Decision Trees and Their Families in Imbalanced Pattern Recognition: Recognition with and without Rejection

Wladyslaw Homenda[1] and Wojciech Lesinski[2]

[1] Faculty of Mathematics and Information Science, Warsaw University of Technology
Plac Politechniki 1, 00-660 Warsaw, Poland
[2] Faculty of Mathematics and Computer Science, University of Bialystok
ul. Sosnowa 64, 15-887 Bialystok, Poland

Abstract. Decision trees are considered to be among the best classifiers. In this work we use decision trees and its families to the problem of imbalanced data recognition. Considered are aspects of recognition without rejection and with rejection: it is assumed that all recognized elements belong to desired classes in the first case and that some of them are outside of such classes and are not known at classifiers training stage. The facets of imbalanced data and recognition with rejection affect different real world problems. In this paper we discuss results of experiment of imbalanced data recognition on the case study of music notation symbols. Decision trees and three methods of joining decision trees (simple voting, bagging and random forest) are studied. These methods are used for recognition without and with rejection.

Keywords: pattern recognition, decision tree, bagging, random forest, optical music recognition, imbalanced data.

1 Introduction

A decision tree is a graph that uses a branching method to illustrate every possible outcome of a decision. It is powerful and popular tool for classification and prediction. Decision trees appeared in the literature in the context of sociological research in the sixties. In the field of machine learning decision trees appeared thanks to the work of Braiman and Quinlan. Currently they are regarded as one of the best classifiers' type. Decision trees connect the high speed of action with high efficiency [12].

A data set is called imbalanced if it contains many more samples from one class than from the rest of the classes. Data sets are unbalanced when at least one class is represented by only a small number of training examples (called the minority class) while other classes make up the majority. In this scenario, classifiers can have good accuracy on the majority class (or classes in multi-class problem) but very poor accuracy on the minority class(es) due to the influence that the larger majority class has on traditional training criteria. Most original classification algorithms pursue to minimize the error rate: the percentage of the

K. Saeed and V. Snášel (Eds.): CISIM 2014, LNCS 8838, pp. 219–230, 2014.

incorrect prediction of class labels. They ignore the difference between types of misclassification errors. In particular, they implicitly assume that all misclassification errors cost equally. For example, we will consider the two-class problem in which 99% of the objects belongs to the prevalent class. In this case, if we include all of the tested elements to this class, this will result in a very high, 99% efficiency.

Most of the publications concerning imbalanced data focus on the two-classes problem (e.g. [5] and [6]). Far less articles (inter alia [1], [16]) pertains to multi-class problems. In our study we focus attention on the multi-class issue. We have chosen the symbols of music notation as the example of this problem. Music notation recognition problem is imbalanced one because of three features: cardinality, shape and size. In this work, we mainly discuss class sizes with some attention given to other features, which also affect the classification effectiveness.

Automatic recognition and classification of music notation is a case of Optical Character Recognition. It may have many applications. This is primarily a music scores backup. Electronic processing of acquired information could be another application. With electronic record of music notation we can attempt to computerize music synthesis, we can also, by using the voice synthesizer, read this music score for the need of blind and visually impaired. Electronic music notation could also be used to verify the performances 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 [7] and [14]. We would like to highlight, that studied problem of imbalance of classes is an original contribution to the field of music symbols classification. The aim of our study is to investigate how decision trees and its families deal with imbalanced data. The research is based on actual opuses. Applied classification algorithms have been implemented in C++. Developed program works with both high and low-resolution images of musical symbols.

2 Theoretical Background

2.1 Decision Trees

A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

To classify an example, filter it down the tree, as follows. For each feature encountered in the tree, the arc corresponding to the value of the example for that feature is followed. When a leaf is reached, the classification corresponding to that leaf is returned.

