Multiple Classifier Fusion using *k*-Nearest Localized Templates

Jun-Ki Min and Sung-Bae Cho

Department of Computer Science, Yonsei University
Biometrics Engineering Research Center
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Korea
loomlike@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. This paper presents a method for combining classifiers that uses *k*-nearest localized templates. The localized templates are estimated from a training set using *C*-means clustering algorithm, and matched to the decision profile of a new incoming sample by a similarity measure. The sample is assigned to the class which is most frequently represented among the *k* most similar templates. The appropriate value of *k* is determined according to the characteristics of the given data set. Experimental results on real and artificial data sets show that the proposed method performs better than the conventional fusion methods.

Keywords: Classifier fusion; Decision templates; *C*-means clustering

1 Introduction

Combining multiple classifiers has been actively exploited for developing highly reliable pattern recognition systems in the past decade [1, 2]. There are two basic parts for generating an ensemble: creating base classifiers and combining the outputs of the classifiers. In order to achieve the higher accuracy of the ensemble, the individual classifiers have to be both diverse and accurate [3, 4]. Two popular methods for creating classifiers are Bagging and Boosting [5]. Bagging creates each individual classifier in the ensemble with a different random sampling of the training set. Thus some instances are represented multiple times while others are left out. In Boosting, examples that were incorrectly predicted by previous classifiers in the ensemble are chosen more often than examples that were correctly predicted.

The outputs of the diverse classifiers have to be combined with some manner to achieve a group consensus. In order to improve further on the performance of the ensemble, several existing and novel combining strategies have been investigated [6, 7]. Some combiners do not require additional training after the classifiers in the ensemble have been trained individually. Majority voting, minimum, maximum, and average are examples of them [8, 9, 10]. Other combiners need training at fusion level. Examples are behavior knowledge space (BKS) [11] and decision templates (DT) [12]. Especially, DT that composes a template for each class by averaging the outputs of classifiers was reported good performance and was used complementarily with a classifier selection method [13]. However, because the DT abstracts the characteristics of a class into a template, there might be the limitation of applying it to complex problems. In our previous work [14], multiple decision templates (MuDTs) which decompose a template into several localized templates using clustering algorithm was investigated to solve this limitation. Since many clustering algorithms rely on a random component, this method would be sensitive to clustering results.

In this paper, we present a novel fusion method, *k*-nearest localized template (*k*-NLT), which refers *k* most similar templates among the multiple decision templates. It may be less affected by clustering results and thus can obtain stable and high accuracy. Finally, to validate the proposed method, its performance are compared with several classifier combining approaches by using real and artificial data sets from the UCI database and ELENA.

2 Background

2.1 Conventional Fusion Methods

Simple fusion methods such as majority voting, minimum, maximum, average, and BKS have been widely used to construct a multiple classifier system.

Majority Voting. For a sample, this method simply counts the votes received from the individual classifiers, and selects the class with the largest number of votes. Ties are broken randomly.

Minimum, Maximum, and Average. These three fusion methods are considered together because they have a similar decision scheme. The minimum method selects the smallest value among the outputs of the classifiers for each class. The minimums are then compared and a class with the larger value is selected. For an *M*-class problem with *L* classifiers, it is calculated as follows:

$$\max_{z=1,\dots,M}\left\{\min_{y=1,\dots,L}\{d_{y,z}(x)\}\right\}. \qquad (\mathbf{1})$$

Here, $d_{y,z}(x_i)$ is the degree of support given by the *y*th classifier for the sample *x* of the class *z*. The maximum and the average methods are the same as the minimum method except that the biggest values are compared as

$$\max_{z=1,\dots,M}\left\{\max_{y=1,\dots,L}\{d_{y,z}(x)\}\right\} \qquad (\mathbf{2})$$

for the maximum method, and the average method compares the mean values as

$$\max_{z=1,\dots,M}\left\{\operatorname{avg}_{y}\{d_{y,z}(x)\}\right\},\ \operatorname{avg}_{y}\{d_{y,z}(x)\}=\frac{1}{L}\sum_{y=1}^{L}d_{y,z}(x). \qquad (\mathbf{3})$$

Behavior Knowledge Space. In this method, possible combinations of the outputs of the classifiers are stored in the BKS-table $T \in \{-1, 1\}^{M^L \times L}$. Each entry in the T contains a class label (most frequently encountered amongst the samples of the training data in this cell) or no label (no sample of the training data has the respective combination of class labels). In tests, a new sample can be classified into the label of the entry with the same outputs of the classifiers. It fails to classify when an output pattern is not found in *T*.

