	testing set
accent	30	65
breve	1	2
crescendo	55	100
diminuendo	51	97
fermata	35	46
clef C	100	178
tie	100	155
thirty-second rest	20	35

Table 3. Cardinality of learning and testing sets for irregular classes

3.2 Feature vector

Performance of every classifier is conditioned by an appropriate description of objects used for its training. In the case of image classification feature vectors describe objects subjected to recognition. Many publications on pattern recognition propose to use different features extracted from an image. Those are: histograms, transitions, margins, moments, directions and many more. It may seem that creating a vector of all known features would be the best solution. Unfortunately such vector could cause a huge load of computation and in turn, it might result in an unacceptable consumption of resources. On the other hand, often adding more features does not increase classifier efficiency.

In this study we used a group of known features often used in optical character recognition. These were: projections, transitions, margins, directions, regular moments, central moments, Zernike moments, field and perimeter of symbol, Euler's vector and other. For accurate description of those features please see [9].

We have experimented with different approaches to feature vector construction. In the first step possibly biggest vector, which included all features, was created. It had 300 components. Next, we employed random forest to evaluate importance of features. Feature vectors comprising of most important features were constructed. We studied feature sets of the following cardinalities: 200, 150, 100, 75, 50, 40, 30, 25, 20, and 15. Lastly, PCA was studied. We used it with 99%, 95% and 90% transferred variance.

feature vector	random forest	SVM	cost-sensitive SVM
whole set	96	96	95
best 200	96	96	95
best 150	97	96	95
best 100	97	97	95
best 75	98	97	96
best 50	98	98	97
best 40	98	98	97
best 30	97	98	96
best 25	96	96	95
best 20	95	95	93
best 15	93	92	90
PCA 99%	94	93	91
PCA 95%	94	95	91
PCA 90%	93	93	91

Table 4. The influence of different feature vectors on recognition accuracy (in percent).

3.3 Results

Three different classifiers were used to perform prediction. These were: random forest, SVM and cost-sensitive SVM (with weights). To evaluate the classifiers three measures were calculated: sensitivity, accuracy and precision. For these calculations our multi class problem was turned to m two-class problems (*one class contra all others*). All measures were calculated for each class. Average measure was determined in the end. Also, standard error and accuracy (Formulas 6 and 7) were calculated. In our previous works we have elaborated on application of decision tree, kNN, bagging, and other methods. One may consult [8], [12] and [10] to compare different techniques.

$$accuracy = \frac{\text{number of all correctly classified}}{\text{number of all elements}} \tag{6}$$

$$error = 1 - accuracy \tag{7}$$

Standard measures The best prediction rate was achieved, when we employed feature selection procedures. Optimal number of characteristics were 50 and 40. For these vectors random forest and SVM obtained 98%. These two classifiers had similar efficiency. Slightly worse results were achieved by SVM with weights. Feature vectors obtained using PCA gave worse results. All results are gathered in Table 4.

Accuracy Accuracy informs about the influence of given class on the whole testing set. The best accuracy was in minority classes. Accuracy of classifiers without weights in the case of the breve note was 99.99%. In the case of SVM with weights it was a little worse. In contrast, sensitivity in this class was 0%! Other rare classes were also recognized with good accuracy. It varied between 99.89% and 99.99%. The worst accuracy was in natural and sharp classes. These classes had relatively poor sensitivity and had many elements in testing set. Similar results were noted in the rest group (quarter, eighth , sixteenth, thirty-second rest). Rests belonging to regular classes were classified with better accuracy, but worse than rare ones. Accuracy of cost-sensitive SVM was a little worse, especially in rare classes. Figure 2 summarizes the results for all classes.

Sensitivity Sensitivity shows the recognition effectiveness in the given class. The highest value of this factor, 100%, was obtained in forte, mezzo-forte and piano classes. Random forest and SVM reached high values of this factor for all regular classes. Among the rare classes the best sensitivity was achieved in the C clef class. The worst sensitivity (0%) was in breve note class. This symbol was not recognized by any classifier. Cost-sensitive SVM reached better sensitivity for rare classes. Results for all classes and all classifiers are illustrated on Figure 3.