Popular algorithms used for construction of decision trees have inductive nature using use top-down tree building scheme. In this scheme, building a tree starts from the root of the tree. Then, a feature for testing is chosen for this node and training set is divided to subsets according to values of this feature. For each value there is a corresponding branch leading to a subtree, which should be created on the basis of the proper testing subset. This process stops when a stop criterion is fulfilled and current subtree becomes a leaf. An example algorithm of tree construction is described in the next section.

Stop criterion shows when construction process needs to be brought to a standstill, that is when for some set of samples we need to make a leaf, not a node. An obvious stop criterion could be situation when:

- a sample set is empty,
- all samples are from the same class,
- attributes set is empty.

In practice criteria given above sometimes bring over-fitting to learning data. So then another stop criteria or mechanisms, such as pruning, is necessary to be applied in order to avoid the over-fitting problem.

Finally, classification of a given object is based on finding a path from the root to a leaf along branches of the tree. Choices of branches are done by assigning tests' results of the features corresponding to nodes. The leaf ending the path gives the class label for the object [4], [12].

2.2 ID3 Algorithm

ID3 (Iterative Dichotomiser 3) is an algorithm used to generate a decision tree. The algorithm was invented by Ross Quinlan [15]. This algorithm uses entropy as a test to divide training set. Entropy for a given set X split to classes $C_1, C_2 ... C_M$ is as follows:

$$entropy(X) = -\sum_{i=1}^{M} p_i(log(p_i)) \tag{1}$$

where $P = (p_1 ... p_M)$ are appearance probabilities of objects from $C_1, C_2, ..., C_M$ classes.

Average entropy in a given node v and for an attribute A_l is defined as follows:

$$avg\_entropy(X_v) = \sum_{i=1}^{k_l} \frac{|T_i|}{|X_v|} * entropy(T_i) \tag{2}$$

where $(T_1, T_2, ..., T_{k_l})$ is a division of the training subset X_v corresponding to the node v attribute A_l, T_i includes testing elements of the subset X_v, which have the value a_{li} of the attribute A_l, and k_l is the number of values of the attribute A_l.

The algorithm ID3 is based on information entropy and can be formulated as follows:

1. put the testing set in the root of the decision tree,
2. if for a given node of the tree all samples belong to the same class C_i, then the node becomes the leaf labelled by the class C_i,

3. if for a given node the attribute set is empty, then the node becomes the leaf labelled by the class C_i having majority in the testing subset in this node,
4. if for a given node the attribute set is not empty and samples in the testing set are not in the same class, then:
   - compute average entropy for each attribute,
   - choose an attribute with minimal entropy,
   - split the testing subset according to values of the chosen attribute,
   - for every set of the split: create the successor of the node and put the set in this node,
   - apply points 2, 3 and 4 for newly created nodes.

2.3 Families of Trees

Ensembles of classifier join computational capabilities of single classifiers and allow to build diverse models. In the case of conjunction methods, classifier is created with a number of other classifiers. Classifiers, which we use for connecting, we can call *weak classifiers*. Depending on the purpose, we may compose a model consisting of various single classifiers, but we may also manipulate with distinct parameters. There are also different ways of model construction. Two of those methods (simple voting and bagging) also may joined other classifiers, but in this work we use them for decision trees connecting.

Simple Voting. Simple voting is one of the simplest conjunction methods. We can use any weak classifiers in this method. Classifiers can be already trained, or they can be in the phase of training. The way of training is also not imposed. The only condition of start-up of this algorithm is having weak classifiers, which are statistically independent from each other. The sample $x \in X$ is tested by every weak classifier, then an answer is counted as a sum. The class which is indicated by largest number of weak classifiers, is chosen as the right one.

Bagging. Bagging, a name derived from "bootstrap aggregation", devised by Breiman [2], is one of the most intuitive and simplest ensemble algorithm providing good performance. It improve the stability, accuracy, reduces variance and avoid over-fitting of machine learning algorithms. Although it is usually applied to decision tree methods, it can be used with any type of method.