2.2 *C*-Means Algorithm

The *C*-means (or *K*-means) algorithm is an iterative clustering method that finds *C* compact partitions in the data using a distance-based technique [15]. The cluster centers are initialized to *C* randomly chosen points from the data, which is then partitioned based on the minimum squared distance criterion

$$I = \sum_{i=1}^{n} \sum_{c=1}^{C} u_{c,i} \left\| x_i - z_c \right\|^2 . \qquad \textbf{(4)}$$

Here, n is the total number of samples in the data set, z_c is the center of the cth cluster, and $u_{c,i}$ is the membership of the ith sample x_i in cluster c. The cluster centers are subsequently updated by calculating the average of the samples in each cluster and this process is repeated until cluster centers no longer change. Although this algorithm tends to find the local minima, it is widely used for clustering because of its simplicity and fast convergence.

2.3 Decision Templates

DT proposed by Kuncheva [12] estimates *M* templates (one per class) with the same training set that is used for the set of classifiers. For the *M*-class problem, the classifier outputs can be organized in a decision profile as a matrix

$$DP(x_i) = \begin{bmatrix} d_{1,1}(x_i) & \cdots & d_{1,M}(x_i) \\ \vdots & d_{y,z}(x_i) & \vdots \\ d_{L,1}(x_i) & \cdots & d_{L,M}(x_i) \end{bmatrix}, \qquad \textbf{(5)}$$

where L is the number of classifiers in an ensemble and $d_{y,z}(x_i)$ is the degree of support given by the yth classifier for the sample x_i of the class z. When decision profiles are generated, the template of the class m is estimated as follows:

$$DT_m = \begin{bmatrix} dt_m(1,1) & \cdots & dt_m(1,M) \\ \vdots & dt_m(y,z) & \vdots \\ dt_m(L,1) & \cdots & dt_m(L,M) \end{bmatrix}, \quad dt_m(y,z) = \sum_{i=1}^{n} u_{m,i} d_{y,z}(x_i) \Big/ \sum_{i=1}^{n} u_{m,i} \qquad \textbf{(6)}$$

In the test stage, the similarity between the decision profile of a test sample and each template is calculated. The sample is then categorized into the class of the most similar template. Kuncheva [16] examined DT with various distance measures, and achieved higher classification accuracies than conventional fusion methods.

3 *k*-Nearest Localized Templates

The DT scheme abstracts features of each class as a template which may be difficult to classify dynamic patterns. For dealing with the intra-class variability and the inter-class similarity of the dynamic patterns, we adopt a multiple template-based approach where patterns in the same class are characterized by a set of localized classification models. Fig. 1 illustrates an overview of the proposed method.