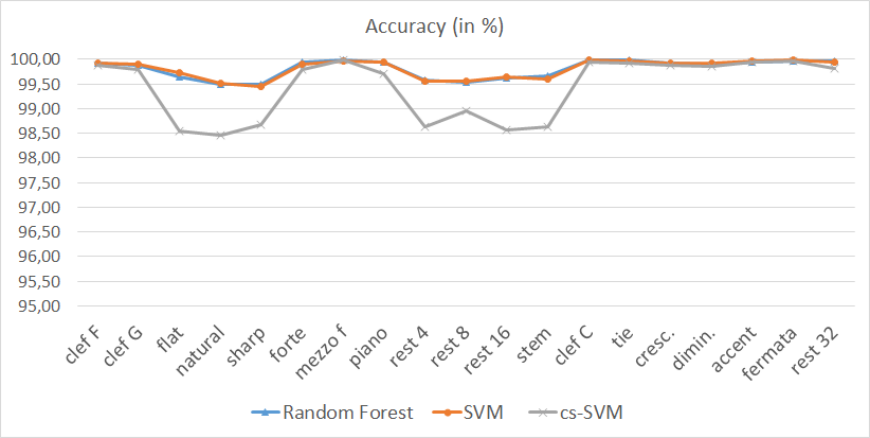

Fig. 2. Accuracy (in percent) for all classes

Precision Precision shows the influence of other classes on a given class. The highest values of this measure were in classes with dynamics symbols and clefs' classes. When we applied random forest and SVM we achieved better precision for regular classes than for rare ones. The worst precision were in crescendo, diminuendo and thirty-second rest classes. This measure was significantly worse for cost-sensitive SVM for rare classes. Figure 4 shows precision for all classes.

4 Conclusion

The problem of pattern recognition for imbalanced data was tackled in this paper on an example of music notation symbols. Authors presented results of classification experiments performed with standard and cost-sensitive classifiers on a dataset consisting of 27.000 elements of 20 classes. 12 of them have been considered as regular classes, the other 8 as irregular classes.

The recognition effectiveness of regular classes was very satisfying. Results obtained by random forest and SVM were similar. Cost-sensitive SVM achieved better sensitivity for rare classes, but all in all they were worse than other factors. We may conclude this study by saying that better recognition of rare classes causes worse recognition of regular classes.

In addition, different versions of feature selection procedures were tested. In particular, for this study feature selection based on random forest feature importance and PCA were used. Best results were achieved by random forest feature selection. Vector with 50 most important features, selected from 300 features, gave the best recognition accuracy performance.

Despite the fact of high efficiency of the proposed techniques, we believe that better results can be achieved. In the next step of our research we will take a closer look at other cost-sensitive classifiers. Also another sets of algorithms for feature selection will be tested. Conducted studies indicate that a vital issue for