Bagging uses multiple versions of a training set, each created by drawing $n < N$ (where N is a number of elements of original training set) samples from training set D with replacement. Each of bootstrap data sets is used to train a different component classifier and the final classification decision is based on the vote of each component classifier. Traditionally the component classifiers are of the same general form - for example, all Hidden Markov models, or all neural networks, or all decisions trees - merely the final parameter values differ among them due to their different sets of training patterns.

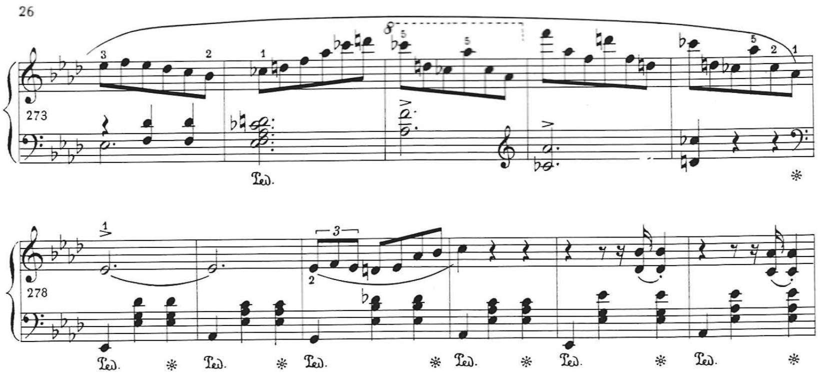

Fig. 1. Example of music score

Random Forest. Random forest is a relatively new classifier proposed by Breiman in [3]. The method combines Breiman's [2] "bagging" idea and the random selection of features in order to construct a collection of decision trees with controlled variation.

Random forest is composed of some number of decision trees. Each tree 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 is randomly chosen for each node.
- Each tree is built to the largest extent possible. There is no pruning.

Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

2.4 Classifiers Evaluation

To evaluate the classifiers in imbalanced data problem we starts from confusion matrix given i Table 1. The parameters given in the matrix 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 this 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,

Table 1. Confiusion matrix for *two classes* problem

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

- TN - the number of elements of other classes correctly classified to other classes (no matter, if correctly, or not).

In this study we consider multi class problem. Hence, parameters of *two classes problem* are turned to *one class contra all others*. Finally, three measures were used assess the quality of the classifier:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

3 Experiment

3.1 Data Set

The recognized set of music notation symbols had about 27.000 objects in 20 classes and about 3000 foreign symbols, which not belong to the recognized classes. There were 12 classes defined as numerous and each of them had about 2.000 representatives. Cardinality of other eight classes was much lower and various in each of them (see Table 2). Part of the examined symbols was cut from chosen Fryderyk Chopin's compositions. Other part of the symbols' library comes from research projects [18] and [19]. Example of music score is on Figure 1.

Classes were divided into two groups: regular and rare classes. Regular classes include flat, sharp, natural, G and F clefs, piano, forte, mezzo-forte, quarter rest, eight rest, sixteenth rest and flagged stem. Irregular classes consist of accent, breve note, C clef, crescendo, diminuendo, fermata, tie and thirty-second rest. As mentioned, images sets coming from regular classes consisted of 2000 objects each. Sets of irregular classes are significantly smaller.

Set of foreign symbols includes various symbols from music scores, letters, digits and objects accidentally cuts from scores (unspecified parts of symbols and staves).

Table 2. Learning and testing sets for irregular classes

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

3.2 Features Vector

In this work experts features selection was used. Features selection based on previous works [9], [10] and [11]. Our features vector counted 50 elements. Features vector includes:

- maximal values and its positions and average values for projections, transitions and margins
- 3 regular moments
- 3 central moments
- 4 Zernike moments
- directions of 45, 90, 135 and 180 degrees
- field of symbol
- symbol's perimeter.