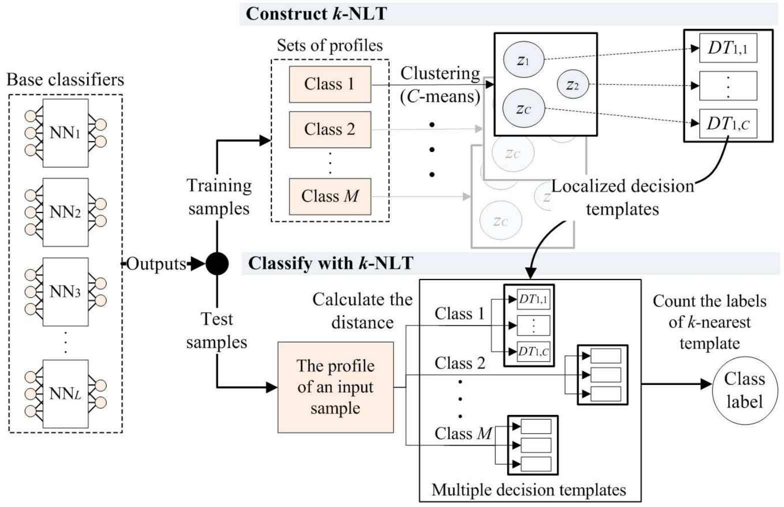

Fig. 1. An overview of the *k*-nearest localized templates

3.1 Estimation of Localized Decision Templates

Localized decision templates are estimated in order to organize the multiple classification models. At first, decision profiles are constructed from the outputs of the base classifiers as Eq. (5) and are clustered for each class using *C*-means algorithm. The localized template of the *c*th cluster in the class *m*, $DT_{m,c}$, is then estimated as follows:

$$DT_{m,c} = \begin{bmatrix} dt_{m,c}(1,1) & \cdots & dt_{m,c}(1,M) \\ \vdots & dt_{m,c}(y,z) & \vdots \\ dt_{m,c}(L,1) & \cdots & dt_{m,c}(L,M) \end{bmatrix}, \quad dt_{m,c}(y,z) = \frac{\sum_{i=1}^{n} u_{m,c,i} d_{y,z}(x_i)}{\sum_{i=1}^{n} u_{m,c,i}} \tag{7}$$

Here, $u_{m,c,i}$ is the membership of the *i*th sample x_i in the cluster *c* of the *m*th class. Finally, M×C templates are constructed where *M* is the number of classes and *C* is the number of clusters per class. In this paper the number of clusters was selected as 20 based on the experiments in section 4.1

3.2 Classification Using *k*-Nearest Localized Templates

In the test stage, the profile of a new input sample is matched to the localized templates by a similarity measure. A distance between the profile of a given sample *x* and the template of each cluster is calculated as follows:

$$dst_{m,c}(x) = \left\| DT_{m,c} - DP(x) \right\| . \tag{8}$$

Since the *C*-means clustering algorithm which was used for generating localized templates is often affected by its random initial instances, it is easy to make error clusters. The error clusters cause a misclassification when the sample is only matched to the nearest template. In order to resolve this problem, the proposed method adopts a *k*-nearest neighbor scheme where the sample is assigned to the class that is most frequently represented among the *k* most similar templates. In this approach, the appropriate value of *k* commonly depends on the properties of a given data set. The proposed method, therefore, analyzes the intra-class compactness *IC* and the inter-class separation *IS* (which were originally designed for the validity index of clustering algorithm [17]) of the data set using:

$$IC = E_1 / E_M \ , \ E_M = \sum_{i=1}^{n} \sum_{m=1}^{M} u_{m,i} \left\| x_i - z_m \right\| \tag{9}$$

$$IS = \max_{i,j=1,\ldots,c} \left\| z_i - z_j \right\| \tag{10}$$

where *n* is the total number of points in the data set, z_m is the center of the *m*th class, and $u_{m,i}$ is the membership of the *i*th sample x_i in class *m*. In this paper we generate a simple rule for *k* as Eq. (11) based on experiments (see section 4.1).

$$k = \begin{cases} 1 & \text{if} \quad IC \le t_{IC} \ \text{ and } \ IS \le t_{IS} \\ C/2 & \text{if} \quad IC > t_{IC} \ \text{ and } \ IS > t_{IS} \end{cases} \tag{11}$$

4 Experiments

In this paper, we have verified the proposed method on 10 real (R) and artificial (A) data sets from the UCI database and ELENA which are summarized in Table 1. Each feature of data sets was normalized to a real value between -1.0 and 1.0. For each data set 10-fold cross validation was performed. The neural network (NN) was used as a base classifier of an ensemble. We trained the NN using standard backpropagation learning. Parameter settings for the NN included a learning rate of 0.15, a momentum term of 0.9, and weights were initialized randomly between -0.5 and 0.5. The number of hidden nodes and epochs were chosen based on the criteria given by Opitz [5] as follows: at least one hidden node per output, at least one hidden node for every ten inputs, and five hidden nodes being a minimum; 60 to 80 epochs for small problems involving fewer than 250 samples, 40 epochs for the mid-sized problems containing between 250 to 500 samples, and 20 to 40 epochs for larger problems (see Table 1).

