combining classifier, multiple classifier systems, pattern recognition, fuser design, ensemble design

Michał WOŹNIAK[1]

COMBINING CLASSIFIERS – CONCEPT AND APPLICATIONS

Problem of pattern recognition is accompanying our whole life, therefore methods of automatic pattern recognition is one of the main trend in Artificial Intelligence. Multiple classifier systems (MCSs) are currently the focus of intense research. In this conceptual approach, the main effort is concentrated on combining knowledge of the set of individual classifiers. Proposed work presents a brief survey of the main issues connected with MCSs and provides comparative analysis of some classifier fusion methods.

1. INTRODUCTION

The aim of a recognition task is to classify a given object by assigning it (on the basis of observing the features) to the one of the predefined categories [7]. Some of recognition methods are built on the basis of strong mathematical background, like probabilistic classifiers, but others based on intuition. There are many propositions how to automate the classification process. We could use number of classifiers for each task [4], like classifier presented in Fig. 1.

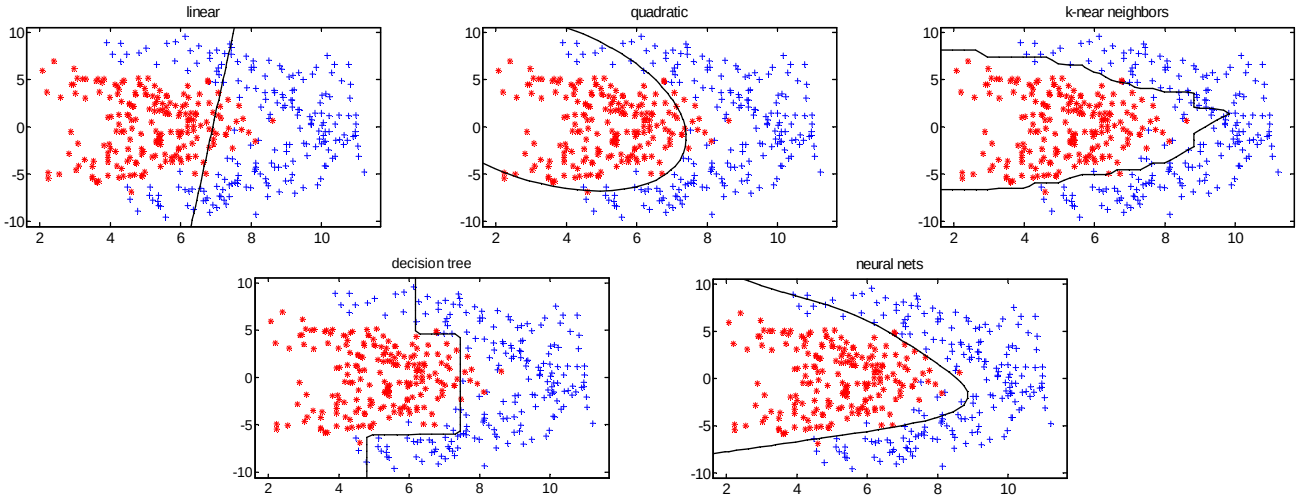


Fig. 1. Different classifiers for a toy classification problem.

According „no free lunch theorem" there are not a single solution which could solve all problems, but classifiers have different domains of competence [29]. It is worth noting that a chosen classifier could make mistakes because of:

- its model does not fit to the real target concept (e.g. model is simplified because of costs),
- learning set is limited,
- learning set is unrepresentative or includes errors.

Fortunately we are not doomed to failure because usually for each classification task we could use many classifiers. We could choose the best one on the basis of evaluation process or use all available classifiers. Let's note that usually an incompetence area, i.e. subset of feature space where all individual classifiers make wrong decision is very small what is shown in Fig. 2.