3.3 Recognition without Rejection

To evaluate the classifiers three measures was calculated: sensitivity, accuracy and precision. For this calculations our multi class problem was turned to m two class problems (*one class contra all others*). All measures was calculated for each class. In the end average measure was determined. In simple voting decision tree, k-Nearest Neighbor and classifier with Mahalanobis minimal distance were joined. In bagging 10 trees were constructed. In the case of random forest also 10 trees were joined and 5 features were drafted for splint in each node.

Accuracy. Accuracy shows influence of given class on whole testing set. The best accuracy was in small classes. Accuracy of all classifiers in breve note was 99.99 percent. For comparison sensitivity in this class was 0%! Other rare classes also had a good accuracy. It was between 99.89% and 99.99%. The worst accuracy was in natural and sharp classes. This classes had relatively poor sensitivity and had many elements in testing set. Related results were in rest group. Rest of regular classes were achieved better accuracy, but worse than rare ones. Results for all classes and all classifiers are shown on Figure 2.

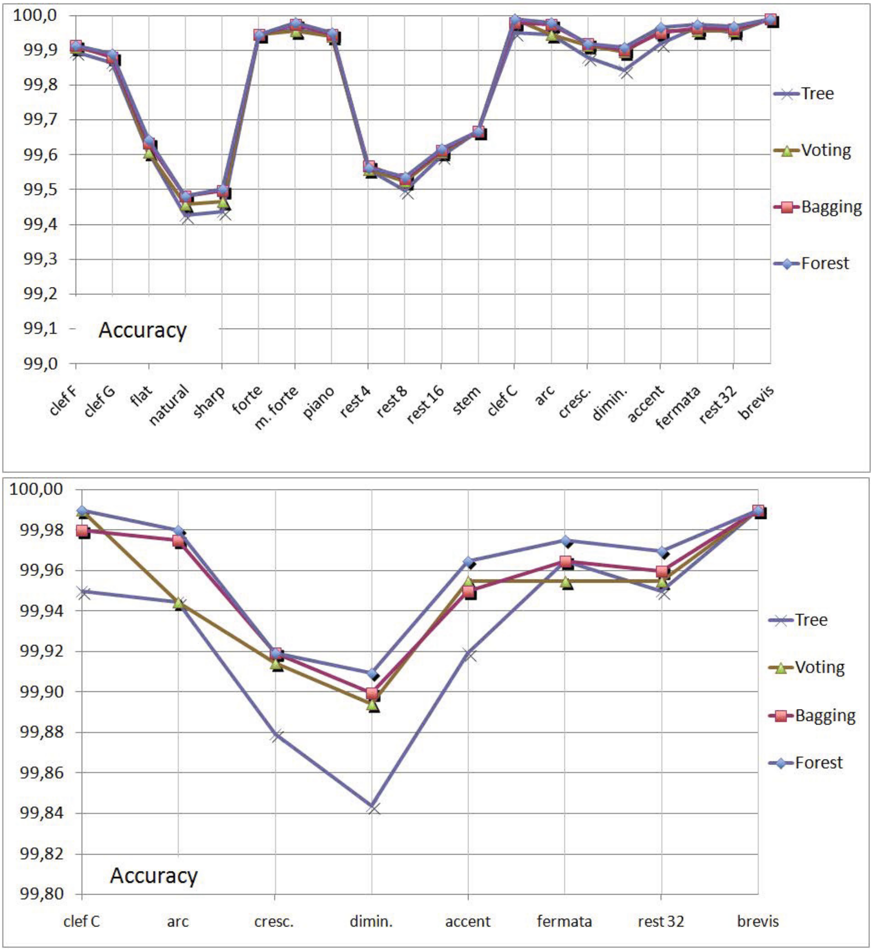

Fig. 2. Accuracy for all and rare classes

As same as in the case of sensitivity the best average accuracy was obtained by random forest classifier. A little worse average accuracy (98.48%) was reached by bagging. All classifiers had a very good values of accuracy. This measure does not show influence of rare classes. Therefore it seems improper measure for this problem.

