VILNIUS UNIVERSITY

Rimantas Kybartas

MULTI-CLASS RECOGNITION USING PAIR-WISE CLASSIFIERS

Doctoral dissertation

Physical sciences, informatics (09 P)

Vilnius, 2010

Dissertation work was carried out at Vilnius University from 2005 to 2010.

Scientific supervisor:

prof. habil. dr. Šarūnas Raudys (Vilnius University, physical sciences, informatics – 09 P)

VILNIAUS UNIVERSITETAS

Rimantas Kybartas

DAUGELIO KLASIŲ ATPAŽINIMAS NAUDOJANT KLASIFIKATORIUS POROMS

Daktaro disertacija

Fiziniai mokslai, informatika (09 P)

Vilnius, 2010

Disertacija rengta 2005 – 2010 metais Vilniaus universitete.

Mokslinis vadovas:

prof. habil. dr. Šarūnas Raudys (Vilniaus universitetas, fiziniai mokslai, informatika – 09 P)

Acknowledgements

Firstly I would like to thank people who lead me through the astonishing path of all my studies. Thanks go to my teacher of mathematics at school, Mrs. Jūratė Raščiauskienė, for trusting in me when I did not. To dr. Ričardas Kudžma, my teacher at university, who made me trust in myself. To dr. Antanas Mitašiūnas who supported that trust. And especially I would like to thank my supervisor prof. habil. dr. Šarūnas Raudys who did not trust in me when I trusted in myself too much.

My heartiest thanks go to my family (especially to my wife Neringa) who sacrificed a lot for my PhD studies.

I also thank my colleagues Virginija Daukutė and Viktoras Golubevas from the Bank of Lithuania for their understanding, support and help.

Thanks to all others who supported and helped me during my PhD journey.

Table of Contents

Notation

C – regularization parameter fr support vector classifiers

$C_{i,j}$ – classifier classifying classes Π_i and Π_j.

d – dimension of the data.

K – the number of classes.

m – mean vector

N – the size of whole data set.

N_e – the number of experiments.

Ni – the size of class Π_i.

N_{Tr} – the amount of vectors in the training data set.

N_{Ts} – the amount of vectors in the testing data set.

p_i – a posteriori probability of the i^{th} class

P – classification error.

q_i – a priori probability of the i^{th} class

S – estimate of the sample covariance matrix

v – weight vector of classifier

w_0 – bias weight of classifier

Π_i – data set of class i

ρ - correlation coefficient

Σ - covariance matrix

Abbreviations

A-B – Anderson - Bahadur (linear discriminant function)

ANN – artificial neural network

BKS – behavior knowledge space method

CM – covariance matrix

DAG – classifier fusion algorithm using direct acyclic graph

DF – discriminant function

EDC – Euclidean distance classifier

GCCM – data model containing data with Gaussian distribution and sharing the common covariance matrixes

H-T – Hastie-Tibshirani pair-wise classifier fusion method

k-NN - *k* nearest neighbors noise adding algorithm

K-SLP – single layer perceptron classifying *K* classes

LDA – linear discriminant analysis

PR – pattern recognition

PWFT – pair-wise fuzzy templates method

QDA – quadratic discriminant analysis

RBF – radial basis function

RDA – regularized discriminant analyis

SLP – single layer perceptron

SVM – support vector machine

WLW – pair-wise classifier fusion method presented by Wu, Lin and Weng

xD – x-dimensional, x is a natural number.

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Research Area

In the real world people face multi-class classification tasks every day, e.g. assigning characters to letters and corresponding sounds while reading, organizing one's daily tasks by their importance, and so on. Moreover, there are plenty of multi-class classification tasks in industry, e.g. mineral classification according to their appearance, plant classification to species according to their look, shape, fruit taste etc. Many of such industry tasks are performed by people, thus making such work rather expensive, error-prone, limited by time or even dangerous. In order to overcome such shortcomings and to make industrial multi-class classification tasks more automated statistical methods of classification were employed. Many complex both - statistical and empirical methods for solving such tasks were developed.

1.2 Problem Relevance

While solving multi-class classification tasks in industrial, social or any other area, standard statistical methods such as Fisher discriminant function or quadratic discriminant function could be used. Unfortunately, such methods only work well enough when data has proper statistical characteristics. In real world tasks, the data is usually rather complex, the relations between attributes are nonlinear, data classes overlap etc. More advanced statistical methods (e.g. employing kernel functions) could be used in such cases. There is also plenty of other multi-class classification techniques based not only on statistics, e.g. heuristic methods such as nearest mean or nearest neighbors approach.

Nowadays (especially in practical tasks of some particular domain) artificial neural networks (ANN) are often used. The standard K-category network of single layer perceptrons (SLPs) is composed of one layer containing K SLPs [1]-[3]. Such multi-category scheme does not successfully work in all the cases. Performance (in the context of time) and generalization issues become more severe while using more complex ANNs.

A powerful stream of research aimed at improving multi-category classification has been done. One of successful solutions for such improvement is based on employing pair-wise classifiers in two-stage decision making. In the first stage pair-wise classifiers are used and their fusion is done in the second stage in order to assign a class label to the data to be classified. Most often the support vector or classifiers based on decision trees are used at the first stage [5] – [6]. Special fusion rules are developed to process the $K(K\text{-}1)/2$ first stage pair-wise outputs and to make the final decision [9]-[12]. The Kullback-Leibler distance [6], diverse variants of the sum of squares of pair-wise conditional probabilities [7], [13] and other approximate expressions of the classification error-rate are used to diminish the classification error-rate in the K-category problem.

In the majority of investigations, the differences between the fusion methods applied were minor. Due to the complexity of methods or nonlinearity or their decision functions, it is usually difficult to estimate their performance. Thus many investigations compare the accuracy of diverse methods by means of simulation. Very often only a small number of experiments is performed. The researchers usually pay most attention to the number of benchmark data sets, however, they often forget about the reliability of their experimental evaluations. If the data size is not large, random selections of the data for multi-class classifier constructing and testing may give diverse results. For this reason, the error rate estimates become unreliable.

Another modification of the popular K-category SLPs consists of K separate classifiers. Each of them is trained to discriminate between one class

and the rest (i.e. one-against-all approach). But it was found that the two-stage pair-wise decision making is more promising [5].

Because of the great variety of pair-wise classifiers and many fusion algorithms it is not clear which kind of pair-wise classifiers and what kind of fusion one has to use in order to get optimal multi-class classification results for a particular task.

Elucidation of existing theories is much more valuable than proposal of new ones without proper theoretical explanation. The two-stage decision making based on pair-wise classifiers is one of such areas. The two-stage algorithms are promising since the researchers may adapt a number of features and complexity of the pair-wise classifiers to the training set size. In addition, it becomes easier to solve problems with imbalanced training sets. Therefore, there is demand for theoretical clarification of reasons why the two-stage decision making schemes are promising.

1.3 Research Object

The research object is multi-class classification using two-stage classification methods where specialized pair-wise classifiers are used at and their outputs are fused and the final decision is made at the second stage.

1.4 The Objectives and Tasks of the Research

The objective of this thesis is to propose a multi-class classification method that would work well both with small sample sizes and imbalanced data sets with unknown data distributions. Besides, a thorough explanation of pros and cons has to be provided in order to not make it a “one more multi-class classification method”. Therefore the following tasks must be completed:

1. Elucidate the complexity and sample size issues of multi-class classification task by employing two-stage classification with simple pair-wise classifiers at the first stage.
2. Clarify the effects of inaccurate criteria employed to construct the fusion rules in the second stage of two-stage classification methods.
3. Accurately compare performance of different pair-wise classifier fusion methods.
4. Provide recommendations for multi-class classifier designer based on the research results.

1.5 Research Methods

Simple linear classifiers were chosen as pair-wise classifiers in order to make clear theoretical conclusions about the possibility of any gain in pair-wise classification fusion. Both – analytical analysis and modeling on generated and real world data were performed. The aspects of sample size, class imbalance and criteria of fusion rules were emphasized during analysis. Analytically obtained results were used in a difficult real world task. Each class data was reshuffled N_e = 250, 500 or even 1000 for simulation in order to get reliable estimates. The classification error rate was used as the performance estimate of the methods used.

1.6 Scientific Novelty

The novelty of this work is that there are clear explanations why pair-wise classifiers may be a good alternative to complex multi-class classifiers. The use of SLP as pair-wise classifier in classification system based on two-stage pair-wise classifiers was proposed by demonstrating its superiority over linear SVM both theoretically and experimentally. The new fusion method for pair-wise classification was proposed with analysis of advantages and

shortcomings. The clarification of difference between competing H-T and WLW methods was done. It was shown that pair-wise classifier fusion correction by noise injection is more promising than analytical correction of biasing. And a wide range of detailed comparison of various types of two stage multi-class classification methods based on pair-wise classifiers was performed.

1.7 Practical Significance of the Work

The practical significance of this thesis is that recommendations for the designer of multi-class classification task solution based on the research results were presented. The problems of sample size, imbalance, choosing proper multi-class classifier architecture, finding an optimal classifier are completely or partially solved by introducing two-stage pair-wise classification based on pair-wise single layer perceptrons.

1.8 Defended Theses of Dissertation

1. Two stage pair-wise based classifiers are a good alternative to complex multi-class classification algorithms.
2. Single layer perceptrons should be used as pair-wise classifiers instead of linear support vector machines.
3. Pair-wise Fuzzy Templates method may outperform other multi-class classification algorithms when statistical parameters of pairs of classes differ considerably, while it is not recommended in symmetric deployment of classes. The introduction of a scaling parameter may improve the performance of this method in both cases.

1.9 Approval of Research Results

Publications in internationally reviewed periodical issues:

- Sarunas Raudys, Rimantas Kybartas, Edmundas Kazimieras Zavadskas. *Multicategory Nets of Single-Layer Perceptrons: Complexity and Sample-Size Issues*. IEEE Transactions on Neural Networks, Vol. 21, No. 5, p. 784 – 795, 2010
- Rimantas Kybartas, Nurdan Akhan Baykan, Nihat Yılmaz, Sarunas Raudys. *Multiclass Mineral Recognition Using Similarity Features and Ensembles of Pair-wise Classifiers*. 23rd IEA-AIE Conference, 2010, Springer-Verlag, Lecture Notes in Artificial Intelligence

Publication in other periodical issues:

- Š.Raudys, R.Kybartas. *Daugelio klasių klasifikavimas naudojant vienasluoksnius perceptronus*. Informacinės technologijos 2007 : konferencijos pranešimų medžiaga 2007, p. 427-430.

1.10 Thesis Structure

Chapter 1 contains the introduction including research area, relevance and object, methods and scientific novelty of the thesis.

Chapter 2 defines the multi-class classification task and emphasizes the main problems. The most popular solutions using multi-class classifiers as well as some methods of multi-class classifier fusion are presented.

The two stage classification methods based on pair-wise classifiers are presented in the Chapter 3. A detailed review of popular pair-wise classifiers is presented. Well known pair-wise classifier fusion methods are reviewed dividing them into trainable and non-trainable ones. A new pair-wise classifier fusion method called *Pair-wise Fuzzy Templates* is also presented. A detailed analysis and experimental results for this new method are presented.

The core difficulties of classification tasks – small sample size and data class imbalance – are addressed in Chapter 4. These difficulties are solved by employing pair-wise classifiers. The ability of single layer perceptrons to deal with these difficulties best is demonstrated.

The results of previously analyzed two-stage classifiers with SLP and SVC as pair-wise classifiers and benchmark one-stage multi-class classifiers are presented in chapter 5. This chapter also discusses a few methodical issues on different method comparison such as using non-exact optimization criteria and a small number of experiments.

Chapter 6 deals with practical application of analytical results to complex geological data classification. The employment of similarity features proved its powerfulness even for dimension-sensitive pair-wise classifiers. Experimental results confirmed the obtained core results.

The conclusions and practical recommendations for multi-class classification task designers are listed in chapter 7.

Chapter 2

Multi-class Classification

The purpose of this chapter is to present the multi-class classification task whose solution approach is the core of this thesis. Multi-class classification task algorithms are also presented. The algorithms and their fusion methods were empirically selected based on the author's experience of the frequency with which they are mentioned in various publications.

In order to extinguish the ambiguity of the task to be solved, Section 2.1 deals with a clear definition of the multi-class classification task. Sections 2.2 and 2.3 review the multi-class classifiers and their fusion methods correspondingly.

2.1 The Multi-class Classification Task

Suppose the case where *d*-dimensional data vectors $x=\{x_1, x_2, \ldots, x_d\}$are analyzed. Each data vector belongs to one and only one of data classes Π_1, Π_2, …, Π_K where $K \geq 3$. Here classes Π_1, Π_2, …, Π_K are already predefined by a specialist of the field, i.e. there is no need for data clusterization. Some of the data vectors are already assigned to their classes (i.e. the data are already classified). The multi-class classification task is to assign a new data vector *x* to one of *K* classes using the information gained from the already classified data. The performance of a task solution is measured by the rate of incorrect assignments of these new vectors.