[1] Department of Systems and Computer Networks, Wroclaw University of Technology, Wyb.Wyspianskiego 27, 50-370 Wroclaw, Poland, e-mail: Michal.Wozniak@pwr.wroc.pl

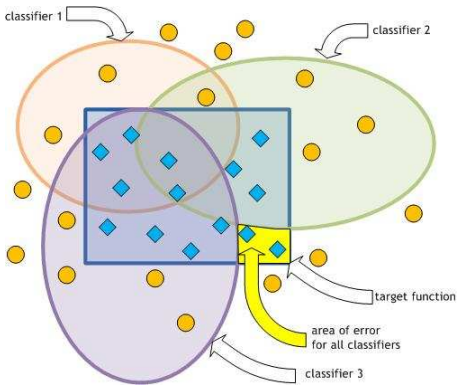


Fig. 2. Decision areas of 3 different classifiers for a toy problem.

In many review articles multiple classifier systems (MCSs) have been mentioned as one of the most promising in the field of pattern recognition [15]. In this conceptual approach, the main effort is concentrated on combining knowledge of the set of elementary classifiers [5, 37]. The main motivations of using MCSs are as follows:

- for small sample MCSs could avoid selection of the worst classifier [21];
- there are many evidences that classifiers combination can improve the performance of the best individual ones and it can exploit unique classifier strengths [8, 10, 28, 31];
- additionally combined classifier could be used in distributed environment, especially in the case that database is partitioned from privacy reason and in each node of computer network only final decision could be available.

Let us observe that designing a MCS is similar to design of a classical pattern recognition [11] application what is shown in Fig.3. Design a typical classifier is aimed to select the best – the most valuable features and choose the best classification method from the set of available ones. Design of the classifier ensemble is similar – it is aimed to create a set of complementary/diverse classifiers. Design of fuser is aimed to create a mechanism that can exploit the complementary/diversity of classifiers form ensemble and combine them optimally.

Fig. 3. Comparison of classical pattern recognition system design and MCS design.

There are a number of important issues while building the MCSs, which could be grouped into the following issues:

- topology of the MCs,
- classifier ensemble design,
- fuser design.

Most of the combining classifiers based on parallel topology, which has good methodological background [20]. In this paper we will focus on two remaining problems.

2. ENSEMBLE DESIGN

One of the most important issue while building MSCs is how to select classifiers in way which make the quality of ensemble better then quality of the best individual one. Combining similar classifiers should not contribute much to the system being constructed, apart from increasing the computational complexity, therefore it is important to select committee members with possibly different components. An ideal ensemble consists of classifiers with high accuracy and high diversity [23], i.e. mutually complementary, what is shown in the left picture of Fig.4.

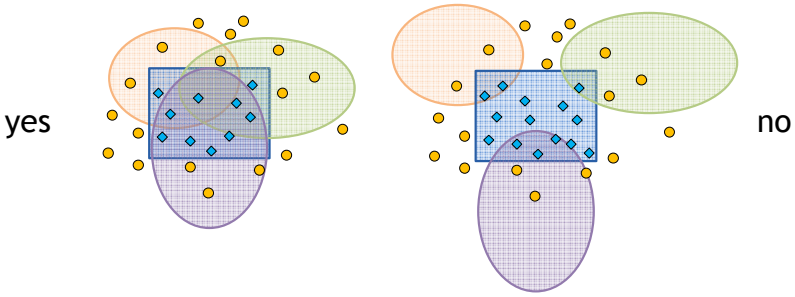


Fig. 4. Pools of classifiers with high diversity but different qualities.

One of current research is trying to answer the question how to measure the diversity. Proposed methods exploit several types of diversity measures which, for example, can be used to minimize the possibility of coincidental failure by different classifiers in the ensemble [18]. A strategy for generating the ensemble members must seek to improve the ensemble's diversity. To enforce classifier diversity we could use varying components of the MCS:

- different input data, e.g. we could use different partitions of data set or generate various data sets by data splitting, cross-validated committee, bagging, boosting [20], because we hope that classifiers trained on different inputs are complementary;
- classifiers with different outputs, i.e. each individual classifier could be trained to recognize subset of predefined classes only (e.g. binary classifier - one class against rest ones strategy) and fusion method should recover the whole set of classes. The well known technique is Error-Correcting Output Codes [6];
- classifiers with the same input and output, but trained on the basis of different models or model's versions.