Table 1. Summary of the data sets used in this paper

						Neural network	
Type	Data set	Case	Feature	Class	Availability	Hidden	Epoch
R	Breast-cancer	683	9	2	UCI[1]	5	20
R	Ionosphere	351	34	2	UCI	10	40
R	Iris	150	4	3	UCI	5	80
R	Satellite	6435	36	6	UCI	15	30
R	Segmentation	2310	19	7	UCI	15	20
R	Sonar	208	60	2	UCI	10	60
R	Phoneme	5404	5	2	ELENA[2]	5	30
R	Texture	5500	40	11	ELENA	20	40
A	Clouds	5000	2	2	ELENA	5	20
A	Concentric	2500	2	2	ELENA	5	20

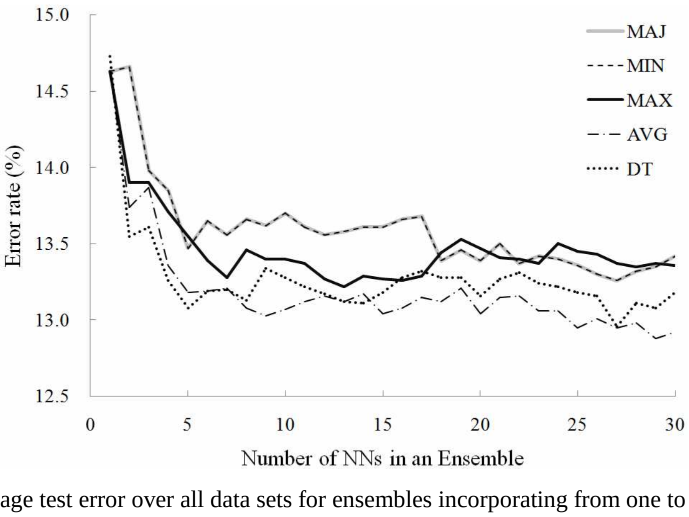

Fig. 2. Average test error over all data sets for ensembles incorporating from one to 30 neural networks

[1] http://mlearn.ics.uci.edu/MLRepository.html
[2] http://www.dice.ucl.ac.be/mlg/?page=Elena

In order to select the appropriate size of an ensemble, preliminary experiments with conventional fusion methods: majority voting (MAJ), minimum (MIN), maximum (MAX), average (AVG), and DT were performed using up to 30 NNs. As shown in Fig. 2, there is no significant error reduction over 25 classifiers. Therefore, ensemble size of 25 was chosen for the remaining experiments.

4.1 Parameter Setting of the *k*-Nearest Localized Templates

Two major parameters of the proposed method, *C* (the number of clusters per class) and *k* (the number of referring templates), were selected based on the characteristics of given data. The data sets used in our studies were partitioned into two groups according to *IC* and *IS* as depicted in Fig. 3. One group had small values of *IC* and *IS* (Ionosphere, Sonar, Phoneme, Clouds, and Concentric), while the other group had large values of *IC* and *IS* (Satellite, Texture, Segmentation, Breast-cancer, and Iris). In this paper, we chose Ionosphere and Satellite as the representative data sets of the two groups, and performed two series of experiments on them to select *C* and generate the rules for *k* (Eq. 11).