The data already assigned to classes will be called *training* data. And the data which is used in order to determine the error rate of classification algorithm will be called *testing* data.

One of the problems in multi-class classification task is that optimality may be lost when using complex multi-class classifiers.

The problem of unbalanced data sets of training data has recently been identified as a crucial one in machine learning and data mining. A higher degree of class imbalance increases the difficulty of multi-class classification [4]. It especially complicates the classification task for methods which assume a balanced distribution of classes. The empirical study in [4] revealed that almost all techniques are effective in a two-class case while most are ineffective for the multi-class task. And although many researchers declare that their conclusions concerning class imbalance problem could be applied to multi-class classification problem [55], there are actually only a few works concerning the class imbalance problem in the multi-class classification task.

2.2 Multi-class Classifiers

There are plenty of methods used for multi-class classification [1], [2] etc. – from simple statistical ones [3] to complex genetic algorithms [50] and fuzzy classification [51]. The obvious and practically (especially in some practical field tasks) the most often used approach of multi-class classification task solving is to make a multi-class classifier.

2.2.1 Artificial Neural Network

The best known multi-class classifier is artificial neural network [1], [2], [3], [54]. The idea of artificial neural networks came from the structure and behavior of neurons in the brain of a biological organism. Artificial neural networks are considered to approximate any function. In [53] in was shown that a standard multilayer feed-forward network with a locally bound piecewise continuous activation function can approximate any continuous function to any degree of accuracy if and only if the network's activation function is not a polynomial.

There are many types of ANNs. The most popular one is a feed-forward artificial neural network. It consists from of net of interconnected neurons. The nodes (neurons) in this ANN do not form a directed cycle. The network consists from three types of layers: the input layer, hidden layers and the output layer. There is only one input layer and one output layer, while the number of hidden layers is unlimited. But in practice only one hidden layer is usually used. The number of neurons in each layer is also unlimited and depends (as the number of hidden layers) the exact task and its complexity. The information being processed is passed to input layer nodes where the first preprocessing of data is performed. Then the outputs of the input layer multiplied by their weights are summed and passed to the first hidden layer. The outputs of the first hidden layer are passed in the same manner to the second hidden layer and so on until output layer produces the response of the ANN. Each node has some kind of activation function which is applied to the input before passing it to the nodes of the successive layer. The schematic structure of feed-forward artificial neural network is presented in Figure 1.

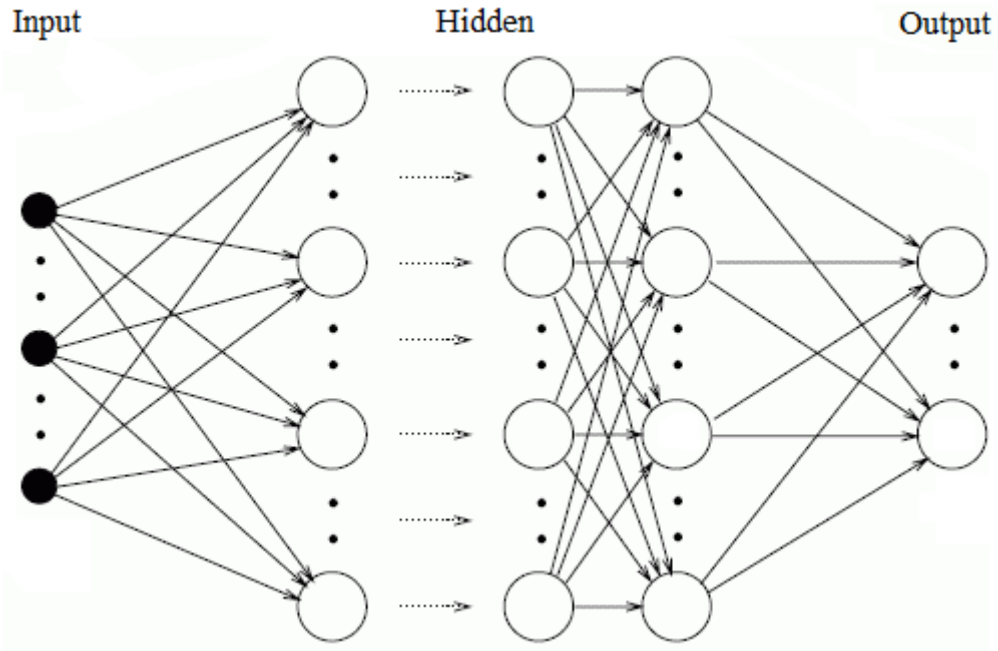


Figure 1. A simplified visualization of a feed-forward artificial neural network with its input, output and hidden layers.

Before applying the artificial neural network to the task it has to be trained (e.g. using backpropagation algorithm [57]) in order to determine the weights of the transitions between nodes. Some additional training may be applied in order to determine the number of nodes in each layer, their activation functions and the number of hidden layers.

Despite good theoretical approximation properties, ANNs are difficult to use in practice. The first problem of such practical employment is that for each task a particular architecture (the number of layers, the number of nodes in layers, activation functions) of ANN should be selected. This task requires a very large amount of computational resources. Another problem is the complex ANN structure may lose it generalization abilities when the sample data is small and the new data is a little bit different.

2.2.2 The Radial Basis Function Neural Networks

The radial basis function (RBF) neural networks are the second (after feedforward ANNs) most often used neural networks. The RBF-based classifier consists of three layers: input layer, a hidden radial basis function layer and a linear output layer [1], [2], [3], [14]. The advantage of this type of neural network is its partially fixed architecture. That's why it was chosen as one of benchmark methods in this thesis. Hidden radial basis layer is composed of G radial basis neurons that calculate $y_i = rad(\| \boldsymbol{C}_{i1} - \boldsymbol{x} \| / H_i)$, $i = 1, \ldots, G$, where rad is a transfer function for radial basis neuron (in experiments of these thesis the model of multivariate Gaussian distribution was used in experiments of this thesis), $\boldsymbol{C}_{i1}$ is the i-th "center" of the radial basis neuron, and H_i is the smoothing parameter. Output layer is linear: $o_i = \boldsymbol{w}_{i2}^T \boldsymbol{y} + b_{i2}$, where $\boldsymbol{y} = (y_1, \ldots, y_g)^T$, $\boldsymbol{w}_{i2}$ is the weight vector and b_{i2} is the bias term. The newly classified vector $\boldsymbol{x}$ is classified according to the maximum of outputs. The Matlab neural network (NN) toolbox was used to make the experiments with RBF networks. Artificial pseudo-validation sets were used to select parameters g and H_i.

2.2.3 Kernel Discriminant Analysis

Nonparametric kernel-based local estimates of conditional probability density functions, $f_{\mathrm{KDA}}(\boldsymbol{x}|\,\Pi_i)$, are applied (see Figure 2) in Kernel discriminant analysis (KDA).

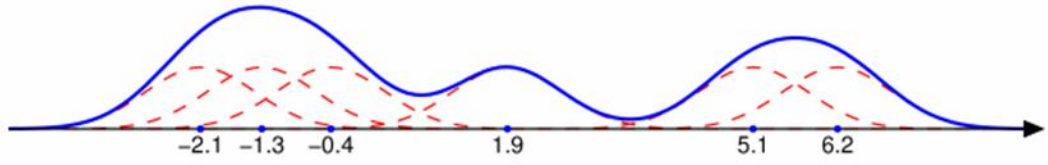


Figure 2. The probability function (in blue) estimated by local kernel-based estimates.

Independent decisions are performed at each point of the feature space ([3], [14]) in the classification phase. This method was used as a benchmark for the experiments of this research. The Gaussian kernel was used and classification according to the maximum of products

$$q_i \times f_{\mathrm{KDA}}(\boldsymbol{x}\,|\,\Pi_{\mathrm{i}}) = \frac{q_i}{N_i}\sum_{j=1}^{N_i}\exp(-h^{-1}(\boldsymbol{x}-\boldsymbol{x}_{ij})^T(\boldsymbol{x}-\boldsymbol{x}_{ij})), \quad (i = 1, 2, \ldots, K,\ j = 1, 2, \ldots, N_i), \tag{1}$$

was done. Here h is a smoothing parameter, q_{i} is a priori probability and N_i is number of samples of class Π_{i}. To use this method as a benchmark and to have truthful comparison of the KDA-based algorithm with the other methods some parameters had to be set. After training set based decorrelation of the data, standard deviations, s_i, of the features at first ($s_i \approx 1.0$) were normalized, and default value $h = 1.0$ was used.

2.3 Fusion of Multi-class Classifiers

As seen in the short review of multi-class classification methods, there are many new approaches for multi-class problem solving. Some of them are better in one case, others – in other. Naturally, researchers were trying to

employ the best features of different classifiers. This way the classifier fusion methods appeared.

Some approaches suggest splitting multi-class classifiers into several binary ones. The split into *K*-1 binary classifiers is proposed in [52]. First the binary classifier classifying one class from all others is used. If the output of this classifier is on behalf of all other classes, the second binary classifier classifying the second class from the remaining classes is used etc. The critical issue of this "one class at a time" approach is the planning of the removal sequence.

2.3.1 Behavior Knowledge Space

A method called *Behavior Knowledge Space* (BKS) was developed in [58]. Let us have L different multi-class classifiers C_j, j=1, …, L separately trained to solve a particular classification task. Let $C_j(x)$ be the class label assigned by classifier C_j to data vector x. Thus each data vector may be represented by a discrete-value vector C=($C_1(x)$, …, $C_L(x)$). The number of possible combinations is $c = K^L$. The BKS method considers each combination as a cell in the BKS table, which is designed by the training set. The table is filled with the class labels which are the mostly representative for the combination representing the cell. The decisions are done upon the label of the obtained combination in the testing data. When the value of a testing data vector falls into an empty cell (not obtained during training), rejection occurs. A threshold on the probabilities of the most representative class may be used in order to control the reliability of decision.

2.3.2 Fuzzy Templates

The aim of Fuzzy Templates [13] is to fuse continuous outputs of several K-category classifiers C_1, C_2, …, C_L. Let $d_{i,j}(x)$ for any data vector x denote the output of classifier C_i, assuming that x belongs to class Π_j, j=1…K. The decision profile for data vector x is made:

$$DP(x) = \begin{bmatrix} d_{1,1}(x) & \dots & d_{1,j}(x) & \dots & d_{1,K}(x) \\ \dots & \dots & \dots & \dots & \dots \\ d_{i,1}(x) & \dots & d_{i,j}(x) & \dots & d_{i,K}(x) \\ \dots & \dots & \dots & \dots & \dots \\ d_{L,1}(x) & \dots & d_{L,j}(x) & \dots & d_{L,K}(x) \end{bmatrix} \quad (2)$$

Then a fuzzy template $F_i = \{f_i(l,k)\}$ for each class is formed. F_i is an *LxK* matrix with elements

$$f_i(l,k) = \frac{\sum_{j=1}^{N} Ind(x_j, i) d_{l,k}(x_j)}{\sum_{j=1}^{N} Ind(x_j, i)} \quad (3)$$

where $x_1, \dots, x_N$ are crisply labeled data and function $Ind(x_j, i)$ is an indicator function with value equal to 1 if x_j belongs to class Π_i and 0 otherwise.

When a new data vector *x* is submitted for classification, its individual decision profile is calculated and a new *K*-dimensional vector with distance elements

$$\mu^i(x) = 1 - \frac{1}{LK} \sum_{l=1}^{L} \sum_{k=1}^{K} (f_i(l,k) - d_{l,k}(x))^2 \quad (4)$$

is obtained. The classification decision is made according to maximum index *i* of the vector element.

2.3.3 Pair-wise Fusion Matrix

The authors of the Pair-wise Fusion Matrix method [8] maintain that their method may overcome the problems where the BKS might not be applicable. This method only analyzes pairs of *L* classifiers and calculates probabilities

$$P(l \mid c(i), c(j), x) = n(x \in l, c(i), c(j)) / n(c(i), c(j)),$$

where $c(i)$ and $c(j)$ are decisions of classifiers C_i and C_j over a data vector *x* and $n(c(i), c(j))$ is the total number of samples on which classifier C_i gives crisp output $c(i)$ and classfier C_j gives crisp output $c(j)$, while $n(x \in l, c(i), c(j))$ is the number of samples with real class label *l*. These probabilities form a pairwise fuzzy matrix. When the new vector *x* arrives for classification, the

decision is done in favor of the class with maximum probabilities, i.e. $k = \arg \max_{l=1}^{K} \max_{i,j=1, i \neq j}^{L/2} P(l \mid c(i), c(j), x)$.