3. FUSER DESIGN

Another important issue is a choice of a collective decision making method. The first group of methods includes algorithms for classifier fusion at the level of their responses [19, 26]. Initially only majority voting schemes were implemented, but in later works more advanced methods were proposed.

Many known conclusions regarding classification quality of MCSs have been derived analytically, but are typically valid only under strong restrictions, such as particular cases of the majority vote (see [12] and Fig.5 where dependencies between number of independent individual classifiers and quality of committee based on majority voting rule is depicted), or make convenient assumptions, such as the assumption that the classifier committee is formed from independent classifiers. Unfortunately, such assumptions and restrictions are mostly of a theoretical character, and not useful in practice.

For this kind of fusion the Oracle classifier is usually used as reference combination method. Many works consider the quality of the Oracle as the limit of the quality of different fusion methods [30]. The Oracle is an abstract fusion model, where if at least one of the classifiers recognizes an object correctly, then the committee of classifiers points at the correct class too. The Oracle is usually used in comparative experiments to show the limits of classifier committee quality [26].

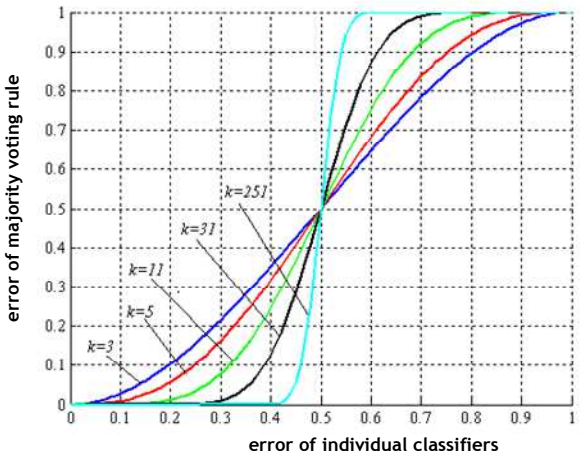


Fig. 5. Dependencies between number of individual classifiers (denoted as k) and quality of ensembles using majority voting rule.

Let us focus on the following problem where we have three independent classifiers of different quality. Let us assume that the first classifier gains the highest classification accuracy $P_{c,1}$ ($P_{c,2}$ denotes accuracy of the second individual classifier and $P_{c,3}$ of the third one respectively). For the purpose of convenience we introduce following notations:

$$\sigma_{12} = P_{c,1} - P_{c,2} \text{ and } \sigma_{13} = P_{c,1} - P_{c,3}, \tag{1}$$

then

$$P_{c,2} = P_{c,1} - \sigma_{12} \text{ and } P_{c,3} = P_{c,1} - \sigma_{13}. \tag{2}$$

Using presented notations we can derive formula for probability of making correct decision that characterize voting committee:

$$\begin{aligned} P(\overline{\Psi}) = {} & P_{c,1}(P_{c,1} - \sigma_{12})(P_{c,1} - \sigma_{13}) + P_{c,1}(P_{c,1} - \sigma_{12})(1 - P_{c,1} + \sigma_{13}) + \\ & + P_{c,1}(1 - P_{c,1} + \sigma_{12})(P_{c,1} - \sigma_{13}) + (1 - P_{c,1})(P_{c,1} - \sigma_{12})(P_{c,1} - \sigma_{13}). \end{aligned} \tag{3}$$

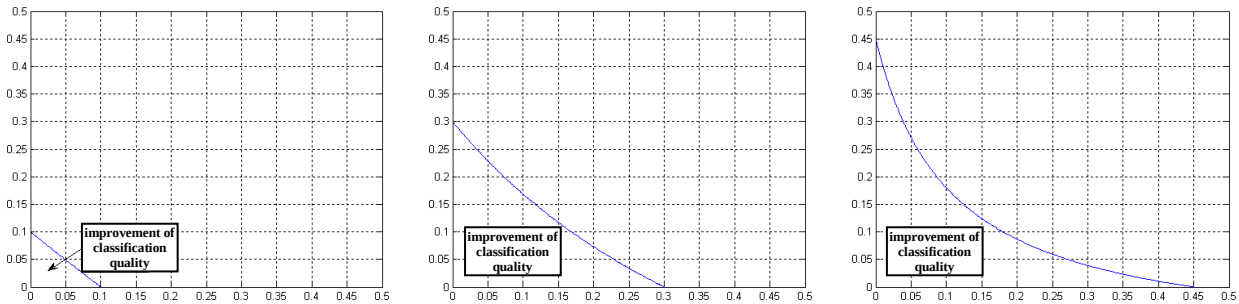
It is clear that making use of the committee consisting of three components with different qualities is advantage only if it outperforms the best of its component, i.e.

