Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume I: Foundations

Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics Volume II: Applications

Editors

Krassimir T. Atanassov
Michał Baczyński
Józef Drewniak
Janusz Kacprzyk
Maciej Krawczak
Eulalia Szmidt
Maciej Wygralak
Sławomir Zadrożny

Editors

Krassimir T. Atanassov
Władysław Homenda
Olgierd Hryniewicz
Janusz Kacprzyk
Maciej Krawczak
Zbigniew Nahorski
Eulalia Szmidt
Sławomir Zadrożny

SRI PAS IBS PAN

Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics
Volume II: Applications

Systems Research Institute
Polish Academy of Sciences

Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics
Volume II: Applications

Editors

Krassimir T. Atanassov
Władysław Homenda
Olgierd Hryniewicz
Janusz Kacprzyk
Maciej Krawczak
Zbigniew Nahorski
Eulalia Szmidt
Sławomir Zadrożny

IBS PAN **SRI PAS**

Systems Research Institute
Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
www.ibspan.waw.pl

ISBN 9788389475305

Study of classifiers conjunction in optical music recognition

Wladyslaw Homenda and Wojciech Lesinski
Institute of Computer Science, University of Bialystok
Sosnowa 64, 15-887 Bialystok, Poland
homenda@ii.uwb.edu.pl, wojtek_lesinski@o2.pl

Abstract

Recognition with use of only one classifier, in many tasks may occur not enough efficient. We can improve that by classifier conjunction. This paper describes few algorithms of classifier conjunction. These conjunct classifiers were used to recognize chosen music notation symbols. Classification without rejection and classification on sets containing undesirable and garbage symbols was tested in this experiment.

Keywords: pattern recognition, classification, Optical Music Recognition.

1 Introduction

Automatic classification and image recognition is still unresolved problem. Some of the images, like typed letters are recognized quite well, however in many disciplines a lot of work has to be done, for example handwriting and music notation symbols recognition. To improve efficiency of recognized images we can test many kinds of classifiers, or different features' vector configurations. These tests will tell us, what kind of classifier is most appropriate for the type of recognized image.

Developments in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics. Volume II: Applications *(K.T. Atanassow, W. Homenda, O. Hryniewicz, J. Kacprzyk, M. Krawczak, Z. Nahorski E. Szmidt, S. Zadrożny, Eds.), IBS PAN - SRI PAS, Warsaw, 2009.*

1.1 Classification formalism

In case of optical music notation symbols recognition, the best type of classifier can be pointed out during tests as well. Simple classifiers as well as conjunct classifiers can be used for recognition of these symbols.

In the recognition task the classification algorithm ψ assigns a class number $i \in M$ for every vector of features:

$$\psi \ : \ X \to M$$

Or, equivalently, it divides features' space onto so-called decision regions:

$$D_X^{(i)} = \{x \in X \ : \ \psi(x) = i\} \quad \text{for every} \quad i \in M$$

Stages of features' segmentation and extraction may cause some errors, resulting with undesirable images on the classifier input. When classifier assigns every object to M class set, causes classification errors for every undesirable symbol. This problem can be carried out, when the classifier will have the possibility to reject images. Rejection of symbols can formally be interpreted in terms of a new class, into which all undesirable and garbage symbols fall:

$$D_X^{(0)} = \{x \in X \ : \ \psi(x) = 0\}$$

We call this region the rejection region, while other classes are called acceptation region:

$$D_X(A) = \bigcup_{i \in M} D_X^{(i)}$$

Of course sum of those regions covers all space X and all former classes are pairwise disjoint, c.f. [2, 6].

$$(\forall i \in M) \ \ D_X^{(i)} \cap D_X^{(0)} = \emptyset \quad \text{and} \quad \bigcup_{i \in M} D_X^{(i)} \cup D_X^{(0)} = X$$

In some cases, when using only one classifier is not giving satisfying results, we can improve efficiency by classifiers conjuction. The point is, that tested sample x is recognized by all used classifiers, next the results are compared to adjust only one system response. Usually, the process of joining results is nothing else, but sum of answers with set weights. The main difference is, that classifiers used in this system are trained in different ways.