2.4 Concluding Remarks

In this chapter, the multi-class classification task was defined. The overview of multi-class classifiers and their fusion methods was presented as well. It was shown that there may be many solutions for each multi-class classification task from well known classical neural networks to a complex net of multi-class classifiers.

There is a lack of information or any kind of recommendations when to use one or other method. Thus there is need for such recommendations or some approach for an abstract and universal method.

Chapter 3

Two Stage Pair-wise Based Classification

In this chapter the superiority of pair-wise classification methods is pointed out. Apart from standard statistical classifiers there are only a few popular types of classifiers which classify two classes: single layer perceptrons and support vector machines. But there are plenty of the fusion methods and it is hard to decide which ones are the best for some particular task. Usually most of pair-wise classifier fusion methods are based on some kind of voting or probability estimation strategy, while the estimation of classifier output value is forgotten. A new strategy based on classifier output similarity similar to a multi-class classifier fusion method called *Fuzzy Templates* is presented.

The presumptive arguments for two stage pair-wise based classifier superiority over single stage multi-class classifier are presented in Section 3.1. Section 3.2 contains an overview of standard statistical classifiers. Popular pair-wise classifiers well known in the field of artificial intelligence are presented in Section 3.3. A wide overview of various pair-wise classifiers is presented in Sections 3.4-3.6. Some consideration about classifier output and fusion method choosing are discussed in Section 3.7 and Section 3.8. The new pair-wise classifier fusion method is presented and thoroughly analyzed in Section 3.9.

3.1 Superiority of the Pair-wise Classifiers

Let's analyze a simple K category SLP network. A network of K single layer perceptrons [1]-[3] has K outputs, $o_i = f(v^T x + w_0) = f(w^T z)$, ($i = 1, 2, \ldots, K$), where v is a d-dimensional weight vector, w_0 is a bias weight, $w = (w_0, v^T)^T$, $z = (1, x^T)^T$, and $f(s)$ is a nonlinear sigmoid activation function, i.e.

$f(s) = 1/(1+\exp(-s))$. For more information about single layer perceptron see Section 3.3.1.

To obtain the weight vectors, the cost function of the sum-of-squares is minimized

$$cost = \frac{1}{N_1 + N_2 + ... + N_K} \sum_{h=1}^{K} \sum_{j=1}^{K} \sum_{s=1}^{N_j} [t_{jh} - f(\mathbf{z}_{js}^T \mathbf{w}_h)]^2 \quad (5)$$

where t_{jh} stands for desired output. The following values were chosen for the sigmoid activation function,: $t_{jh} = 1$ if $h = j$, and $t_{jh} = 0$ if $h \neq j$. In multi-class case K terms $\sum_{h=1}^{K} \sum_{s=1}^{N_h} (t_{jh} - f(\boldsymbol{z}_{js}^T \boldsymbol{w}_h))^2$, $(j = 1, ..., K)$ of loss function (5) are minimized independently.

In pair-wise classification of classes Π_i and Π_j, the $(d+1)$-dimensional weight vector $\boldsymbol{w}_{ij}$ is being searched for, taking into account training vectors of these two pattern classes. If the result of minimization of loss (5) is used to perform *pair-wise classification* of classes Π_i and Π_j, $\boldsymbol{w}_{ij}^{\mathrm{B}} = \boldsymbol{w}_i - \boldsymbol{w}_j$ have to be employed. Training vectors of all other classes affect the components of vectors $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$. For that reason, discrimination of separate pairs of the classes by standard net of K SLPs can become notably worse as that performed by individually trained pair-wise perceptrons.

Figure 3(a) demonstrates such a situation with a two-dimensional (2D) example. Each single decision boundary (a line) is formed by outputs of two SLPs. In the 2D feature space, three pair-wise discrimination lines intersect at point O. An implicit requirement concerning the intersection is a severe constraint. This restriction follows from the criterion of the sum of squared errors (5). Furthermore, the result of this restriction is that at times *the hyper-planes classify the pairs of classes unsatisfactorily.* The area AOC is attributed to class Π_1. The area COB is attributed to class Π_2, and the remaining area, BOA, is attributed to class Π_3.

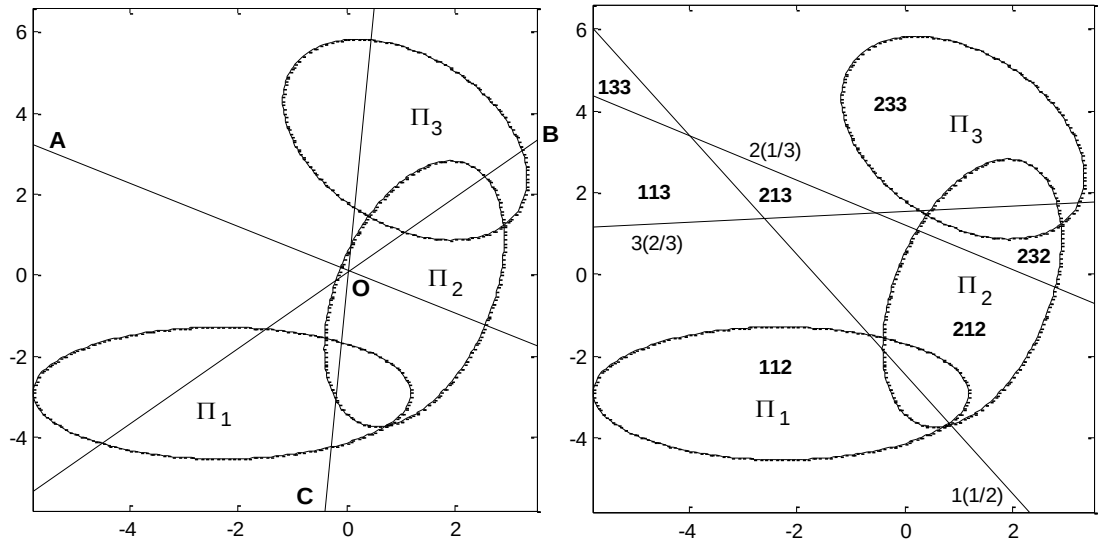


Figure 3. Classification regions formed by: a) a net of "three-node SLP" aimed at classifying three classes; b) three separate pair-wise SLPs.

Consider a two-stage decision making procedure where a pair-wise classification is performed at the first stage and a class label is assigned to each single region formed by the pair-wise classifiers. Three pair-wise SLP based linear classifiers, 1(1/2), 2(1/3) and 3(2/3), split the 2D space into seven regions in Figure 3(b). Every region is marked by a triplet of digits, 1, 2, or 3. The first digit indicates the class label assigned by the first binary classifier (class Π_1 or Π_2), etc.

If two binary decision rules agree, classification of an unknown vector, x, is simple: x is assigned to the class with majority assignments. If all the classifiers indicate diverse class labels, ambiguity arises. This situation occurs in the region marked by 213. In the current example, the ambiguous region is empty. Therefore, there is hope that in such a two-stage decision making procedure the results will be near to optimal. If the ambiguity region is not empty, problems may arise.

A proper initialization and early stopping are powerful tools that reduce the classifier's complexity and generalization error [1] - [3], [14], [16] - [18]. In the two-category (binary) situation, one can acquire extra benefits. If the sample mean vector of the pair of the classes is moved to a zero point, and the

training starts with initial weight vector composed of zero components, the Euclidean distance classifier (EDC) is obtained just after the first total gradient training iteration. Further, *SLP evolves through more complex classification* algorithms: regularized discriminant analysis, the Fisher, robust, minimum empirical error classifiers. Hypothetically, one may even approach the maximum margin (support vector) classifier after suitable training [3], [17]. A successfully defined initial weights vector contains useful information which could be saved if the training is topped in time [16]. By performing *a whitening data transformation* based on specially constructed sample estimates of covariance matrix (CM) of the input data prior to training the perceptron a valuable initial weight vector can be obtained after the first iteration [3]. In principle, by prudent controlling parameters of the covariance matrix estimation and learning algorithm one can obtain the classifiers of suitable complexity. The simplest and most reliable way to determine the optimal moment to stop training is the employment of a validation set.

Another merit of pair-wise classifiers is that unclear class imbalance problem in multi-class classification is transferred to two-class level where much more research is done and more promising results are obtained.

3.2 Statistical Classifiers and Their Implementation

Statistical decision theory proposes a method to design optimal *K*-category classifiers, provided that probability density functions and prior probabilities $q_1, q_2, \ldots, q_K$ of the classes are known. Assuming that distribution densities of the *d*-dimensional input feature vector *x* are multivariate Gaussian characterized by different mean vectors $m_1, m_2, \ldots, m_K$ and common covariance matrix (CM) Σ, an optimal *K*-category linear classifier [1], [14] is obtained. In practice, unknown parameters $m_1, m_2, \ldots, m_K$ and $\boldsymbol{\Sigma}$, are replaced by training set based estimates $\bar{x}_1, \ldots, \bar{x}_K$ and *S*. In the *K*- category case, a

standard Fisher linear discriminant function (DF) actually is a set of K functions

$$g_i^{\mathrm{LDF}}(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{v}_i + w_{0i}, \quad (i = 1, 2, \ldots, K) \tag{6}$$

where weight vector $v_i = S^{-1}\bar{\boldsymbol{x}}_i$, bias term $w_{0i} = -\bar{\boldsymbol{x}}_i^T S^{-1}\bar{\boldsymbol{x}}_i + \ln q_i$, and an allocation of vector, $\boldsymbol{x}$, is performed according to the maximum of $g_1^{\mathrm{LDF}}(x)$, $g_2^{\mathrm{LDF}}(x)$, …, $g_K^{\mathrm{LDF}}(x)$.

If the dimensionality d is very high, and training set sizes $N_1, N_2, \ldots, N_K$ are too small to obtain a reliable estimate of the $d \times d$ matrix Σ, then small positive constants, λ, are added to the diagonal elements of matrix S. This way the *regularized discriminant analysis* (RDA, see [3], [14]) is obtained. If $\lambda \to \infty$, one ignores the matrix S and obtains *Euclidean distance classifier* (EDC). The EDC rule is simpler than RDA, and RDA is simpler than Fisher DF. If CMs of the classes are diverse, quadratic DFs should be used instead of the linear. Quadratic DFs are very sensitive to training set size – covariance matrices are supposedly diverse and a large number of samples to obtain estimates S_1, S_2, …, S_K is needed. One of practically useful remedies is to use pair-wise classifiers with pooled sample estimates of the covariance matrices. For the pair Π_i and Π_j $S_{ij}(\alpha_{ij}) = S_i + S_j\alpha_{ij}$ are used and determine the bias terms in a special way (the Anderson-Bahadur (A-B) linear DF [3], [14]). In this type of linear classifiers, classes Π_i and Π_j use "common" covariance matrix. For that reason, the A-B rule has much better small sample properties in comparison with quadratic DFs [15]. The SLP based classifier can realize the Fisher DF and – in further training – the Anderson-Bahadur DF. Most often it performs well even when $S_j \neq S_i$.

3.3 Pair-wise Classifiers

The idea of pair-wise multi-class classification is to reduce a complex multi-class classification problem to many simple classification tasks where

simpler pair-wise classifiers classifying only two classes of data (omitting anything else) are used.

Pair-wise classifiers are the binary classifiers that classify data into only two classes, i.e. the classification function is a decision boundary separating two different data sets. Let's denote $C_{i,j}$ as a base binary classifier which separates classes Π_i and Π_j. The classifiers $C_{i,j}$ are class-symmetric i.e. $C_{i,j} = 1-C_{j,i}$ (just for the classifiers used in this thesis - it may be not true in general). If pair-wise classification is used for a K class classification, then there are $(K-1)K/2$ pair-wise classifiers.

3.3.1 Single Layer Perceptron

The single layer perceptron is a mathematical model of a biological perceptron. SLP may be expressed as $f(v^T x + w_0)$, where v and w_0 are weight vector and bias, obtained during perceptron training, x is a d-dimensional data vector, and f is the output activation function. There are many types of activation functions, e.g. sigmoid, hyperbolic tangent, signum or step functions. The sigmoid function (7) was chosen as the activation function in this research, see Figure 4.

$$f(x) = \frac{1}{1+e^{-x}} \tag{7}$$

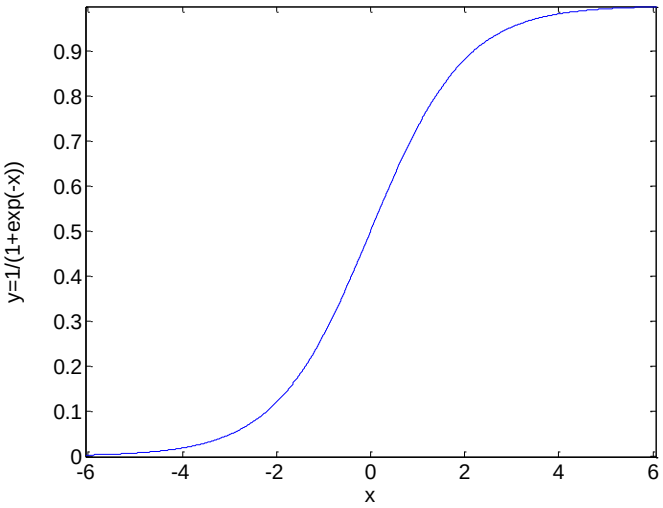


Figure 4. Sigmoid activation function

This was done because of the SLP behavior evolvement through seven different statistical classifiers while being trained using the sigmoid activation function [17].