$$P_{c,1} < P(\overline{\Psi}), \tag{4}$$

what means:

$$0 \le -2(P_{c,1})^3 + 2(1.5 + \sigma_{12} + \sigma_{13})(P_{c,1})^2 - 2(\sigma_{12} + \sigma_{13} + \sigma_{12}\sigma_{13} - 1)(P_{c,1}) + \sigma_{12}\sigma_{13}. \tag{5}$$

Figure 6. presents graphical interpretation of the relation (5) for selected values of $P_{c,1}$. It means that it is worthy to combine classifiers only if a difference among their quality is relatively small. It has to be noticed that the higher probability of misclassification of the best classifier the smaller quality difference should be in order to get effective committee that outperforms their components. Some additional information about voting classifier quality can be found in [1, 18-23].



$P_{c,1} = 0{,}60$ $P_{c,1} = 0{,}80$ $P_{c,1} = 0{,}95$

Fig. 6. Correct classification probability of MSC vs. difference of correct classification probabilities of its components. Subsequent plots relate to different probability of correct classification of the best classifier in the committee. Area under curved line marks σ_{12} and σ_{13} for which committee outperforms the best of its component.

The formal model of fusion based on classifier responses is as follows. Let us assume that we have n classifiers $\Psi^{(1)}, \Psi^{(2)}, ..., \Psi^{(n)}$ and each of them decides if a given object belongs to class $i \in \boldsymbol{M} = \{1, ..., M\}$. The decision rule of combining classifier $\overline{\Psi}$ is as follows:

$$\overline{\Psi}\left(\Psi^{(1)}, \ ..., \Psi^{(n)}\right) = \arg\max_{j \in \boldsymbol{M}} \sum_{l=1}^{n} \delta\left(j, \Psi^{(l)}\right) w^{(l)} \Psi^{(l)}, \tag{6}$$

where $w^{(l)}$ is the weight assigned to the l-th classifier and

$$\delta(j,i) = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases}. \tag{7}$$

Let us note that weights used in (6) play the key-role in establishing the quality of $\overline{\Psi}$. There is much research dedicated to weight configurations, e.g. in [12, 20] authors proposed to train a fuser. Weights could be dependent on: (1) classifier, (2) classifier and class number, (3 features, classifier, and class. The only model based (partial) on the class label which could achieve better results than the Oracle is a classifier which returns decisions on the basis of class labels given by set of n individual classifiers and feature vector values [14, 20, 24, 25].

Let us consider an alternative model for the construction of a combining classifier, one that performs classifier fusion on the basis of the discriminants of individual classifiers. The main form of discriminants are posterior probability estimators, typically associated with probabilistic models of the pattern recognition task [7], but it could be given for e.g. by the output of neural networks or that of any other function whose values are used to establish the decision of the classifier.

One concept is known as the Borda count. Such classifier based on this concept makes decision by giving each class support corresponding to the position in the ranking.

To see how this method works let us consider the following example:

- 5 class decision problem is given with the following class labels {A,B,C,D,E},
- a pool of 3 classifiers in our disposal,
- a given object belongs to class A.

For a given object the ranked outputs of the individual classifiers is given and presented in Tab.1.

Let us compute ranks for all classes:

rank for class A 4+4+4=12, rank for class B 5+1+5=11, rank for class C 1+5+1=7

rank for class D 2+3+3=8, rank for class E 3+2+2=7

Let us note that Oracle and other voting methods use only the most ranked class and they decide that given object belongs to class B.