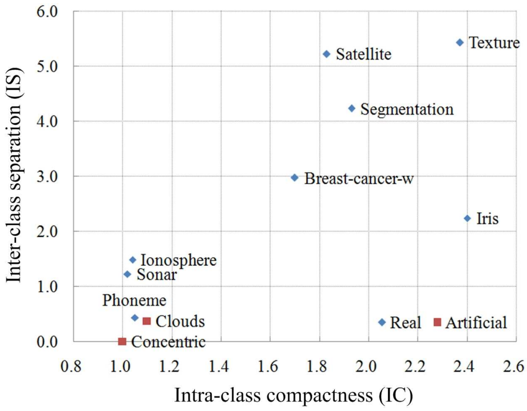

Fig. 3. Characteristics of the data sets used in this paper. *IC* and *IS* are estimated as Eq. (9) and Eq. (10), respectively.

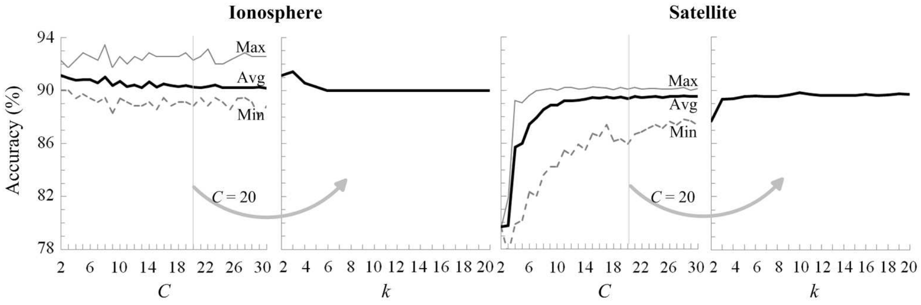

Fig. 4. Accuracies for the two data sets according to *C* (where *k* = 1~*C*) and *k* (where *C* = 20)

First, we investigated the value of *C* where it had changed from one to 30 while *k* had changed from one to *C*. Since the accuracies were converged after 20 values of *C*, we fixed *C* as 20 and changed *k* from one to 20 in the second series of experiments. As shown in Fig. 4, accuracy was decreased when *k* was increasing for the Ionosphere. In case of Satellite, on the other hand, accuracy was increased when *k* was increasing. Therefore, for the remaining experiments, we simply selected *k* based on Eq. (11) where $t_{IC} = 1.5$, $t_{IS} = 2.0$, and $C = 20$.

4.2 Classification Results

We performed the comparison experiments with *k*-NLT against the conventional fusion methods. Table 2 provides the accuracies of 10-fold cross validation experiments for all data sets except Ionosphere and Satellite used for the parameter selection of the *k*-NLT. SB indicates the single best classifier among 25 NNs used in the ensemble. MuDTs, which combine the outputs of the classifiers using localized templates like *k*-NLT, only refer the class label of the nearest template. Oracle (ORA) was used as a comparative method which is assign the correct class label to an input sample if at least one individual classifier produces the correct class label of the sample. As shown in Table 2, the localized template-based methods (MuDTs and *k*-NLT) achieved a high classification performance for the overall data sets. Especially, *k*-NLT showed the best accuracies on more than half of the data sets.

Table 2. Average test accuracy (%) for each data set. Marked in boldface are the best accuracies in each column.