While using sigmoid activation function when weights of SLP are equal to zero or very tiny, the output of SLP is approximately equal to 1/2. When weights grow, the output of SLP approaches 0 or 1 (depending on the sign of x in Equation (7)), i.e. indication of belonging to one or other class in two-class case. Thus in this research mostly the output weighted sum of input vector (i.e. $v^T x + w_0$) was enough since the sign of sum provides enough information in pair-wise classification to make a decision. Therefore it may be considered that in such cases the output function was signum (8) in such cases.

$$f(x) = \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases} \quad (8)$$

But it should be pointed out that despite the use of signum function in decision stage, the single layer perceptron was trained using sigmoid function (7).

In order to obtain SLP weights and bias, iterative gradient descent approach [1], [3] was used in this study. So the longer the SLP is trained, the higher are weights and the closer the values of SLP output are to 0 or 1.

It is necessary to stop at the proper time in order to get optimal weights and bias while training SLP – not to undertrain (not yet good enough to classify data) and not to overtrain (biased to the training data and not able to properly classify new data). Validation data is used to optimally stop SLP. The weights and bias that provide the least classification error rate on this validation data are selected as the proper ones. Usually it is a luxury to divide data into training, testing and validation sets. In order to make validation data, noise injection technique [19] was used in this research (for more see Section 3.3.4).

While training nonlinear single layer perceptron by gradient descent algorithm, one may obtain seven well known statistical classifiers [3], [17] which are optimal for particular data sets. If the mean of training data is set to zero and initial weights of SLP are also zero, Euclidean distance classifier is obtained after the first iteration. Afterwards, the SLP evolves to linear regularized discriminant analysis, standard linear Fisher classifier or the Fisher classifier with a pseudo-inverse of covariance matrix. After that, the SLP approaches robust discriminant analysis and at the end, when the perceptrons weights become large, one may approach the minimum empirical error or maximum margin (i.e. support vector) classifiers.

When classifying vector x between two classes, the output value of pair-wise classifier for that vector depends on position, i.e. whether it is considered as a vector from the first or the second of two classes. In order to avoid this confusion, the direct output of SLP (as well as SVC, see the next subsection) has to be taken when the vector is considered to belong to the first class. And the value 1-"classifier output" has to be taken when vector x is considered to belong to the second class.

3.3.2 Support Vector Classifier

Support vector classifier was proposed by V. Vapnik et al. [29] in 1992. It solves the following primal problem:

$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$ subject to $y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i$, $\xi_i \geq 0$, $i=1, \ldots, l$	(9)

where $y_i \in \{1,-1\}$ is the label of one of l training vectors x_i, $C>0$ is the upper bound, and $\phi(x_i)$ is a kernel function.

Since SLP is a linear method in the context of classification tasks, linear SVC was selected for experimental and theoretical comparison. The regularization parameter C was selected using grid search on validation data. As suggested in [21], a set of $[2^{-7}, 2^{-6}, \ldots, 2^{10}]$ was used for possible C values. Validation data for the selection of C was generated the same way as validation data for SLP stopping (see the above subsection), i.e. using colored noise injection technique [19]. The weighting of classes was also employed. The weight assigning method (used most often [22]) when weights are set inversely proportional to the sample count of each class was applied. SVC classifiers realized in *LIBSVM* library [20] were used.

After SLP or SVC is trained, one obtains weights and bias (v, w_0). It is proposed to use modified SVC outputs (i.e. $v^T x + w_0$) before applying them to the *Pair-wise Fuzzy Templates* in these theses. The modification has to be done by applying sigmoid function (7) to SVC outputs. Such modification proved its superiority over the use of pure SVC outputs.

Preliminary data transformations for SV and SLP based classifiers were performed in the same way: the input features were roughly decorrelated, standard deviations of them were set to 1 and means to 0.

3.3.3 Decision Trees

Decision tree classifiers are the easiest to represent and understand. They are graphs with particular rules in the nodes. When a new data vector x to be classified arrives, it is passed to the root node, and depending on the decision of the rule, it is passed to another child decision-node until a leaf node of the tree representing the class label is reached. The merit of decision trees against

other classifiers is that there are rules in the nodes of the tree readable for humans. Sometimes this may help to better understand the difference between pattern classes.

The ID3 algorithm [59] presented by J. R. Quinlan may be used for decision tree classifier training.

3.3.4 Noise Injection

While training classifiers, optimal parameters have to be obtained. But to use the same training data both for training and parameter estimation is not recommended since (especially in cases when small training data set is obtained) it is easy to get biased classifier parameters. It means that the classifiers may be too adapted to the provided data and malfunction with new data not used in training and parameter selection. In order to avoid this unfavorable situation, the data provided for classifier's training has to be spitted into two parts – the training set used for classifier training and the validation set used for parameter estimation.

The split of training data is not a problem, when huge amount of data is obtained. But in small training data cases it is a luxury to split the data into training and validation sets. One possible solution to bypass the small sample size complication is to use all training set vectors as the training set and form a (virtual) validation set from the training vectors by means of a noise injection.

The simplest solution of noise injection is to add random additional points for each point in random direction within a predefined distance. This way a white noise injection is obtained. But white noise adds "clouds" around each point and distorts the data geometry. A colored *k*-NN noise injection was suggested for reducing the data distortion [19]. *A colored noise injection actually introduces additional information that declares: a space between the nearest vectors of a single pattern class is not empty, but instead it is filled up with vectors of the same category*. To generate such a noise, *k* nearest neighbors of the same pattern class are found for each single training vector $\boldsymbol{x}_{js}$. An artificially generated noise only in a subspace formed by the vector $\boldsymbol{x}_{js}$ and

k neighboring training vectors, $\boldsymbol{x}_{js}^{1}$, $\boldsymbol{x}_{js}^{2}$, …, $\boldsymbol{x}_{js}^{k}$ is added. Random Gaussian, N(0, $\sigma_{\text{noise}}^{2}$), variables are added ni_{nn} times along the k lines connecting vector $\boldsymbol{x}_{js}$ and $\boldsymbol{x}_{js}^{1}$, $\boldsymbol{x}_{js}^{2}$, …, $\boldsymbol{x}_{js}^{k}$. Three parameters are required to realize the colored noise injection procedure: 1) k, the number of neighbors to be taken into account while generating noise, 2) ni_{nn}, the number of new, artificial vectors generated around each single training vector $\boldsymbol{x}_{js}$, and 3) σ_{noise}, the noise standard deviation. As preliminary results showed, optimal values of these parameters may depend on the data. But the selection of optimal parameters for each data was out of range of this study. Thus the values of colored noise injection parameters were empirically selected to be k=2, ni=2 and σ_{noise}=1.0.

3.4 Pair-wise Classifier Fusion Methods and Their Implementations

In order to fuse the pair-wise decisions, a number of methods have been developed. There are mainly two types of fusion methods: trainable and not trainable fusion rules. The first ones need to be trained in order to be exploited – i.e. some parameters have to be adjusted according to particular fusion rule training data. Non-trainable fusion rules do not require any additional efforts or data for their adjusting – they just use some rule on pair-wise classifier output in order to make final classification decision.

3.5 Non-trainable Pair-wise Fusion Rules

3.5.1 Voting

There are mainly two voting strategies for pair-wise based classifiers: one-against-one (or “max wins”) principle and one-against-others. As it was already mentioned, one-against-all voting strategy may be considered as a multi-class single layer perceptron, since this way five pair-wise SLPs

considering some particular class as the first one, and all the other classes as the second one, are obtained. The pure pair-wise voting strategy uses one-against-one strategy where $K(K\text{-}1)/2$ pair-wise classifiers are used. Besides it was shown [5] that this strategy is more promising than one-against-all.

Any simple voting rule does not require training. The one-against-one voting strategy deals with $K(K\text{-}1)/2$ pair-wise classifiers and assigns the label to the vector according to the majority of the votes.

3.5.2 Directed Acyclic Graph

The idea of DAGSVM [11] is to organize pair-wise SVMs in rooted binary direct acyclic graph to make final decision. Decision making process in directed acyclic graph (DAG) may be considered as a sort of a decision tree. When a vector is submitted for classification, it is firstly evaluated by the root classifier (root DAG node). Subsequently, decision making is passed to the left or right node depending on the current node decision until one of K nodes with no children is reached. This node labels a new vector. In experiments of this thesis, the pair-wise SLP's were also used instead of SVMs. This way a generalization of the DAG algorithm for any other type of the pair-wise classifiers was done. After getting the pair-wise classifiers, $C_{1,K}$ was set as the root classifier. When a new vector is submitted for classification, the output of the root classifier is calculated. If the output is 1, then vector is redirected to the classifier $C_{1,K-1}$ or to the classifier $C_{2,K}$ otherwise. And so on: if the output of classifier $C_{i,j}$ is i, then $j=j\text{-}1$ else $i=i+1$ and if $i \neq j$, the vector to be classified is redirected to the classifier $C_{i,j}$. If $i=j$, then the output of the DAG is i.

3.5.3 The Quick Weighted Voting

A method for a quick fusion of pair-wise classifier decisions called *QWeigted* was proposed in [49]. The main idea is that there is a moment when

evaluating the ranks (number of votes) of classes by pair-wise classifiers when a class may be excluded from further evaluation due to its having already gained a low rank. During such fusion procedure the pair-wise classifiers are selected according to a so called *loss* value $l_i = e_i - r_i$ where e_i is the number of evaluated incident classifiers of class Π_i and r_i is the current rank (i.e. amount of votes) of class Π_i. First, the pair-wise classifier $C_{a,b}$ with minimal losses l_a and l_b of the relevant classes Π_a and Π_b is selected, provided that the classifier $C_{a,b}$ has not yet been evaluated. When a few classes have the same minimal loss and no further distinction exists, a class from this set is selected randomly. The losses l_a and l_b are updated based on the evaluation returned by $C_{a,b}$. These two steps will be repeated until all classifiers for the class Π_n with the minimal loss have been evaluated. Thus the current loss l_n is the correct loss for this class. As all other classes already have a greater loss, Π_n is the correct top rank class and the output of the *QWeighted* algorithm is n.

3.6 Trainable Pair-wise Fusion Rules

3.6.1 Hastie-Tibshirani

To classify an input vector $\boldsymbol{x}_z$, one needs to estimate posterior probabilities $p_1, p_2, \ldots, p_K$ of classes $\Pi_1, \Pi_2, \ldots, \Pi_K$. The Hastie-Tibshirani (H-T) method [6] utilizes information contained in $K(K\text{-}1)/2$ conditional probabilities $\mu_{ijz} = \text{Prob}(\boldsymbol{x}_z \in \Pi_i \mid \Pi_i, \Pi_j)$, $\mu_{jiz} = \text{Prob}(\boldsymbol{x}_z \in \Pi_j \mid \Pi_i, \Pi_j) = 1 - \mu_{ijz}$ and minimizes the sum of Kullback-Leibler distances between estimates $\hat{r}_{ij}$ and true probabilities μ_{ij}

$$D_{\text{KL}}(p_1, p_2, \ldots, p_K, \boldsymbol{x}_z) = \sum_{i \neq j} (N_i + N_j) \hat{r}_{ij} \log(\hat{r}_{ij} / \mu_{l_j}), \quad \text{subject to } \sum_{i=1}^{K} p_i = 1, p_i \geq 0. \tag{10}$$

Note that μ_{ijz} and μ_{jiz} define *a posteriori* probabilities conditioned to a particular vector $\boldsymbol{x}_z$. The *a posteriori* probabilities μ_{ijz} and μ_{jiz} should not be

confused with the probabilities of misclassification P_{ij} and P_{ji} when the pair-wise discriminant function, $g_{ij}(\boldsymbol{x}_z) = x_z^T \boldsymbol{x}^T \boldsymbol{w}_{ij} + w_{0ij}$, is used for classification of vectors from classes Π_i or Π_j.