Table 1. Ranks for the exemplary problem.

rank			rank
classifier 1	classifier 2	classifier 3	value
B	C	B	5
A	A	A	4
E	D	D	3
D	E	E	2
C	B	C	1

The aggregating methods, which do not require learning perform fusion with the help of simple operators, such as the maximum, minimum, average, or product are typically relevant only in specific, clearly defined conditions [9, 17]. Weighting methods are an alternative and the selection of weights has a similar importance as in the case of weighted voting [3].

The formal model of fusion based on discriminants is as follows. Let us assume that we have each classifier makes a decision on the basis of the values of discriminants. Let $F^{(l)}(i, x)$ denotes a function that is assigned to class i for a given value of x, and which is used by the l-th classifier $\Psi^{(l)}$. A common classifier $\hat{\Psi}(x)$ is described as follows [16]

$$\hat{\Psi}(x)=i \quad if \quad \hat{F}(i, x)=\max_{k\in M}\hat{F}(k, x), \text{ where } \hat{F}(i, x)=\sum_{l=1}^{n} w^{(l)}F^{(l)}(i, x) \text{ and } \sum_{i=1}^{n} w^{(l)}=1. \tag{8}$$

Let us consider the possibilities of weight dependent on: (1) classifier, (2) classifier and feature vector, (3) classifier and class number, (4) classifier, class number, and feature vector. If we consider the two class recognition problem only for the last two cases where weights are dependent on classifier and class number it is possible to produce ensemble which could achieve quality equal or better than Oracle one [34]. But when we take into consideration more the two class problem we could see that it is possible in all aforementioned cases get results better then Oracle one. Weights independent from x could be assigned to linear separated problem, in other cases we should use weights depended on classifier, class number, and feature vector values. More details and illustrative example of the features of weighted voting using weights dependent classifier and class numbers could be found in [32, 33].

Let us focus on the problem of establishing weights dependent on classifier and class number only. For the case where weights are additionally dependent on feature values mentioned weights are functions. Their estimation is more complicated and could required some prior knowledge about them in the form of additional constrains and assumptions. This observation drives us to the problem of function estimation.

For the case where weights dependent on classifier and class number an ensemble learning task leads to the problem how to establish the following vector W [36]

$$W=\left[W^{(1)}, W^{(2)}, ..., W^{(n)}\right] \tag{9}$$

which consists of weights assigned to each classifier and each class number

$$W^{(l)}=\left[w^{(l)}(1), w^{(l)}(2), \ldots, w^{(l)}(M)\right]^{T} \tag{10}$$

We could formulate the following optimization problem. The weights should be established in such way to maximize the accuracy probability of fuser:

$$\Phi(W)=1-P_e(W), \tag{11}$$

where $P_e(W)$ is frequency of misclassification.

In order to solve the aforementioned optimization task, we could use one of a variety of widely used algorithms like genetic algorithms or neural networks [24, 25, 33, 34]. Neural networks can be used to

model complex relationships between inputs and outputs and to solve optimization problems. In experiments we present in the next section we decided to use one layer neural network which model is presented in Fig. 7.