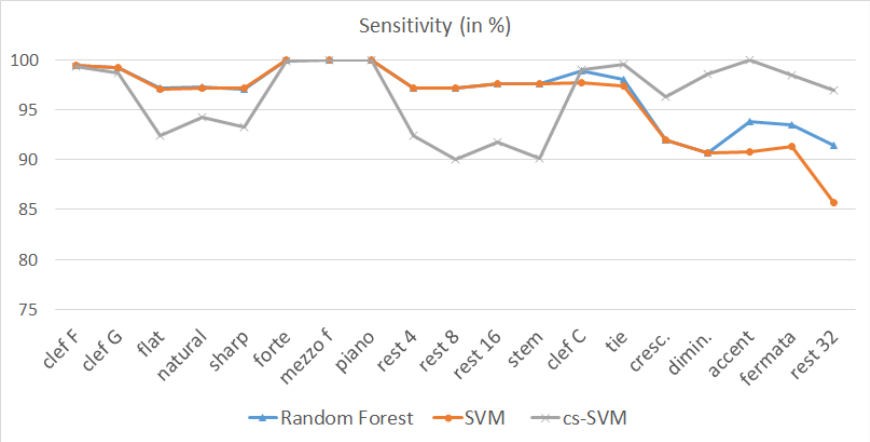

Fig. 3. Sensitivity (in percent) for all classes

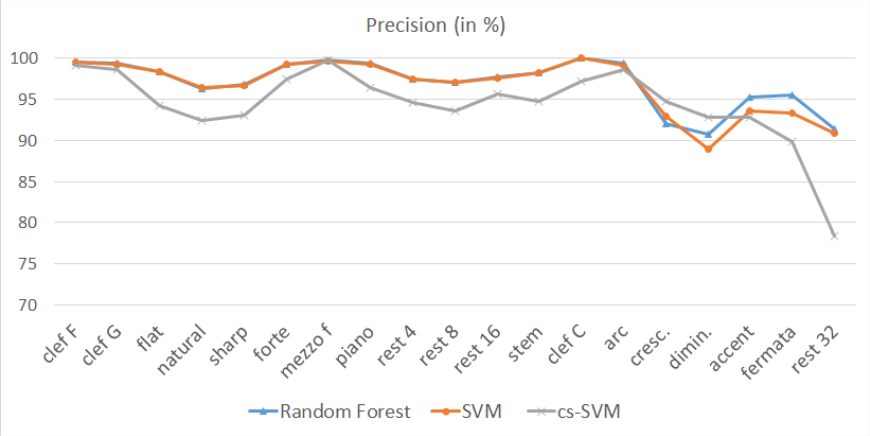

Fig. 4. Precision (in percent) for all classes

pattern recognition tasks are "garbage" elements. Existence of garbage elements is often due to errors that occur at the point of image segmentation. There is an urgent need for methods that deal with rejection of such garbage. This issue is not easy though, as garbage elements are not know at the point of classifier construction. We come towards conclusion that involvement of imprecise knowledge representation schemes, for example balanced fuzzy sets, [7] or fuzzy sets [11] in classification might help us to deal with this problem.

References

1. Abe N., Zadrozny B., Langford J., An Iterative Method for Multi-Class Cost-Sensitive Learning, in: Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, pp. 3-11, 2004.
2. Breiman L., Random Forests, in: Machine Learning 45, pp. 5-32, 2001.
3. Elkan C., The foundations of cost-sensitive learning, in: Proc. of the 17th International Joint Conference on Artificial Intelligence, Seattle, pp. 973-978, 2001.
4. Garcia V., Sanchez J. S., Mollineda R. A., Alejo R., Sotoca J. M. The class imbalance problem in pattern recognition and learning, in: II Congreso Espanol de Informatica, pp. 283 - 291, 2007.
5. He, H., Garcia, E. A., Learning from imbalanced data, in: IEEE Transactions on Knowledge and Data Engineering 21(9), pp. 1263-1284, 2009.
6. Homenda W., Optical Music Recognition: the Case Study of Pattern Recognition, in: Computer Recognition Systems. Springer Verlag, pp. 835-842, 2005.
7. Homenda W., *Balanced Fuzzy Sets*, in: Information Sciences 176, pp. 2467-2506, 2006.
8. Homenda W., Lesinski W., Optical Music Recognition: Case of Pattern recognition with Undesirable and Garbage Symbols, in: Image Processing and Communications Challenges - Choras R. et al (Eds.), Exit, Warsaw, pp. 120-127, 2009.
9. Homenda W., Lesinski W., Features Selection in Character Recognition with Random Forest Classifier, in: Lecture Notes In Artificial Intelligence 6922, vol. 1, Springer Verlag Berlin-Heidelberg, pp. 93-102, 2011.
10. Homenda W., Lesinski W., Decision Trees and Their Families in Imbalanced Pattern Recognition: Recognition with and without Rejection, in: Saeed K. et al (Eds.), Lecture Notes in Computer Science Volume 8838, pp. 219-230, 2014.
11. Homenda W., Pedrycz W., Processing of uncertain information in linear space of fuzzy sets, in: Fuzzy Sets & Systems 44, pp. 187-198, 1991.
12. Lesinski W., Jastrzebska A., Optical Music Recognition as the Case of Imbalanced Pattern Recognition: a Study of Single Classifiers, in: Skulimowski A. M. J. (Ed.) Proceedings of KICSS'2013, Progress & Business Publishers, Krakow, pp. 267-278, 2013.
13. Rebelo A., Fujinaga I., Paszkiewicz F., Marcal A. R. S., Guedes C., Cardoso J. S., Optical music recognition: state-of-the-art and open issues, in: International Journal of Multimedia Information Retrieval 1, pp. 173 - 190, 2012.
14. Vapnik V., The nature of statistical learning theory, Springer-Verlag, 1995.
15. Zhou Z. H., Liu X.Y., On Multi-Class Cost-Sensitive Learning, in: Computational Intelligence 26, pp. 232-257, 2010.