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. All regular classes reached a high values of this factor. Among the rare classes best sensitivity was achieved in C clef. The worst (0%) was in breve note class. This symbol was not recognized by any classifier. Results for all classes and all classifiers are shown on Figure 3.

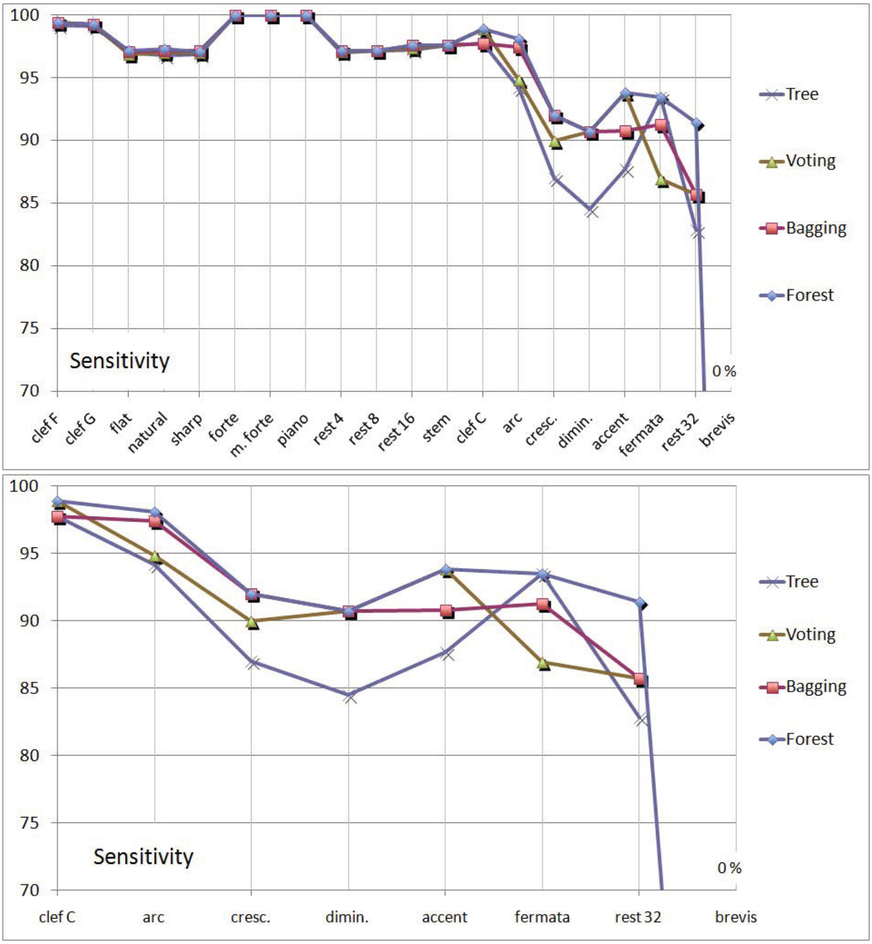

Fig. 3. Sensitivity for all and rare classes

Best average sensitivity was obtained by random forest classifier. It was 91.91%. Bagging, simple voting and decision tree also reached good performances. Breve note had big influence on this factor. If we calculate it without this class, it was obtained 96.74 for random forest classifier.

Precision. Precision shows how other classes influenced on given class. The highest values of this measure were in dynamics symbols classes and clefs classes. Precision for regular classes was better than for rare ones. The worst precision were in crescendo, diminuendo and thirty-second rest. For breve not this factor was undetermined, because TP and FP was equal 0 in this case. Results for all classes and all classifiers are shown on Figure 4.