To estimate probabilities μ_{ijz} and μ_{jiz}, after obtaining weight vectors $\boldsymbol{w}_{ij}$, and threshold terms w_{0ij} of the linear pair-wise classifier for each pair of the classes, Π_i and Π_j, one calculates the output of decision rule, $g_{ij}(\boldsymbol{x}_z) = x_z^T \boldsymbol{w}_{ij} + w_{0ij}$. In this thesis, the probabilities μ_{ijz} and μ_{jiz} will be evaluated under the assumption that weighted sums $g_z = g(\boldsymbol{x}_z)$, are Gaussian distributed with means m_i, m_j, and standard deviations s_i, s_j:

$$\text{Prob}(\boldsymbol{x}_z \in \Pi_i \mid \Pi_i, \Pi_j) = \frac{\phi(x_z \mid m_{i/ij}, s_{i/ij})}{\phi(x_z \mid m_{i/ij}, s_{i/ij}) + \phi(x_z \mid m_{j/ij}, s_{j/ij})}, \tag{11}$$

where $\phi(g_z \mid m_i, s_i) = \frac{1}{\sqrt{2\pi} s_{i/ij}} \exp\{-\frac{1}{2}(g_z - m_{i/ij})^2 / s_{i/ij}^2\}$ stands for standard Gaussian density function at point $g(\boldsymbol{x}_z) = x_z^T \boldsymbol{w}_{ij} + w_{0ij}$, $\hat{m}^{\mathrm{R}}_{i/ij}$, $\hat{m}^{\mathrm{R}}_{j/ij}$ and $\hat{s}_{i/ij}$, $\hat{s}_{j/ij}$ are the means and standard deviation of $g(\boldsymbol{x}_z) = x_z^T \boldsymbol{w}_{ij} + w_{0ij}$ estimates from training set (in some of the experiments, these parameters were estimated from pseudo-validation or test sets).

Then the estimates of the two pair-wise conditional probabilities are:

$$\begin{aligned} \hat{r}_{ijz} &= \frac{\phi(x_z \mid \hat{m}^N_{i/ij}, \hat{s}_{i/ij})}{\phi(x_z \mid \hat{m}^N_{i/ij}, \hat{s}_{i/ij}) + \phi(x_z \mid \hat{m}^N_{j/ij}, \hat{s}_{j/ij})}, \\ \hat{r}_{ji} &= \frac{\phi(x_z \mid \hat{m}^N_{i/ij}, \hat{s}_{i/ij})}{\phi(x_z \mid \hat{m}^N_{i/ij}, \hat{s}_{i/ij}) + \phi(x_z \mid \hat{m}^N_{j/ij}, \hat{s}_{j/ij})}. \end{aligned} \tag{12}$$

If *the multi-class data is balanced*, i.e. the number of training vectors $N_1 = N_2 = \ldots = N_K = N$, for each single vector $\boldsymbol{x}_z$ to be classified, then $\hat{r}_{ijz}$ is calculated, where $i, j = 1, 2, \ldots K$. After that sums $\hat{r}_{\mathrm{i.z}} = \sum_{j=1:K,\ j \neq i} \hat{r}_{ijr}$ are found and allocation according to the maximum of $\hat{r}_{1\cdot z}, \hat{r}_{2\cdot z}, \ldots, \hat{r}_{K\cdot z}$ is performed [6], [7].

If *the data is unbalanced*, then the Kullback-Leibler distance between $\mu_{i,j}$ and estimates $r_{i,j}$ needs to be minimized. The following iterative algorithm [6] was used to find $\mu_{i,j} = \Pr ob(i \mid j)$:

Start with some initial $\hat{\phi}_i(x \mid m_i, s_i)$ and the corresponding value of $\hat{\mu}_{i,j}$.

1. Repeat (i =1, 2, …, K, 1, …) until the convergence:

$$\hat{\phi}_i(x \mid m_i, s_i) \leftarrow \hat{\phi}_i(x \mid m_i, s_i) \frac{\sum_{j \neq i} n_{ij} r_{ij}}{\sum_{j \neq i} n_{ij} \hat{\mu}_{ij}}$$

renormalize $\hat{\phi}_i(x \mid m_i, s_i)$ and recalculate corresponding $\hat{\mu}_{ij}$.

2. $\hat{\phi}(x \mid m, s) \leftarrow \hat{\phi}(x \mid m, s) / \sum \hat{\phi}_i(x \mid m_i, s_i)$.

Estimates $r_{i,j}$ are evaluated by using the following expression (see [7], Section 2.4):

$$r_{i,j}(\boldsymbol{x}_z) = \frac{1}{1 + \sqrt{\frac{wc_i w}{wc_j w}} \exp\left(-\frac{1}{2} \times \left(\frac{(w(\boldsymbol{x}_z) - w(\boldsymbol{m}_j))^2}{wc_j w} - \frac{(w(\boldsymbol{x}_z) - w(\boldsymbol{m}_i))^2}{wc_i w} \right) \right)} \quad (13)$$

where $wc_i w = \boldsymbol{w}_{ij} * S_i * \boldsymbol{w}_{ij}^T$, S_i is the sample covariance matrix for the i-th class, $w(\boldsymbol{x}_z) = \boldsymbol{v}_{ij} \boldsymbol{x}_z + w_{ij}$, $w(\boldsymbol{m}_j) = \boldsymbol{v}_{ij} \boldsymbol{m}_j + w_{ij}$, $\boldsymbol{v}_{ij}$ and w_{ij} are the weight vector and the bias term of the pair-wise classifier of classes Π_i and Π_j, $\boldsymbol{m}_i$ is the sample mean vector of class Π_i.

3.6.2 Wu, Lin and Weng

Wu, Lin and Weng (WLW) [7] introduced two new algorithms of finding $\phi_i(\boldsymbol{x}_z \mid m_i, s_i)$ and the corresponding $\mu_{i,j} = \Pr ob(i \mid j)$. The second one was chosen in this thesis since the second method performed better in 11 cases out of 14 ones in empirical comparisons [7],. This algorithm minimizes the sum of

squared differences between weighted estimates of the pair-wise conditional probabilities

$$D_{\mathrm{WLW2}}(p_1, p_2, \ldots, p_K, \boldsymbol{x}_z) = \sum_{i \neq j} (p_i \hat{r}_{ji} - p_j \hat{r}_{ij})^2$$
$$\text{subject to } \sum_{i=1}^{K} p_i = 1,\ p_i \geq 0. \tag{14}$$

Consider $r_{i,j}$ defined in (13) as the estimate of $\mu_{i,j}$, and define matrix $Q = \{Q_{ij}\}$, where $Q_{ij} = \sum_{s:s \neq i} r_{si}^2$ when $j = i$ and $Q_{ij} = -r_{ji} r_{ij}$ when $j \neq i$. Now the algorithm is:

1. Start with some initial $\phi_i = \phi_i(x \mid m_i, s_i)$, $\sum_{i=1}^{k} \phi_i = 1$.
2. Repeat (i = 1, 2, …, K, 1, …)

$$\phi_i \leftarrow \frac{1}{Q_{ii}} \left[- \sum_{j: j \neq i} Q_{ij} \phi_j + \phi^T Q \phi \right], \text{ normalize } \phi, \text{ until the condition}$$

$$\begin{bmatrix} Q & e \\ e^T & 0 \end{bmatrix} \begin{bmatrix} \phi \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{15}$$

where $\phi = (\phi_{1,\ldots,} \phi_K)$, $b = -\phi^T Q \phi$ and e is the K (number of classes) length vector of ones, is satisfied.

In the experiments of this thesis the accuracy parameter ε=0.001 and difference

$$diff(i) = \begin{bmatrix} Q & e \\ e^T & 0 \end{bmatrix} \begin{bmatrix} \phi \\ b \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{16}$$

were defined.

It was considered that equality (15) was satisfied if $\forall i = \{1,2 \ldots K\}, |diff(i)| < \varepsilon$. The maximum number of iterations for this algorithm was also set. If after 100 iterations of estimation (15) the condition (16) was not satisfied, the algorithm was terminated anyway.

To avoid division by zero when $Q_{ii} = \sum_{s:s\neq i} r_{si}^2 = 0$, Q_{ii} was set to be equal to 0.000001.

3.6.3 Resemblance Model

Hamamura et al. proposed a *Resemblance Model* method based on virtual classes in [28]. Let's examine *K*=3 example where all three classes are located in such a manner that they form an isosceles triangle. Let's divide class Π_1 into virtual classes Π_{12} and Π_{13}, where class Π_{12} contains data vectors which are closer to class Π_2 and the class Π_{13} contains data vectors which are closer to class Π_3. Let's make the same division for classes Π_2 and Π_3, i.e. divide them into virtual classes Π_{21}, Π_{23}, Π_{31} and Π_{32}. Then the following equations are assumed:

$p_{12} : p_{21} = P(\Pi_1 \mid o) : P(\Pi_2 \mid o)$, where p_{ij} (= 1 - p_{ji}) is the the a posteriori probability of class Π_i if there are only classes Π_i and Π_j, *o* is the output of a discriminant function (pair-wise classifier) and $P(\Pi_i \mid o)$ is the a posteriori probability that the vector to be classified belongs to the class Π_i, provided value of discriminant function *o*. But the decision boundary of these two classes is actually between the confusing subsets Π_{12} and Π_{21}. Thus it is better to use the following equation:

$$p_{12} : p_{21} = P(\Pi_{12} \mid r) : P(\Pi_{21} \mid r)$$

Furthermore, let's assume that:

$$p_{12} : p_{21} = P(\Pi_{31} \mid r) : P(\Pi_{32} \mid r)$$

This assumption means that $p_{12} : p_{21}$ also stands for the ratio between the degree of the pattern's resemblance to Π_1 and Π_2. Let's make the same assumption on the ratios $p_{13} : p_{31}$ and $p_{23} : p_{32}$. Then the following equation is obtained:

$$P_{12}:P_{21}:P_{13}:P_{31}:P_{23}:P_{32} = p_{12}p_{13}p_{23} : p_{21}p_{13}p_{23} : p_{12}p_{13}p_{32}$$
$$: p_{12}p_{31}p_{32} : p_{21}p_{31}p_{23} : p_{21}p_{31}p_{32}$$

Here P_{ij} stands for $P(\Pi_{ij}|o)$. Now the ratio of the a posteriori probability of each class is comuputed by:

$$P(\Pi_1|o): P(\Pi_2|o): P(\Pi_3|o) = (P_{12}+P_{13}):(P_{21}+P_{23}):(P_{31}+P_{32}) =$$

$$p_{12}p_{13} : p_{21}p_{23} : p_{31}p_{32}$$

A posteriori probability $P(\Pi_1|o)$ can be computed by normalizing the sum to be 1.

In order to generalize the *Resemblance Model* to the *K*-class case, the authors denote virtual classes Π_t where $t = (t_1, \ldots, t_M)$, $t_i \in \{0,1\}$, $M = \binom{K}{2}$. The *i*-th index t_i is 0 when Π_t is closer to Π_p than to Π_q, otherwise index t_i is equal to 1. Π_p and Π_q are the *i*-th pair out of $\binom{K}{2}$ pairs sorted in lexical order. Then the following equation must hold:

$$p_{ij} : p_{ji} = P(\Pi_{t_1,\ldots,t_{k-1},0,t_{k+1},\ldots,t_M} \mid o) : P(\Pi_{t_1,\ldots,t_{k-1},1,t_{k+1},\ldots,t_M} \mid o) \qquad (17)$$

Here t_k corresponds to the pair of classes Π_i and Π_j. Equation (17) stands for $M2^{M-1}$ condition which can be written without inconsistency in the following equation:

$P(\Pi_t \mid o) = \alpha \prod_{i,j,i<j} l_{ij}$, where $l_{ij}=p_{ij}$ if $t_k = 0$, or $l_{ij}=p_{ji}$ if $t_k = 1$ and α is a normalization factor. Then the equation $P(\Pi_i \mid o) = \dfrac{\prod_{j,j\neq i} p_{ij}}{\sum_l \prod_{j,j\neq l} p_{lj}}$ is computed using the condition $\sum P(\Pi_i \mid o) = 1$. The decision is made upon the *i* with the highest posterior probability $P(\Pi_i \mid o)$.