1.2 Classifier conjunction

The first tested conjunct classifier was **simple voting** method. This is one of the most simple conjunction method. We can use any classifiers components in this methods. Classifiers can be already trained, or stage of training can succeed during system creation. Also the way of training algorithm components is not imposed. The only condition of start-up of this algorithm is having trained 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 most of classifiers, is chosen as the right one. Numbers of modifications of this algorithm can be used. One of the modifications is weights setting, in order to prefer classifiers that have best results. For this we can define helping function $b(x, y)$

$$b : M \times M \longrightarrow \{0, w_j\}$$

where w_j is the classifier j weight. We can define weights, as natural numbers representing, for example classifier number on the list sorted rising according to its effectiveness, and so the least efficient classifier gains weight number 1 and number 1 on the list and so on.

Bagging - a name derived from ”boosting aggregation” - uses multiple versions of a training set, each created by drawing n' ¡ n samples from training set D with replacement. Each of these 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 from - 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. In general, bagging improves recognition for unstable classifiers because it effectively averages over such discontinuities, c.f. [2].

The goal of **boosting** is to improve the accuracy of any given learning algorithm. In boosting we firs create a classifier with accuracy on the training set greater than average, and add new component classifiers to form an ensemble whose joint decision rule arbitrarily high accuracy on the training set. There are number of variations on basic boosting. The most popular, **AdaBoost** - from ”adaptive boosting” - allows the designer to continue adding weak learners until some desire low training error has been achieved. In AdaBoost each training pattern receives a weight that determines its probability of being selected for a training set for an individual component classifier. If a training pattern is accurately classified, then its chance of being used in a subsequent component classifier is reduced; conversely, if the pattern is not accurately classified, then its chance of being used again is raised. In this way, AdaBoost ”focuses in” on the informative or ”difficult” patterns. Specifically, we initialize the weights across the training set to be uniform. On each iteration k, we draw a training set at random according to these weights, and then we train component classifier Ψ_k on the patterns selected. Next we increase weights of training patterns misclassified by Ψ_k decrease weights of the patterns correctly classified by Ψ_k. Patterns chosen according to this new distribution are used to train the next classifier Ψ_{k+1}, and the process is iterated, c.f. [2].

2 Testing experiment

Nine classes of music notation symbols were studied:

- rest 1/4 - symbol means silence that lasts as long as quarter - note
- rest 1/8 - symbol means silence that lasts as long as quaver
- rest 1/16 - symbol means silence that lasts as log as semiquaver
- sharp - symbol which increases diatonic scale about an undertone
- flat - symbol which decreases diatonic scale about an undertone
- natural - symbol which cancels all chromatic sighs
- clef G - it’s shape is similar to letter G and it is used as treble clef.
- clef F - it’s shape is similar to letter F and it is used as bass clef and contrabass clef.
- flagged stem - this symbol is a part of a quaver. Position of this symbol depends on where is placed the head of the note on the stave. If head is placed under a third line, then stem is directed upwards. In situation where head of the note lays above the third line, stem is directed downwards. The flag is always on the right.

The experiment was performed on testing set of symbols listed above. It included 9012 copies of symbols in total and about a thousand copies of every symbol.

Undesired symbols were included in three classes of dynamic symbols: piano, forte and mezzoforte. Each of these classes included a thousand copies of the symbol.
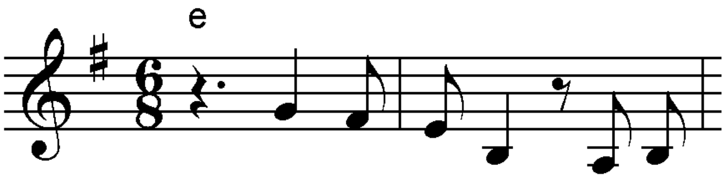
Figure 1: An example of music notation

An extra set of garbage images included 1000 symbols. This set included accidentally cut of parts of music notation as well as different symbols of music notation (notes, chords, C clefs, triplets etc.