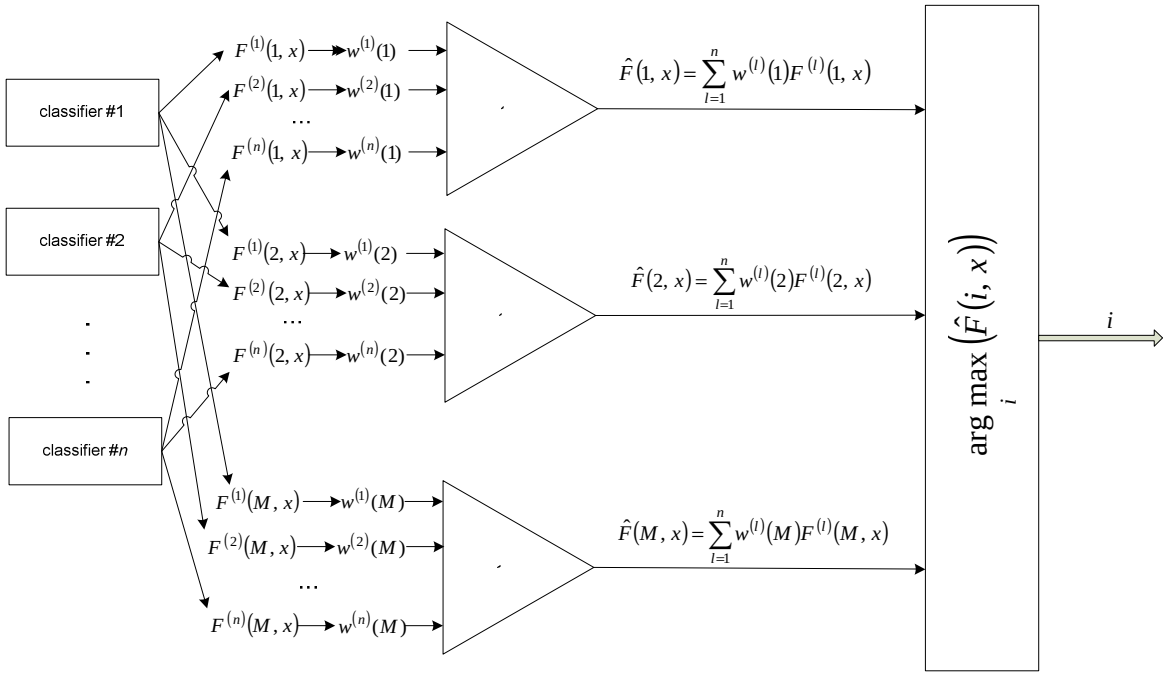


Fig. 7. One layer neural network as a fuser which uses weights depend on classifiers and class numbers.

4. EXPERIMENTAL INVESTIGATION

The aim of the experiments is to evaluate the performance of fuser based on weights dependent on classifier and class number.

The experiment was carried out in Matlab environment using PRTools toolbox [27] and our own software. For the purpose of this experiment, there were five neural networks prepared that could be treated as individual classifiers. They were slightly undertrained (the training process was stopped early for each classifier) to ensure their diversity. The details of used neural nets are as follows:

- Five neurons in the hidden layer,
- sigmoidal transfer function,
- back propagation learning algorithm,
- number of neurons in last layer equals number of classes of given experiment.

Additionally the qualities of the classifiers mentioned above, were compared with the abstract fusion model Oracle classifier, which is usually used in comparative experiments to show the limits of classifier committee quality. To evaluate the experiment we used two databases from UCI Machine Learning Repository [2]:

- *Breast Cancer Wisconsin* dataset (2 classes, 10 attributes, 699 examples) was obtained from the University of Wisconsin Hospitals, Madison and presents 10 cellular characteristic for two classes: malignant and benign.
- *Haberman's Survival* dataset (2 classes, 3 attributes, 306 instances) which contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.

For trained fuser realized according the idea depicted in Fig.7 number of iterations to train was fixed to1500. For each database the experiment was repeated ten times with different epoch of learning for NN. For each obtained fuser its quality was evaluated using 10-fold cross validation method. The best results obtained in those experiments are presented in Fig. 8 with additional information about the result obtained by the Oracle classifier and the majority vote.