3.7 Kind of Pair-wise Classifier Output

As it is seen from the reviewed fusion methods, different methods use different output types of pair-wise classifiers. The analysis of classifier fusion is done and some explanations on the use of classifier output type are provided in [44]. When the classifiers are good for a particular feature space, the crisp

output may be used. In this case, the fusion rule may be rather simple (e.g. voting). But if classifiers are weak in the provided classification task, then continuous outputs have to be used (e.g. H-T, WLW). This time all the difficulty of classification problem is transferred to the fusion method. There could be an intermediate case between continuous and crisp outputs of classifiers when using non-linearly transformed variables. E.g. if $o_1=f(y_1)=1/(1+\exp(-y_1))$ and $o_2=f(y_2)=1/(1+\exp(-y_2))$, then outputs $o_j=1/(1+\exp(-\gamma y_j))$ may be considered to be generalized outputs distinguished by different positive γ, which in fact controls linearity of the classifier's output transformation and this way influences the complexity of the fusion rule and affects its generalization error. If γ is small, almost linear transformation resulting in classification in continuous feature space is obtained, while with very large γ the crisp outputs are obtained.

3.8 Consideration for Fusion Methods' Choosing

Three types of classifier fusion methods are mainly used in practice: some kind of voting, probability estimation based methods and methods based on some kind of similarity of outputs. Thus the simple voting and well known and easy to implement DAG methods were chosen as representatives for the first group. The widely known approach presented by Hastie-Tibshirani was chosen as the representative from probability estimation based method group. The Wu, Lin and Weng method was also chosen since the authors declare [7] that it outperforms the Hastie-Tibshirani method. The newly constructed method of pair-wise Fuzzy Templates (see the next section) was chosen as the similarity based method.

The SLP and SVC were chosen for pair-wise classifiers. The first one was chosen doe to its feature to obtain seven different statistical classifiers during its training, thus making it adaptable to different data distributions. The support vectors were chosen because of their popularity during recent years and its mathematical similarity to single layer perceptrons.

3.9 Pair-wise Fuzzy Templates Method

3.9.1 Reasoning for a New Method

Most of the pair-wise classifier fusion methods are based on voting or probability estimation. Trained classifiers with the similar data produce similar outputs, thus the similarity of the output vectors may also be used as classification criteria. In the review of existing methods for pair-wise classifier fusion a method based on the similarity of numeric outcome values of pair-wise classifiers was missing. That's why there was need for a construction of method based on similarity of pair-wise classifier outputs. The multi-class classifier fusion method [13] based on output value similarities was presented in Section 2.3.2. A new method for pair-wise classifier fusion was constructed on the basis of this method.

3.9.2 Description of the Pair-wise Fuzzy Templates Method

Since the original Fuzzy Templates method [13] (see Section 2.3.2) is aimed to fuse continuous outputs of several *K*-category classifiers the adaptation of this algorithm to $L=K(K\text{-}1)/2$ pair-wise classifiers was done. First, the fuzzy template of class Π_i was modified to vector $F_i = \{f_i(l)\}$ with *K*-1 attributes, where

$$f_i(l) = \frac{\sum_{z=1}^{Nk} C_{r,s}(x_z)}{N_k} \tag{18}$$

for all l=1..*K*-1 classifiers $C_{r,s}$ where $r = i$ or $s = i$, $\{\boldsymbol{x}_z, z = 1..N\}$ is crisply labeled training data, N_k is the number of training vectors in class Π_k, $C_{r,s}(\boldsymbol{x}_z)$ is the output of the pair-wise classifier. Only classifiers with class labels

$$(i, j) = (1, 2), (1, 3) \ldots (K\text{-}1, K) \tag{19}$$

are considered, because $C_{j,i}(\boldsymbol{x}_z) = 1 - C_{i,j}(\boldsymbol{x}_z)$. So, there are *K*-1 associated pair-wise classifiers for each pattern class. The features in the Fuzzy Template vector were ordered as described in Equation (19).

Then, having a new vector $\boldsymbol{x}_z$ to be classified, the calculation of its decision profiles (vectors) for each class is done:

$DP_i(\boldsymbol{x}_z) = \{C_{r,s}(\boldsymbol{x}_z)\}$, where $r = i$ or $s = i$, ordered as described in Equation (19).

Final decision making was made according to $\max_i(S(F_i, DP_i(\boldsymbol{x}_z)))$, where

$$S(F_i, DP_i(\boldsymbol{x}_z)) = 1 - \frac{1}{K-1}\sum_{l=1}^{K-1}(f_i(l) - DP_{i,l}(\boldsymbol{x}_z))^2 \qquad (20)$$

When classifying vector *x* between two classes, the output value of pair-wise classifier for that vector depends on position, i.e. whether it is considered as a vector from the first or the second of the two classes. In order to avoid this confusion the direct output of SLP (as well as SVC, see the next subsection) has to be taken when the vector is considered to belong to the first class. And the value 1-"classifier output" has to be taken when vector *x* is considered to belong to the second class.

3.9.3 Weaknesses and Strengths of Pair-wise Fuzzy Templates Method

Since the result of PWFT is directly produced from outputs of pair-wise classifiers, PWFT may work effectively when outputs of pair-wise classifiers are highly diverse. While given a particular vector x from class Π_i the multi-class classifier is expected to produce diverse results for each class, the pair-wise classifiers of different classes may produce the same results thus confusing the fusion rule. Using data sets allowing such situations with this method should be avoided. When using SLP as pair-wise classifier, this situation occurs when all the classes are arranged in similar manner. Thus the

duration of SLP training process is approximately the same for all pairs which results in similar outputs of pair-wise classifiers.

Let's examine two examples of 2-dimensional data sets. In the first one, five data classes having the same covariance matrixes and arranged in a symmetric manner were generated (see Figure 5(a)). The Fisher discriminant function should optimally separate classes and the duration of training for all pair-wise SLP classifiers should be approximately the same in this situation. Five data classes with different covariance matrixes and arranged in random manner (see Figure 5(b)) were generated in the second data set. Different statistical classifiers should be optimal for different pairs in this situation, thus forcing pair-wise SLPs to stop at different moments of learning.

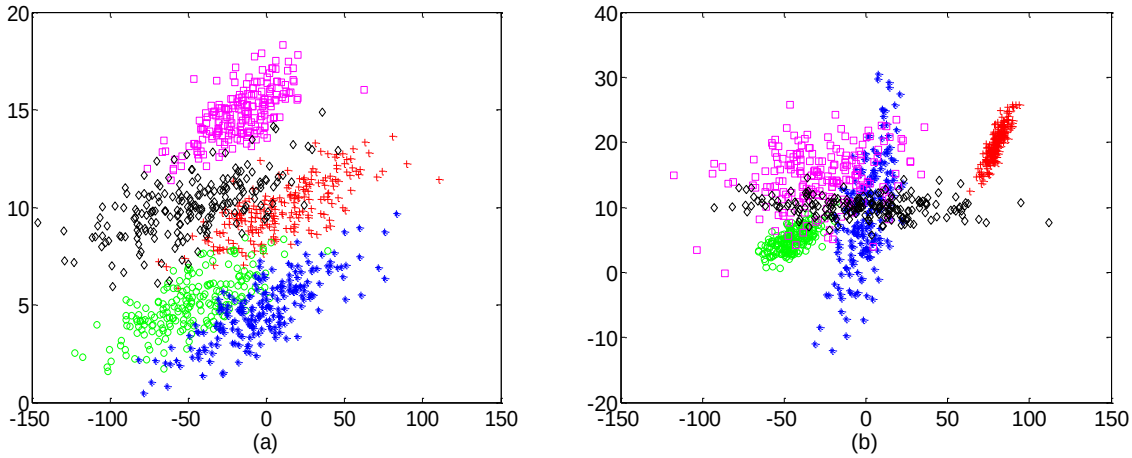


Figure 5. Five two dimensional Gaussian classes a) plotted in a symmetric manner with the same covariance matrices; b) plotted in a random manner with different covariance matrixes.

The results of experiments with these data sets are shown in Table 1. In the first case, when SLP is used as a pair-wise classifier, PWFT method performs very badly compared to other fusion rules. This is because all the pair-wise classifiers result in almost the same values. The overall average value of fuzzy template vectors' attributes was 0.6519, i.e. closer to1/2 than to 1 which means that the training of SLPs had been stopped approximately at the moment they evolved to the Fisher discriminant function. The overall

minimum and maximum values of fuzzy template vectors were 0.5603 and 0.7347 accordingly, and both of them belonged to fuzzy template vector of the class marked by black diamonds (the class that intersects with nearby classes the most). The situation was much better when SVC with modified outputs was used as a pair-wise classifier instead of SLP. Despite the average value of such fuzzy templates vectors' attributes being 0.8815 and the minimum and maximum values 0.8034 and 0.9456 accordingly, the fuzzy template vectors' attribute value interval narrowness did not affect the decision of SVC. This is because the proposed modification using sigmoid function (7) for widely scattered SVC output values was applied.

Data	Classifier	Pair-wise FT	H-T	Voting	DAG
Data (a)	SLP	0.294	**0.115**	0.121	0.121
	SVC	0.119	**0.117**	0.118	0.118
Data (b)	SLP	**0.193**	0.210	0.257	0.266
	SVC	**0.175**	0.194	0.237	0.230

Table 1. Results of two generated data sets with various fusion methods (best of them marked as bold)

The data set (b) provides a much more favorable situation for PWFT method with SLPs as pair-wise classifiers (PWFT+SLP). The values of pair-wise SLPs outputs were scattered all over the interval between 1/2 and 1. The average of the fuzzy template vector attributes' value was 0.7486 (the mean value between 1/2 and 1), while the minimum and maximum values were 0.5020 and 0.9520 accordingly. Thus different pair-wise classifiers were obtained during pair-wise SLP training processes which resulted in PWFT outperforming other fusion rules. This time PWFT with modified output SVC as pair-wise classifier (PWFT+SVC) also showed a considerable improvement.

Results on generated data show that despite its adaptability to data, SLP as a pair-wise classifier is outperformed by the pair-wise SVC with modified output. However, this is not a rule – everything depends on data. It should be

pointed out that when overtrained, SLP is much more sensitive to dimensionality than SVC. But in any case, the number of classes, dimensionality, and other data characteristics may have significant impact on results of this method, so they should also be considered when applying this method.

Now let's introduce a new parameter α which is used as weight multiplier, i.e. Equation (7) is modified to:

$$f(x) = \frac{1}{1+e^{-\alpha x}} \tag{21}$$

This way the range of input values is stretched, but the range of output values is narrowed. But when employed in forming PWFT+SLP fuzzy template vectors' values, the same effect as in PWFT+SVC case is obtained – despite the narrower range of SLP output values, the results are better due to a wider range of input values. Even in unfavorable cases, PWFT+SLP method performance with proper α is not worse than the performance of standard voting methods. E.g. if the Equation (21) with parameter α=50 is used instead of the Equation (7), the PWFT+SLP method with data set (a) results in error rate of 0.121 which is the same as voting and DAG methods with SLP as a pair-wise classifier. The error rate for PWFT+SLP could be improved further by using different parameters for different pair-wise classifier outputs in fuzzy templates. But this requires a particular procedure to determine the optimal α value for each pair-wise classifier which is out of scope of this study.

3.10 Concluding Remarks

The presumptions for two stage pair-wise classifier based approach over standard multi-class classifiers in multi-class pattern recognition task were discussed in this chapter. Then the analytical overview of pair-wise classifiers and their fusion methods was presented. It was decided to choose SLP and SVC as pair-wise classifiers due to their flexible statistical features and popularity. Two voting strategies (Voting and Directed Acyclic Graph) and

two probability estimating based (Hastie-Tibshirani and Wu, Lin, Weng) fusion methods were chosen for further analysis.

Due to the lack of pair-wise classifier fusion methods based on output similarities, the new method *Pair-wise Fuzzy Templates* method was presented. The theoretical and experimental analysis of its advantages and disadvantages was done. Experiments with real world data and comparison with benchmark methods proved the presumed properties of newly constructed method. This method was also chosen for further comparison.

Chapter 4

On the Issues of Multi-class Classification Task

One faces many challenges while solving real world classification tasks. Small sample size and imbalanced data are only a few but very important of them. The origin of small sample problem is presented in Section 4.1. Section 4.3 it is shows how analyzed pair-wise classifiers deal with it. The theory needed for this analysis is presented in section 4.2.

Section 4.4 deals with imbalance problem solving with pair-wise classifiers, while some attempts to enhance multi-class classifier introducing new modified cost function are presented in its subsection 4.4.1.

4.1 Small Sample Size Problem

The so called small sample size problem occurs when class sample sizes N_k, k=1, …, K, are small compared to the dimensionality of the data d. Then the estimates of covariance matrices used in standard statistical classification techniques (e.g. linear or quadratic discriminant analysis) become highly variable. Especially with $N_k < d$, when not all parameters could be obtained. The effect of this problem on discriminant analysis is clearly presented by J. H. Friedman in [42].