Symbols undergo preprocessing before classification. Images in gray-scale are converted to monochromatic images. Conversion is executed with use of defined threshold. Pixels having value below that threshold obtain value 0 and become white, while pixels with value above that threshold become black and obtain value 1. The next stage of normalization is conversion of input rectangular image to image $N \times N$. In this case the sizes are 16×16 or 32×32. The scaling process consists of two stages. In the first stage rectangular image is scaled, in order to give its' longer side the N length. In the second stage white pixels are added to shorter side on both sides of the picture, in order to make the center of gravity cover with center of newly obtained pattern, c.f. [4].

In the recognition process important thing is to define features' element vector. In this case there is no possibility to define the best features' vector. The choice of proper features has to be made in the experimental way. The features used in experiments are:

- histograms,
- projections,
- transitions,
- directions,
- moments,
- margins,
- average 3
- average 5
- difference
- maximum
- minimum

- average
- maximum position
- minimum position
- average position

3 Results

3.1 Recognition without rejection

Simple voting method was tested first of all. The following sizes of training sets were exploited 1, 10, 20, 50, 100, 200 and 400 for every of recognized class. Influence of training set size on classifier efficiency was tested. There were 3 classifiers conjunct:

- k-NN (for k=1)
- k-means (for k=1)
- Naive Bayes

Efficiency in testing singular classifiers was considered when choosing those algorithms. Efficiency of used simple classifiers was described in [4]. For the tests a set of 9012 symbols was used. This is a set of all 9 classes of symbols used in this paper. The bigger the size of training set, the higher the efficiency of classification. For set containing only one element efficiency reached just 60 %. The efficiency for sets of 10 elements was at a level 87%. The training set containing 400 elements gave the best results - 97%.

Another method of classifier conjunction was **bagging**. The sizes of training sets exploited in this test were the same as in simple voting method. The influence of training set size on classifier efficiency was tested. There were 4 classifiers conjunct:

- k-NN (for k=1)
- k-means (for k=1)
- Naive Bayes
- decision tree

Furthermore, combination of both methods mentions above was studied. First of all there were created k classifiers of one kind (e. g. Naive Bayes k classifiers) with bagging method. This way we obtain 4 groups of classifiers with k elements.

Then all $4 * k$ classifiers are joined like in simple voting method, and the system response is nothing else, but response of all of those classifiers.

In case of simple bagging, the worst results were gained for training set containing one element - recognition on testing set reached 60%. Considerable increase of efficiency was obtained for 10 element sets, for which the efficiency reached 90%. For sets containing 400 elements efficiency reached 97%, similar to simple voting method.

The tests of modified bagging method were executed on sets of 9012 symbols for training set containing 20 elements from every of examined classes. The best results were gained when classifiers with efficiency on 97% level were conjunct. Systematic efficiency increase, which can not be noticed in singular groups, is another advantage. From singular groups the most efficient in Naive Bayes, and the least efficient is k-means method. Note: the feature of bagging method is an improvement of component classifiers efficiency, when training sets have small sizes. The fault of this classifier is its time consuming tests of classifier groups. In tests described above the most time consuming was group built from k - Nearest Neighbors classifiers.

AdaBoost algorithm was tested on training sets containing 1, 10, 20, 50, 100, 200, 400 symbols from every class. The influence between set size and the recognition effectiveness of this classifier was examined. These classifiers were utilized in testing:

- k-NN (for k=1)
- k-means (for k=1)
- Naive Bayes
- decision tree

Furthermore, combination of two methods was examined. First of all, by the use of boosting method, there have been created k classifiers of one kind. Next, all groups of classifiers were conjunct like in simple voting method.

In the testing set, best results were gained by 4 classifiers conjunction method, which is a consequence of methods' independence during the conjunction, as it was in boosting aggregation. Note: such good result was gained due to high efficiency of component classifiers, such as decision tree and k - Nearest Neighbors, which gained 97% effect.