Dataset	Breast-cancer	Iris	Segmentation	Sonar	Phoneme	Texture	Clouds	Concentric
SB	97.5 ±1.8	97.3 ±4.7	94.2 ±1.9	85.5 ±6.4	80.4 ±2.0	99.6 ±0.2	79.9 ±3.6	96.2 ±2.7
MAJ	96.9 ±1.6	96.7 ±4.7	94.1 ±1.9	85.5 ±6.4	80.2 ±1.5	**99.7** ±0.2	79.5 ±2.5	97.7 ±1.2
MIN	97.1 ±1.6	96.7 ±4.7	93.6 ±2.2	81.0 ±8.4	80.3 ±1.6	99.6 ±0.2	79.3 ±2.5	97.6 ±1.2
MAX	97.1 ±1.7	96.0 ±4.7	94.4 ±1.9	82.5 ±9.2	80.3 ±1.6	99.6 ±0.3	79.3 ±2.5	97.6 ±1.2
AVG	97.1 ±1.8	**97.3** ±4.7	94.5 ±1.7	**86.0** ±6.6	80.3 ±1.4	**99.7** ±0.2	79.4 ±2.5	97.8 ±0.8
BKS	95.9 ±2.1	93.3 ±8.3	87.7 ±2.8	72.5 ±14.	79.8 ±1.6	97.8 ±0.7	78.6 ±2.4	92.6 ±2.5
DT	**97.2** ±1.8	**97.3** ±4.7	94.5 ±1.7	85.5 ±6.4	80.4 ±1.5	**99.7** ±0.2	79.6 ±2.5	98.0 ±0.8
MuDTs	95.4 ±2.1	95.3 ±5.5	**96.2** ±1.4	84.0 ±7.8	**80.7** ±1.8	99.6 ±0.2	**81.9** ±1.7	**98.8** ±0.6
k-NLT	**97.2** ±1.8	96.7 ±4.7	94.6 ±1.5	84.0 ±7.8	**80.7** ±1.8	**99.7** ±0.2	**81.9** ±1.7	**98.8** ±0.7
ORA	98.7 ±1.8	98.7 ±2.8	98.8 ±0.5	98.0 ±3.5	93.1 ±1.2	99.9 ±0.1	84.7 ±3.7	100 ±0.0

Fig. 5 shows the average test errors and averaged standard deviations over all data sets. The standard deviation can be interpreted as the stability measure of algorithm. BKS showed the worst performance while *k*-NLT yielded the highest accuracy with stable performance among the compared methods. The paired t-tests between *k*-NLT and comparable methods, AVG, DT, and MuDTs which produced relatively high accuracies, were conducted and revealed that the differences were statistically significant ($p<0.02$, $p<0.002$, and $p<0.007$ respectively).

In order to compare the relative performance of each method with respect to the others, we calculate a rank score. For each data set, each method was assigned a rank with respect to its place among the others. The highest possible score which assigned to the best model was nine, and the lowest was one. The ranks for each method were then summed to give a measure of the overall dominance among the methods. As shown in Fig. 6, *k*-NLT achieved highest score among the others.

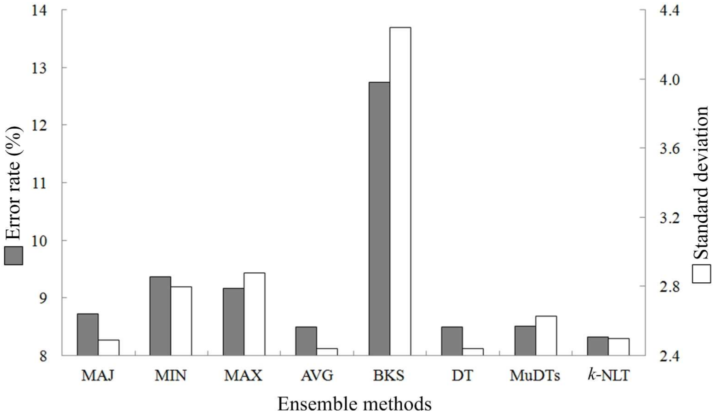

Fig. 5. Average test errors and standard deviations over all data sets used in this paper

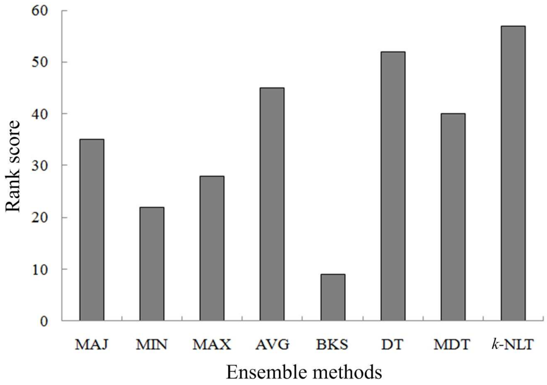

Fig. 6. The sum of rank scores for all data sets. The higher the score, the better the fusion method.