First, let's denote Σ_k as the covariance matrix of the k^{th} data class. Then its spectral decomposition can be written as

$$\Sigma_k = \sum_{i=1}^{d} e_{ik} w_{ik} w_{ik}^T ,$$

where e_{ik} is the ith eigenvalue of covariance matrix Σ_k (ordered in decreasing value) and w_{ik} is the corresponding eigenvector. Then the inverse of matrix Σ_k could be written as

$$\Sigma_k^{-1} = \sum_{i=1}^{d} \frac{w_{ik} w_{ik}^T}{e_{ik}}.$$

So, when class sample sizes N_k are small compared to the dimensionality of the data d, the covariance matrix estimates become highly variable. Besides, not all parameters of covariance matrix are even identifiable when $N_k < d$. The effect this has on discriminant analysis can be seen by spectral decomposition of class covariance matrices:

$$\Sigma_k = \sum_{i=1}^{d} e_{ik} w_{ik} w_{ik}^T .$$

Using such a representation of the covariance matrix, the inverse of it is represented by

$$\Sigma_k^{-1} = \sum_{i=1}^{d} \frac{w_{ik} w_{ik}^T}{e_{ik}} \tag{22}$$

Most of the classical statistical classification rules are based on the normal distribution [3]:

$$f_k(x) = \frac{1}{\sqrt{(2\pi)^2 |\Sigma_k|}} \exp(-1/2 (x - m_k)^T \Sigma_k^{-1} (x - m_k)),$$

where m_k is the mean of data vectors from class k = 1, 2, …, K.

Then the quadratic discriminant analysis (QDA) using following discriminant functions is performed:

$$d_k(x) = (x - m_k)^T \Sigma_k^{-1} (x - m_k) + \ln|\Sigma_k| - 2 \ln p_k \tag{23}$$

where p_k is the prior probability of the k^{th} class. When all Σ_k are the equal the linear discriminant analysis (LDA) is obtained. Now let's replace Σ_k in (23) with Equation (22). The following form of discriminant function is obtained:

$$d_k(x) = \sum_{i=1}^{d} \frac{\left[w_{ik}^T (x - m_k) \right]^2}{e_{ik}} + \sum_{i=1}^{d} \ln e_{ik} - 2 \ln p_k \tag{24}$$

The new expression of discriminant function in Equation (24) shows that its result highly depends on the values of the smallest eigenvalues and their corresponding eigenvectors. When designing the discriminant function, the

estimates of the mean vectors ($\hat{m}_k$) and covariance matrices ($\hat{\Sigma}_k$) are obtained from the sample data provided for training:

$$\hat{m}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i^k \tag{25}$$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} (x_i^k - \hat{m}_k)(x_i^k - \hat{m}_k)^T \tag{26}$$

where x_i^k is i^{th} vector from k^{th} sample class.

Iw we now replace the corresponding values in Equation (24) by Equation (25) and Equation (26) the discriminant function based on sample data would be obtained. Correspondingly such discriminant function depends upon the smallest eigenvalues and their corresponding eigenvectors. But estimates of such eigenvalues are biased. The largest eigenvalues are biased towards high values and the smallest ones are biased toward the values that are too low. The biasing has worse impact when estimation of eigenvalues obtained from sample data are approximately eaqual, while highly diverse values of eigenvalues estimation results in less severe impact of biasing. Anyway the less the sample data size, the worse impact it has in the sense of biasing. When sample size N_k becomes less or equal to dimensionality d, the d-N_k+1 smallest eigenvalues are estimated to be zero. So the impact of biasing results on discriminant analysis results in excessive importance of eigenvectors corresponding to the eigenvectors having the smallest estimates from the sample data.

There are some technical approaches to overcome this situation [42]. One of them is to try to obtaining more reliable estimates of the eigenvalues by correcting eigenvalue distortion in the sample covariance matrix (e.g. [45], [46]). Another approach is to employ regularization method (e.g. [47], [48]).

On the other side if we look from a wider point of view (i.e. before applying discriminant analysis), there are two main causes of the small data set size problems: 1) too high dimensionality; 2) too small number of training data

size. As there are two causes of problems, there are two obvious solutions – either enlarge training data set or reduce dimensionality. There are two types of methods to reduce dimensionality: a) feature reduction when some methods are applied to reject some not informative or redundant features, and b) feature extraction when some method is applied in order to make new informative and less dimensional features.

It should also be mentioned that quadratic discriminant analysis (QDA) based classification rules are known to require generally larger samples than te ones based on LDA [43].

4.2 Generalization Error of the Fisher Classifier

In the two-stage approach, the possibility to choose the feature subset and the type of classifier that is individual to each pair of the classes is acquired. In this context, the perceptron's quality that *while training the nonlinear SLP, the classifier's complexity is gradually increasing* becomes very attractive. A theoretical justification of this opinion is presented by analyzing a hypothetical situation, where the classes are multivariate Gaussian distribution and share the common covariance matrix (GCCM data model). For the GCCM data model the standard linear Fisher DF is an asymptotically (when $N_1 \to \infty$, $N_2 \to \infty$) optimal decision rule. This classifier is considered because the SLP based classifier behaves like the Fisher DF during one of the phase of its evolution,. The generalization error of binary Fisher DF may be calculated by following asymptotic formula [3], [27]

$$EP_N^{\mathrm{F}} \approx q_i \Phi\left\{-\frac{1}{2}\delta_{ij} T_{\mathbf{m}_{i/ij}} T_{\boldsymbol{\Sigma}}\right\} + q_j \Phi\left\{-\frac{1}{2}\delta_{ij} T_{\mathbf{m}_{j/ij}} T_{\boldsymbol{\Sigma}}\right\} \tag{27}$$

where $\Phi\{a\} = \int_{-\infty}^{a} (2\pi)^{-1/2} \exp(-t^2/(2\sigma^2))dt$ is the standard Gaussian cumulative distribution function,

$$
\begin{aligned}
T_{\mathbf{m}_{i/ij}} &= \frac{1 - d/(N_i \delta_{ij}^2) + d/(N_j \delta_{ij}^2)}{\sqrt{1 + d/(N_i \delta_{ij}^2) + d/(N_j \delta_{ij}^2)}} \\
T_{\mathbf{m}_{j/ij}} &= \frac{1 + d/(N_i \delta_{ij}^2) - d/(N_j \delta_{ij}^2)}{\sqrt{1 + d/(N_i \delta_{ij}^2) + d/(N_j \delta_{ij}^2)}} \\
T_{\bar{\mathbf{\Sigma}}} &= \sqrt{\frac{N_i + N_j - d}{N_i + N_j}}
\end{aligned}
\tag{28}
$$

where δ_{ij} is a Mahalanobis distance between pattern classes Π_i and Π_j

$$\delta_{ij} = ((\Delta_{ij})^T \Sigma^{-1} \Delta_{ij})^{1/2} \tag{29}$$

$\Delta_{ij} = m_i - m_j$, m_i and m_j are mean vectors of the classes Π_i and Π_j and Σ is their "common" covariance matrix with elements calculated by obtaining the arithmetical average of all the elements same positions in the covariance matrices of all classes..

Terms $T_{mi/ij}$ and $T_{mj/ij}$ in Equation (27) emerge due to inexact sample estimation of the mean vectors of the classes. Term $T_{\mathbf{\Sigma}}$ appears due to inexact sample estimation of the covariance matrix that is supposed to be common to Π_i and Π_j. The covariance matrix is not estimated in EDC design. Hence for EDC the term $T_{\mathbf{\Sigma}}$ in Equation (27) has to be omitted. In the latter case, the standard Mahalanobis distance (29) no longer determines the asymptotic probability of misclassification. An "effective distance" and "effective dimensionality" ([3], Chapter 3) are obtained instead:

$$
\begin{aligned}
\delta^*_{ij} &= (\Delta_{ij})^T \Delta_{ij} ((\Delta_{ij})^T \Sigma \Delta_{ij})^{-1/2} \\
d^*_{ij} &= ((\Delta_{ij})^T \Delta_{ij})^2 \, \mathrm{tr}\Sigma^2 ((\Delta_{ij})^T \Delta_{ij})^{-1/2}
\end{aligned}
\tag{30}
$$

The Mahalanobis distance, Δ_{ij} and effective parameters δ^*_{ij}, d^*_{ij} are specific to each pair of the classes. Usually $\delta^*_{ij} < \delta_{ij}$, and $1 < d^*_{ij} < \infty$ [3].

4.3 Small Sample Size Solution

If sample sizes $N_2 = N_1 = N$ and $q_2 = q_1 = 0.5$, expressions (28) become more simple. In the following illustration, three-category 30D Gaussian data were generated. Mean vectors and covariance matrices of the first 17 features were taken from the realty data (see Section 5.3.1). 13 non-informative features with low dispersions, $\sigma^2 = 0.1^2$ (standard deviations of the first two features were approximately 1) were added. Due to particularities of the real-life realty data used to construct the artificial data, three classes are located approximately on an arched curve (see Figure 3). Thus the asymptotic classification errors of the Euclidean distance and Fisher classifiers, P_∞^{E} and P_∞^{F} are almost equal for the class pairs Π_1, Π_2 or Π_2, Π_3. To increase the difference between asymptotic errors of EDC and Fisher classifiers, covariance matrices $\Sigma_{1\mathrm{new}} = \Sigma_1/\theta$, $\Sigma_{2\mathrm{new}} = \Sigma_2 \times \theta$ were diversified in novel experiment, $\Sigma = ½(\Sigma_{1\mathrm{new}} + \Sigma_{2\mathrm{new}})$ were used in calculations. Table 2 shows parameters δ_{ij}, δ^*_{ij}, d^*_{ij} and P_∞^{F}, P_∞^{E}.

Experiment	δ_{IJ}	δ^*_{ij}	P_∞^{F}	P_∞^{E}	d^*_{ij}
Figure 6, $\theta = 0.7$	3.646	3.612	*0.0342*	*0.03557*	1.90
Figure 7, $\theta = 0.4$	3.234	3.030	*0.0530*	*0.06492*	1.84

Table 2. Parameters of Three Class 30-dimensional Gaussian data.

The effective dimensionality was $d^*_{12} = 1.90$ in the computational example. It is much smaller than d=30, the formal dimensionality of the data. This fact advocates that the sensitivity of EDC to the training set size is low in this particular case. In small learning set situations, however, generalization error rate of Fisher DF is much higher than the asymptotical one (curves 1 and 2 (black dashes and green dots) in Figure 6).

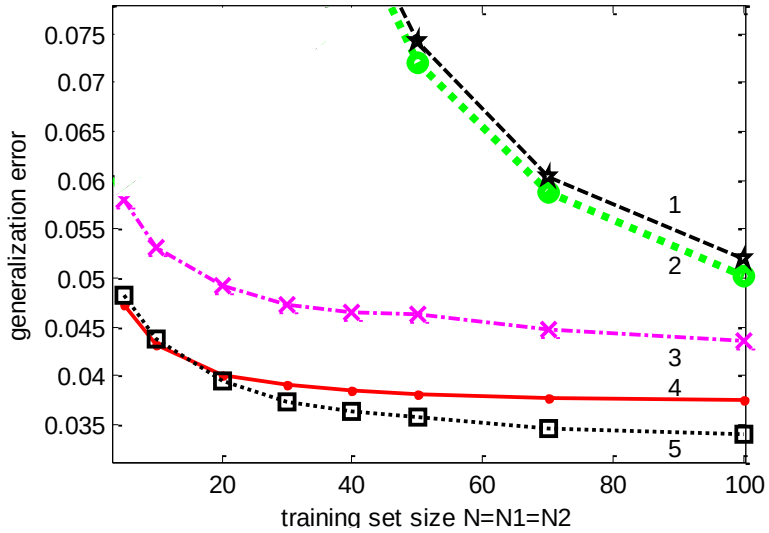


Figure 6. Generalization errors as functions of sample size: 1 – Fisher DF (experiment), 2 – Fisher DF (theory, Equations (27), (28)), 3 – SV classifier, 4 - EDC classifier, 5 – pseudo-validation set stopped SLP.

Generalization errors (evaluated experimentally in 1,000 runs) of EDC, the pseudo-validation set stopped SLP (black squares) and SV classifier (magenta, crosses) with parameter C evaluated using generalization errors estimated from pseudo-validation set are also presented in Figure 6.

This data model is unfavorable for SV classifier. Exploitation of the default C parameter (C = 1) resulted in notably smaller generalization errors that were very close to that of EDC (compare curves 3 and 4). The pseudo-validation set stopped SLP was the best classifier.

4.4 The Unbalanced Sample Size

When prior probabilities of the classes are different and the training set sizes are imbalanced, terms $d/(N_i\delta_{ij}^2)$, $d/(N_j\delta_{ij}^2)$ in Equation (28) and nonlinearity of cumulative distribution function $\Phi\{a\}$ in Equation (27) cause that EDC and Fisher classifiers to be not optimal. The non-optimality is inherent to the SV and SLP based classifiers, too.