We can observe that relation between efficiency and training set size is less noticeable in this method for majority of classifiers. In this method influence of amount of classes is slighter than in bagging method. As it was in tests described above, best results were given by simple voting method based on all used in experiment classifiers. Unfortunately, AdaBoost has not such systematic increase of recognition efficiency like it took place in boosting aggregation.

3.2 Recognition with rejection

The classifiers mentioned above were tested also in classification with rejection process. In this experiment of classification with rejection, three different sets had been added to the set of examined classes:

- set ’dynamic symbols’, which contains three classes representing piano, forte and mezzoforte symbols taken for undesirable elements
- set ’others’ containing different elements which could be found on the stave
- conjunction of both sets mentioned above.

The same classifiers were used in conjunction as, in case of recognition without rejection.

The most efficient was boosting algorithm in recognizing ’dynamic symbols’ set. For this algorithm, the percentage of classification accuracy reached 92, for correctly found reached 91 and for correctly rejected - 97. Slightly worse appeared to be the bagging method, which correctly found 87% of elements, and rejected 97% of elements. The worst results gave simple voting method which found 87% images.

In this experiment the highest efficiency was obtained by boosting aggregation method for rejecting ’others’ set, which reached 97% effectiveness for 10 classes. Set ’others’ was recognized with best results by AdaBoost method, and obtained result on 84% level. Also here, the worst results gave simple voting method with 85% efficiency for 10 classes and reached 68% when recognizing ’Others’ set.

In third examined case both sets ’Dynamic Symbols’ and ’others’ were added to recognized symbols. In this task, the best efficiency was obtained by bagging method, representing 87% efficiency for 10 classes. The best results in recognizing ’Others’ + ’Dynamic Symbols’ set, gave AdaBoost algorithm, gaining the result of 87%.

Additionally, the influence of training set size, on classification with rejection for classifiers conjunction algorithm, was examined. The Figure 2 shows relation between training set size and efficiency of set ’Others ’recognition. Figure 2 presents dependence between classification efficiency and training set size. The same as in case of classification without rejection, in all conjunction methods used in this experiment, the dependence was as following: the bigger the training set

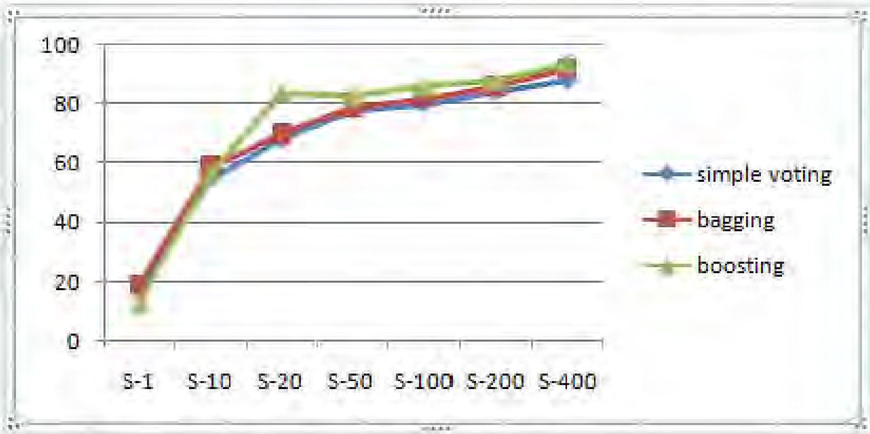

Figure 2: Influence of training set size on recognition efficiency

size, the higher accuracy in recognition of tested symbols. Note: for set $S-400$ the obtained results are approximate to results without an extra class, for bagging and AdaBoost algorithm. For those algorithms, with the increase of training set size, the influence of number of classes on recognition effectiveness is reduced.