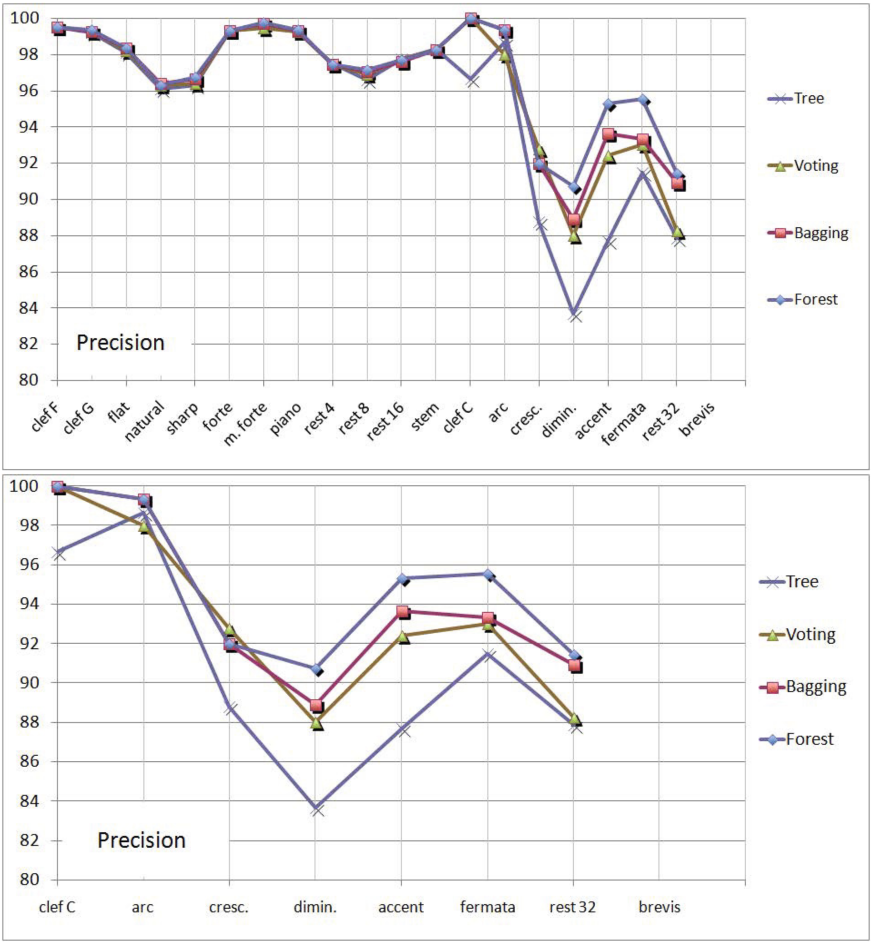

Fig. 4. Precision for all and rare classes

The best average precision was achieved by random forest classifier. It was 97.03 percent. Bagging had a little worse values of this factor (96.68%). Simple voting and decision tree also obtained a good results.

3.4 Recognition with Rejection

Pattern recognition problem not always has accurate symbols' extraction stage. Segmentation and extraction steps often produce many extraordinary undesirable symbols and ordinary garbage. We can call them *foreign symbols* in contrast to *native symbols* of recognized classes, c.f. [8]. In such a case a classification module, which assigns all extracted symbols to designed classes, will produce misclassification for every undesirable symbol and for every garbage symbol.

Improvements of classification require construction of such classifiers which could assign designed symbols to correct classes and reject undesirable and garbage symbols.

In our paper we treated undesirable and garbage symbols as representatives of one or more adding classes. In this case original set of classes M was increased by adding set of rejected classes M_R. The effectiveness of rejection was defined as the ratio of correct rejected symbols to the all symbols from rejected classes. In other words it was sensitivity of rejected classes.

In first experiment all rejected symbols were in one class. That was the worst way of rejection. In this case only 38 percent of symbols from undesired class was correctly rejected. Rest of this symbols were mistakenly classified to other class. Most of it was misclassified to the regular classes. Sensitivity for native classes was like in recognition without rejection, but precision was lower.