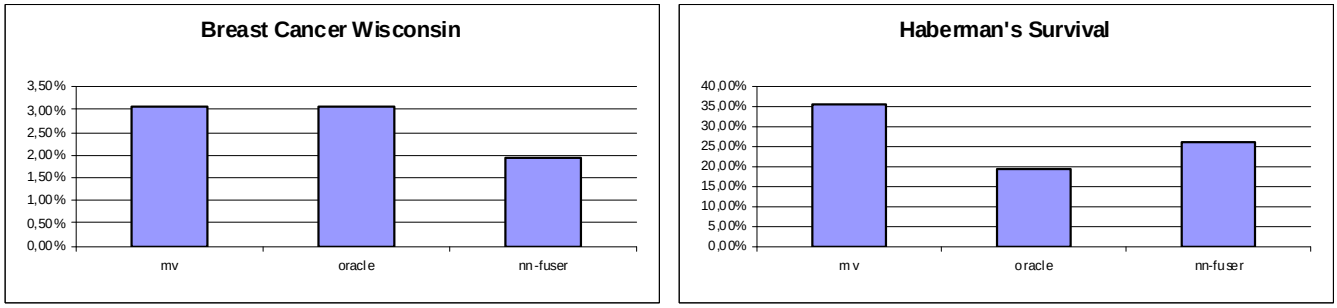


Fig. 8. Classification errors of majority vote (mv), Oracle classifier (oracle) and trained fuser realized as one-layer neural network (nn-fuser) for 2 benchmark datasets.

The results presented in Fig. 5 prove that proposed neural model is very efficient tools for solving optimization problems. As stated before, when weights depend on the classifier and the class number, it is possible to achieve results that are better than the Oracle classifier. We should always remember that the tool that was used in our experiments are somehow black boxes and only appropriate settings of all the parameters can give the best results. Unfortunately, it is not possible to determine values of weights in the analytical way, therefore using heuristic methods of optimization seems to be a promising research direction.

5. CONCLUSION

There is no single ensemble design algorithm or fuser rule that is universally better than others. All of the aforementioned ideas have been shown to be effective on a wide range of real world and benchmark datasets.

Some methods of classifier fusion were presented in this paper. For all of them typical topologies, ensemble and fuser design methods were described. For the last topic the limit of different approaches based on weighted voting was shown. According no free lunch theorem and presented results of experiments, there is no single ensemble design algorithm or fuser rule that is universally better than others.

ACKNOWLEDGEMENT

This work is supported in part by the Polish State Committee for Scientific Research under a grant for the period 2010-2013.

REFERENCES

[1] ALEXANDRE L.A., CAMPILHO A.C., KAMEL M., Combining Independent and Unbiased Classifiers Using Weighted Average, Proc. of the 15th Internat. Conf .on Pattern Recognition, Vol. 2, 2000, pp. 495–498.
[2] ASUNCION A., NEWMAN D.J. , UCI ML Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html], Irvine, CA: University of California, School of Information and Computer Science.
[3] BIGGIO B., FUMERA G., ROLI F., Bayesian Analysis of Linear Combiners, LNCS, Vol. 4472, 2007, pp. 292–301.
[4] BISHOP Ch.M., Pattern Recognition and Machine Learning, Springer, 2006.
[5] CHOW C.K., Statistical independence and threshold functions, IEEE Trans. on Electronic Computers, EC–16, 1965, pp. 66–68.
[6] DIETTERICH T.G., BAKIRI G., Solving multiclass learning problems via error–correcting output codes, Journal of Artificial Intelligence Research, 2, 1995, pp. 263–286.
[7] DUDA R.O., et al., Pattern Classification, Wiley–Interscience, 2001.
[8] DUIN R.P.W., TAX, D.M.J., Experiments with Classifier Combining Rules, LNCS, No. 1857, 2000, pp. 16–29.
[9] DUIN R. P.W., The Combining Classifier: to Train or Not to Train?, Proc. of the ICPR2002, Quebec City, 2002.