For the two other remaining sets, recognition efficiency was increasing, when the training set size was increasing. With the training set size increase, correct recognition of tested symbols was improving, for all conjunction methods mentioned in this paper. The same situation took place in case of classification without rejection. Note: for set $S-400$ obtained results are approximate to results obtained without an extra class, for bagging and AdaBoost algorithm. For those algorithms with the increase of training set size, the influence of number of classes on the recognition efficiency was reduced.

4 Conclusions

The most efficient and statistically independent in recognition process classifiers, were used in conjunction method. The best results from all methods mentioned in this paper, had obtained the AdaBoost algorithm, presenting 97% efficiency, with use of 4 classifiers for conjunction. Unfortunately, this algorithm needs training process modification, which lengthens learning process. The advantage of this method is recognition time, which is approximate to time obtained with single classifier.

Classifiers conjunction gives much better results than singular methods. When constructing those algorithms you have to pay attention to component classifiers. The component classifiers can not use complicated calculations and have to be high efficient. From classification methods mentioned in this paper, the best for conjunction are decision tree and classifiers based on Bayes rule. Algorithms obtained this way are high efficient and fast working. Another advantage of these algorithms is high efficiency when examined set of images is being expanded, how it was in case of dynamic symbols.

In order to attain better results, other classifiers can be used in further experiments or more efficient methods of rejection could be considered. Division of undesirable symbols to many classes seems to be a good idea.

References

[1] Breiman L. (1996), Bagging predictors, in Machine Learning, 26(2), pp. 123-140

[2] Duda R. O., Hart P. E., Stork D. G. (2001) Pattern Classifictation, John Wiley & Sons, Inc

[3] Homenda W. (2005) Optical Music Recognition: the Case Study of Pattern Recognition, in: Computer Recognition Systems, Kurzynski et al (Eds.), pp. 835-842, Springer Verlag

[4] Homenda W., Lesinski W. (2009), Optical Music Recognition:a Case of Pattern Recognition with Undesirable and Garbage Symbols, in: Image Processing and Communications Challenges, pp. 120-127, Exit

[5] Kaminski L. (2006) Application of collaborating pattern recognition methods to music notation recognition (in Polish), MSc Thesis, Warsaw University of Technology

[6] Kurzynski M. (1997) Images recognition statistical methods (in Polish), Oficyna Wydawnicza Politechniki Wroclawskiej

[7] Michalski A. (2006) Application of competitive pattern recognition methods to music notation recognition (in Polish), MSc Thesis, Warsaw University of Technology

[8] Schapire R. E., (1990), The strenght of weak learnability, in Machine Learning, 5(2), pp. 197-227

The papers presented in this Volume 2 constitute a collection of contributions, both of a foundational and applied type, by both well-known experts and young researchers in various fields of broadly perceived intelligent systems.
It may be viewed as a result of fruitful discussions held during the Eighth International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets (IWIFSGN-2009) organized in Warsaw on October 16, 2009 by the Systems Research Institute, Polish Academy of Sciences, in Warsaw, Poland, Centre for Biomedical Engineering, Bulgarian Academy of Sciences in Sofia, Bulgaria, and WIT – Warsaw School of Information Technology in Warsaw, Poland, and co-organized by: the Matej Bel University, Banska Bistrica, Slovakia, Universidad Publica de Navarra, Pamplona, Spain, Universidade de Tras-Os-Montes e Alto Douro, Vila Real, Portugal, and the University of Westminster, Harrow, UK:

http://www.ibspan.waw.pl/ifs2009

The Eighth International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets (IWIFSGN-2009) has been meant to commence a new series of scientific events primarily focused on new developments in foundations and applications of intuitionistic fuzzy sets and generalized nets pioneered by Professor Krassimir T. Atanassov. Moreover, other topics related to broadly perceived representation and processing of uncertain and imprecise information and intelligent systems are discussed.

We hope that a collection of main contributions presented at the Workshop, completed with many papers by leading experts who have not been able to participate, will provide a source of much needed information on recent trends in the topics considered.

ISBN-13 9788389475305
ISBN 838947530-8
9 788389 475305