5 Conclusions

In this paper, we proposed *k*-nearest localized templates (*k*-NLT) for combining multiple classifiers. First, decision profiles (the outputs of classifiers) of training set were clustered for each class. Second, localized templates were estimated by averaging decision profiles of each cluster. The templates were then matched to the decision profile of a test sample by a similarity measure. Finally, the sample was assigned to the class which was most frequently represented among the *k* most similar templates. Here, the appropriate value of *k* was selected according to the intra-class compactness and the inter-class separation of a given data set. Experimental results on ten real and artificial data sets showed that the proposed method performed better than conventional fusion methods. The advantage of *k*-NLT can be proved with theoretical backgrounds about the subclass-based model and the *k*-nearest neighbor approach. In near future, further experiments on additional data sets shall be conducted to analyze the parameters of *k*-NLT.

Acknowledgments. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University.

References

1. Cho, S.-B., Kim, J.-H.: Multiple Network Fusion using Fuzzy Logic. IEEE Trans. Neural Networks. 6 (2), 497--501 (1995)
2. Cho, S.-B.: Pattern Recognition with Neural Networks Combined by Genetic Algorithm. Fuzzy Sets and Systems. 103 (2), 339--347 (1999)
3. Hansen, L., Salamon, P.: Neural Network Ensembles. IEEE Trans. Pattern Analysis and Machine Intelligence. 12, 993--1001 (1990)
4. Wanas, N.M., Dara, R.A., Kamel, M.S.: Adaptive Fusion and Co-operative Training for Classifier Ensembles. Pattern Recognition. 39 (9), 1781--1794 (2006)
5. Opitz, D., Maclin, R.: Popular Ensemble Methods: An Empirical Study. J. Artificial Intelligence Research. 11, 169--198 (1999)
6. Alkoot, F.M., Kittler, J.: Experimental Evaluation of Expert Fusion Strategies. Pattern Recognition Letters. 20, 1361--1369 (1999)
7. Kuncheva, L.I.: A Theoretical Study on Six Classifier Fusion Strategies. IEEE Trans. Pattern Analysis and Machine Intelligence. 24 (2), 281--286 (2002)
8. Lam, L. Suen, C.Y.: Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance. IEEE Trans. Systems, Man, and Cybernetics. 27 (5), 553--568 (1997)
9. Kittler, J.: Combining Classifiers: A Theoretical Framework. Pattern Analysis and Applications. 1 (1), 18--27 (1998)
10. Kittler, J., Alkoot, F.M.: Sum versus Vote Fusion in Multiple Classifier Systems. IEEE Trans. Pattern Analysis and Machine Intelligence. 25 (1), 110--115 (2003)
11. Huang, Y.S., Suen, C.Y.: A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals. IEEE Trans. Pattern Analysis and Machine Intelligence. 17 (1), 90--94 (1995)
12. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision Templates for Multiple Classifier Fusion: An Experimental Comparison. Pattern Recognition. 34 (2), 299--314 (2001)
13. Kuncheva, L.I.: Switching between Selection and Fusion in Combining Classifiers: An Experiment. IEEE Trans. Systems, Man, and Cybernetics. 32 (2), 146--156 (2002)
14. Min, J.-K., Hong, J.-H., Cho, S.-B.: Effective Fingerprint Classification by Localized Models of Support Vector Machines. In: Zhang, D., Jain, A.K. (eds.) ICB 2006. LNCS, vol. 3832, pp. 287--293. Springer, Heidelberg (2006)
15. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
16. Kuncheva, L.I.: Using Measures of Similarity and Inclusion for Multiple Classifier Fusion by Decision Templates.: Fuzzy Sets and Systems. 122 (3), 401--407 (2001)
17. Maulik, U., Bandyopadhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. IEEE Trans. Pattern Analysis and Machine Intelligence. 24 (12), 1650--1654 (2002)