[10] FUMERA G., ROLI F., A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems, IEEE Trans.on PAMI, 27(6), 2005, pp. 942–956.
[11] GIACINTO G. , Design Multiple Classifier Systems, PhD thesis, Universita Degli Studi di Salerno, 1998.
[12] HANSEN L.K., SALAMON P. , Neural Networks Ensembles, IEEE Trans. on PAMI, Vol. 12, No. 10, 1990, pp. 993–1001.
[13] HASHEM S., Optimal linear combinations of neural networks, Neural Networks, 10(4), 1997, pp. 599–614.
[14] INOUE H., NARIHISA H., Optimizing a Multiple Classifier Systems, LNCS, Vol. 2417, 2002, pp. 285–294.
[15] JAIN A.K., DUIN P.W., MAO J., Statistical Pattern Recognition: A Review, IEEE Trans. on PAMI, vol 22., No. 1, 2000, pp. 4–37.
[16] JACKOBS R.A., Methods for combining experts' probability assessment, Neural Computation, No. 7, 1995, pp. 867–888.
[17] KITTLER J., ALKOOT F.M., Sum versus Vote Fusion in Multiple Classifier Systems, IEEE Trans. on Pattern Analysis and Machine Intelligence, 20, 2003, pp. 226–239.
[18] KRZANOWSKI W., PARTRIGE D., Software Diversity: Practical Statistics for its Measurement and Exploitation, Department of Computer Science, University of Exeter, 1996.
[19] KUNCHEVA L.I., WHITAKER C.J., SHIPP C.A., DUIN R.P.W., Limits on the Majority Vote Accuracy in Classier Fusion, Pattern Analysis and Applications, 6, 2003, pp. 22–31.
[20] KUNCHEVA L.I., Combining pattern classifiers: Methods and algorithms, Wiley, 2004.
[21] MARCIALIS G.L., ROLI F., Fusion of Face Recognition Algorithms for Video–Based Surveillance Systems, in Foresti G.L., Regazzoni C., Varshney P (Eds.), Multisensor Surveillance Systems: The Fusion Perspective, Kluwer Academic Pub., 2003.
[22] POLIKAR R., Ensemble based systems in decision making, IEEE Circuits and Systems Magazine, 3rd quarter, 2006, pp. 21–45.
[23] RAO N.S.V., A Generic Sensor Fusion Problem: Classification and Function Estimation, LNCS, Vol. 3077, 2006, pp. 16–30.
[24] RAUDYS S., Trainable fusion rules. I. Large sample size case, Neural Networks 19, 2006, pp. 1506–1516.
[25] RAUDYS S., Trainable fusion rules. II. Small sample–size effects, Neural Networks 19, 2006, pp. 1517–1527.
[26] TUMER K., GHOSH J., Analysis of Decision Boundaries in Linearly Combined Neural Classifiers, Pattern Recognition, 29, 1996, pp. 341–348.
[27] VAN DER HEIJDEN F., DUIN, R.P.W., de RIDDER, D.,TAX,D.M.J. , Classification, parameter estimation and state estimation – an engineering approach using Matlab, John Wiley and Sons, 2004.
[28] VAN ERP M., VUURPIJL L.G., SCHOMAKER L.R.B. , An overview and comparison of voting methods for pattern recognition, Proc. of IWFHR.8, Canada, 2002, pp. 195–200.
[29] WOLPERT D.H., The supervised learning no–free–lunch theorems. In: Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications, 2001.
[30] WOODS K., KEGELMEYER W.P., Combination of multiple classifiers using local accuracy estimates, IEEE Transactions on PAMI, Vol. 19, Issue 4, 1997, pp. 405–410.
[31] WOZNIAK M., Experiments with trained and untrained fusers [in:] Corchado E. et al. (eds) Innovations in hybrid intelligent systems, Springer series "Advances in Soft Computing", Berlin, 2007, pp. 144–150.
[32] WOZNIAK M., JACKOWSKI K., Some remarks on chosen methods of classifier fusion based on weighted voting, LNCS Vol. 5572, 2009, pp. 541–548.
[33] WOZNIAK M., Evolutionary approach to produce classifier ensemble based on weighted voting, Prodeedings of World Congress on Nature & Biologically Inspired Computing, NaBIC' 2009 , 9–11 December 2009, Coimbatore, India, 2009, pp. 648–653.
[34] WOZNIAK M., ZMYSLONY M., Fuser on the basis of discriminants evolutionary and neural methods of training, LNCS Vol. 6077, Springer, 2010, pp. 590–597.
[35] WOZNIAK M., ZMYSLONY M. , Method of designing classifier fuser – evolutionary approach, Proceedings of the 44th Spring International Conference Modelling and simulation of systems, Ostrava, 2010, pp. 88–92.
[36] XU L., KRZYZAK A., SUEN Ch.Y., Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, IEEE Trans. on SMC, No. 3, 1992, pp. 418–435.