In next stage of tests the foreign class has been divided into smaller parts. The process of dividing was made by k-means clustering algorithm. Set of rejected symbols was split on 3, 5, 10 and 20 subclasses. In this way symbols were classified to 23, 25, 30 and 40 classes. The increasing the number of foreign classes caused rejection effectiveness increase. Sensitivity for rejected symbols (TP - correct rejected symbols, FN - incorrectly non rejected symbols) was 62% for 3 foreign classes, 81% for 5 foreign classes, 92% for 10 foreign classes and 94% for 20 foreign classes.

Unfortunately increase the number of foreign classes decreases the effectiveness of recognition of native classes. In the case of 20 foreign classes sensitivity of regular classes decreased by about 5 percent, for rare classes - from 10 to 15 percent.

4 Conclusions

In the paper was discussed the problem of imbalanced image recognition on the example of music notation symbols recognition. Authors present results of classification experiments performed with classifiers using decision trees on a dataset consisting of 27 000 elements of 20 classes.

Simple decision tree obtained efficiency equal 98%. The merger techniques gave slightly better results. This is particularly evident in the case of rare classes. Best results was obtained by random forest classifier. Decision trees and ensembles classifiers based on it were proved to be very effective. Tests for all measures gave good results. The study showed that the symbols of musical notation can be considered as imbalanced data.

Recognition with rejection was tested also. In this case rejected symbols were placed to one or more added classes. Rejection's effectiveness was better when rejected symbols were placed in a large number of classes. Unfortunately, in this case effectiveness of recognition of native symbols was worse.

Acknowledgement. The research is supported by the National Science Center, grant No 2012/07/B/ST6/01501, decision no UMO-2012/07/B/ST6/01501.

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.: Bagging predictors. Machine Learning 26(2), 123–140 (1996)
3. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Inc., New York (2001)
5. 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)
6. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)
7. Homenda, W.: Optical Music Recognition: the Case Study of Pattern Recognition. In: Computer Recognition Systems, pp. 835–842. Springer (2005)
8. Homenda, W., Luckner, M., Pedrycz, W.: Classification with rejection: concepts and formal evaluations. In: Andrzej, M.J. (ed.) Proceedings of KICSS 2013, pp. 161–172. Progress & Business Publishers, Krakow (2013)
9. Homenda, W., Lesinski, W.: Optical Music Recognition: Case of Pattern recognition with Undesirable and Garbage Symbols. In: Choras, R., et al. (eds.) Image Processing and Communications Challenges, pp. 120–127. Exit, Warsaw (2009)
10. 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, pp. 267–278. Progress & Business Publishers, Krakow (2013)
11. Lesinski, W., Jastrzebska, A.: Optical Music Recognition as the Case of Imbalanced Pattern Recognition: A Study of Complex Classifiers. In: Swiątek, J., Grzech, A., Swiątek, P., Tomczak, J.M., et al. (eds.) Advances in Systems Science. Lesinski W., Jastrzebska A, vol. 240, pp. 325–335. Springer, Heidelberg (2014)
12. Koronacki, J., Cwik, J.: Statystyczne systemy uczace sie. Exit, Warszawa (2008) (in Polish)
13. Kuncheva, L.I.: Combining Pattern Classifiers. Methods and Algorithms. John Wiley & Sons (2004)
14. 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. International Journal of Multimedia Information Retrieval 1, 173–190 (2012)
15. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1, 81–106 (1986)
16. Zhou, Z.H., Liu, X.Y.: On Multi-Class Cost-Sensitive Learning. Computational Intelligence 26, 232–257 (2010)
17. `http://www.stat.berkeley.edu/users/breiman/RandomForests/`
18. Breaking accessibility barriers in information society. Braille Score - design and implementation of a computer program for processing music information for blind people, the research project no N R02 0019 06/2009 supported by by The National Center for Research and Development, Poland (2009-2012)
19. Cognitive maps with imperfect information as a tool of automatic data understanding. Ideas, methods, applications, the research project no 2011/01/B/ST6/06478 supported by the National Science Center, Poland (2011-2014)