While analyzing the imbalance problem in the multi-category situation (see the next Subsection 4.4.1 for details), it is seen that the generalization error of the Fisher DF diminishes permanently with an increase in the training set size of one of the classes. But now let us now consider *a two category PR task*, where the true learning set sizes N_1, N_2 do not reflect q_1 and q_2. Let's assume that $q_2 = 0.75$ and N_2 is varying. The total learning set size remains unchanged: $n = N_1+N_2=100$.

The generalization errors for the pair classes Π_1, Π_2 ($\boldsymbol{\Sigma}_2 \neq \boldsymbol{\Sigma}_1$ here) as functions of N_2 for EDC, Fisher DF, SV, and the modified SLP based classifiers (see Equation (31) in the next Subsection 4.4.1) are presented in Figure 7. Averages of 1,000 experimental runs performed with the data model considered in previous experiment are plotted. This time, the covariance matrices diversity parameter $\theta = 0.4$ (see. Table 2). The optimally stopped SLP based classifier was almost insensitive to imbalance of N_1, N_2. It was the best choice for classification task in all cases. When ratios of N_i to total sample size n were used instead of *a priori* probabilities, classification results were worse.

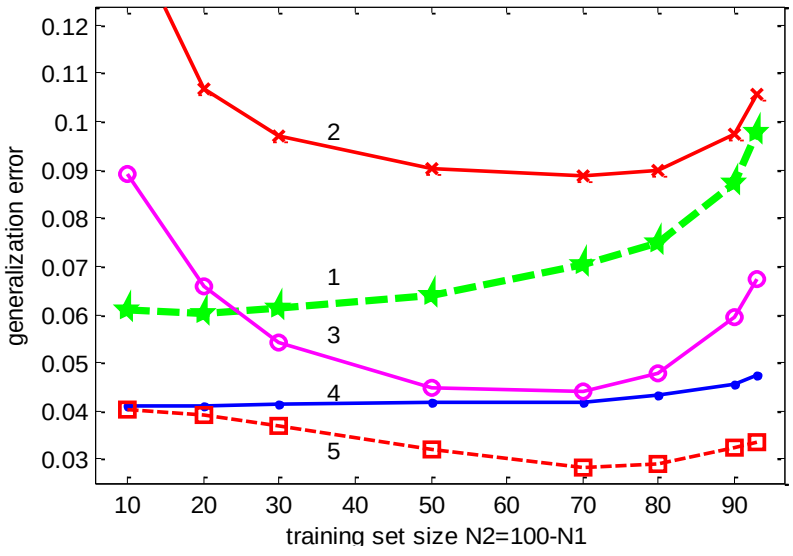


Figure 7. Generalization errors as functions of sample size: 1 – Fisher (experiment), 2 – Fisher DF (theory, Equations (27), (28)), 3 – SVC, 4 - Euclidean distance classifier, 5 – pseudo-validation set stopped novel SLP.

4.4.1 Enhancement of Single-stage K-category SLP-based Neural Net for Classification

The non-optimality of the multi-category neural network in the cases of sufficiently large learning set sizes was witnessed in Section 3.1. Additional difficulties arise if design sample sizes are small and imbalanced [4]. The generalization error is related to the classifier's complexity and the size of the training set $n = N_1 + N_2 + \ldots + N_K$ in the asymptotic learning theory [26], [27]. The use of a solitary sample size parameter n is appropriate if the optimal sample based classification rules are used. Nevertheless, *classification rules without theoretical proof of their optimality are often applied in practice*. It has been shown analytically that the expected generalization errors of particular plug-in sample-based classification rules (the quadratic classifier designed for multidimensional Gaussian classes, and a multinomial classifier designed for discrete valued features) may start rising with an increase in the learning set size N_2 while keeping N_1 constant [27], [30]. No theoretical results exist for multi-category situations.

Experimentally evaluated relationships between the sample size N_3 and the generalization error of the Fisher linear DF (F) and the 3-category network of SLPs in a case where $N_1 = N_2$ is presented as an illustration in Figure 8. The 3-category Gaussian classes, considered in Section 4.3, were used. But this time instead of 13 additional features, 33 of them were added in order to get 50-dimensional data. The curves are averages of 1,000 runs of the experiment where $N_1 = N_2 = 50$, and test set sizes $n_1 = n_2 = n_3 = n_{\text{test}} = 2000$ (prior class probabilities $q_i = 1/3$). If N_3 is small ($N_3 < 50$), the total learning set size n is insufficient to estimate the 50×50-dimensional *covariance matrix* reliably. So the generalization error of the Fisher classifier is high. With an increase in N_3, the total learning set size is increasing. For that reason, the generalization error declines permanently. For very small values of N_3 a similar behavior of the net of SLPs was also observed. The red solid curve 1 marks the result obtained for the optimal number of training epochs (in this experiment $n_{\text{test}} = 2000$; that is

the generalization errors were evaluated rather exactly), and curve 2 marks the result after optimal stopping determined by the artificial pseudo-validation set. When N_3 approaches N_1=N_2=50, the generalization error of the network diminishes down to its minimum. Later, with a further increase in the sample size, N_3, the generalization error starts increasing. This fact confirms that classification error of *non-optimally designed classification rule* depends also on the balance of the data sizes. Curves 1 and 2 demonstrate that generalization error increases more than twice if $N_3 \rightarrow 1{,}000$.

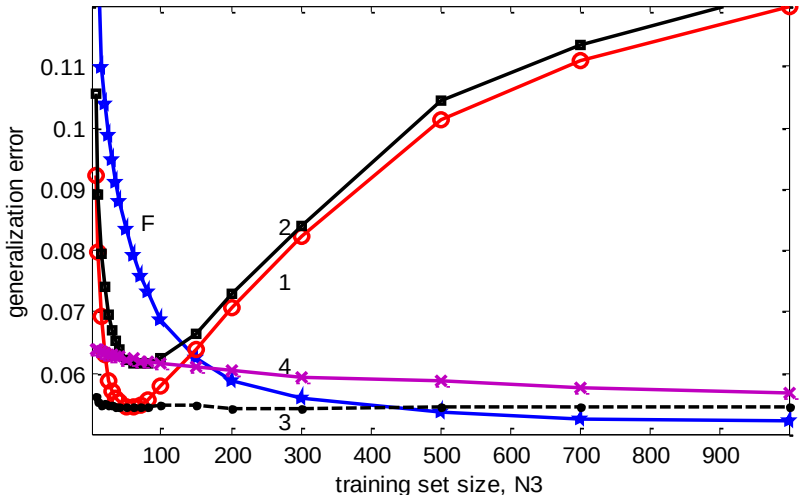


Figure 8. Generalization error as a function of sample size N3 for Fisher DF, standard and modified versions of the SLP-based three category classifiers: F (blue, pentagram) – Fisher DF, 1(red circle) – ideally stopped original KSLP, 2 (black, solid) – original KSLP when stopping was based on pseudo-validation set data, 3 (black x-mark) – ideally stopped modified KSLP, 4 (magenta stars)– modified KSLP when stopping was based on pseudo-validation set data.

A question arises: can this undesirable effect of the standard *K*-category network be avoided? To find an answer attention to the cost function (5) has to be paid. Theoretically, while increasing N_3, the aim is to evaluate the third class influence on the cost function more precisely. If N_3 is increased twice, the contribution of the third class training vectors increases twice. On the other hand, *the balance* between the training set sizes (actually, it is an indirect estimate of prior probabilities of the classes) *becomes corrupted*. In order to

improve the cost function, prior probabilities of each class have to be restored. Hence *the unbalance correcting terms*, q_i/N_i are introduced in [56]. Instead of cost (5), a modified variant is used:

$$\cos t = \sum_{h=1}^{K} \sum_{i=1}^{K} \frac{q_i}{N_i} \sum_{j=1}^{N_l} [t_{hij} - f(\boldsymbol{x}_{ij}^T \boldsymbol{v}_h + w_{h0})]^2 \tag{31}$$

The modified cost is important when values q_i, *genuine prior probabilities of the classes*, are not proportional to N_i. The lower curves, 3 (ideal stopping, based on generalization errors estimated from the test set) and 4 (pseudo-validation set based stopping) in Figure 8, show the generalization error of "the modified network of *K*-category SLPs". The generalization error curves almost stop increasing with an excessive increase in N_3. Moreover, if training process is stopped at the right time, a notable improvement is achieved even if N_3 is very small.

4.5 Concluding Remarks

The two serious issues – small sample size and class imbalance – which are usually faced by the multi-class classifier designer were addressed in this chapter. Analytical experiments showed that single layer perceptron performs with optimal generalization error rate compared to standard statistical classifiers and support vector machine classifier. Therefore, SLP may absorb the addressed issues when used in two stage classification strategy.

Chapter 5

Experimental Issues and Results of Classifier Comparison

Apart from the issues coming with statistical origins, other – methodical issues are very important. Critical methodical issues on classifier comparison and reliable performance estimation are often omitted. Thus the conclusions of research sometimes are not reliable. This chapter addresses mainly two methodical issues: a) the effect of not exact criteria use in classifier optimization and b) too small number of experiments performed. These issues are discussed in Section 5.1 and Section 5.2 correspondingly.

The results of classifier comparison are presented in this chapter as well. The real world data from various sources was used for the experiments. The explanation of the procedure of the experiments, data description, results and their analysis are presented in Section 5.3.

Section 5.4 addresses biasing of trainable classifier fusion methods. The use of new pseudo-validation data next to analytical correction of results is proposed.

5.1 The Effect of Simplified Performance Measures and Sample Size on Fusion Accuracy of the Pair-wise Classifiers

The sum of K-L distances (10) between the estimates $\hat{r}_{ij}$ and true probabilities μ_{ij} is minimized in the H-T fusion method. In the WLW method, the sum of squared differences of pair-wise conditional probabilities (14) is employed. The Fuzzy Template method uses another sum of squared differences (20). Two sources of errors affect the accuracy of the performance

estimates:

(a) the simplified performance measures are only approximately related to classification error,

(b) the classifiers are based on the training data, while the test data was used to estimate the generalization error.

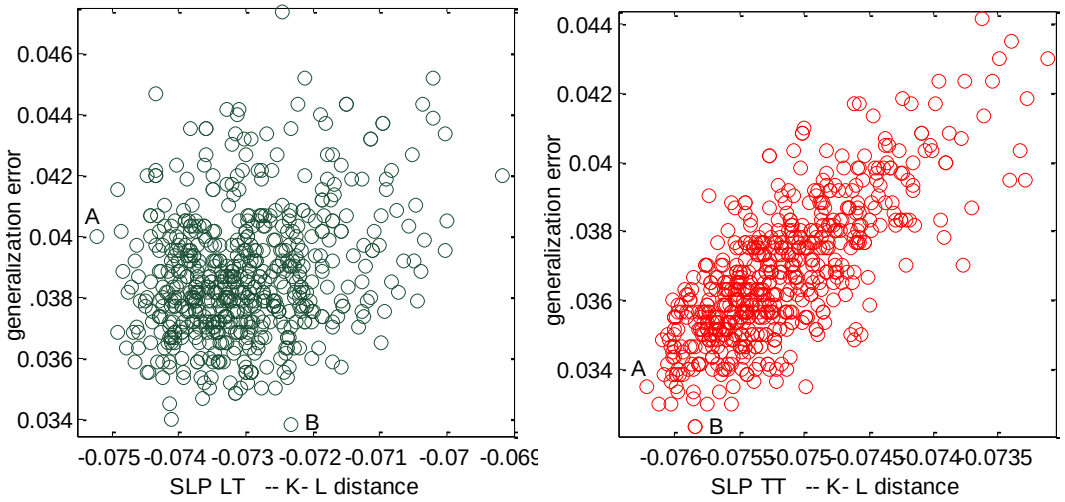
The complexities of the support vectors and single layer perceptrons used as the linear pair-wise classifiers in this thesis are governed by a) regularization parameter *C* or 2) the number of training epochs. The fact that the complexity of a classifier gradually increases during its training process is positive feature of the SLPs is [3], [17]. This way, a SLP may adapt its complexity to a particular design of the data set. Artificial pseudo-validation data sets generated by means of noise injection were used to find the best values for the parameters mentioned.

The difference between the two criteria – the K-L distance and classification error rate – affects the accuracy of the fusion rule. In dealing with effects of this diversity in finite sample size conditions, the complexity of the PR task and effects of high dimensions have to be taken into account. Hence, three-category 50D Gaussian data was generated. Mean vectors and covariance matrices of the first 17 features were taken from the realty data. In order to get higher dimensions, 33 non-informative features with low dispersions, $\sigma^2 = 0.1^2$ (standard deviations of the first two features were approximately 1), were added.

Due to the non-linearity of activation function, exact non-asymptotic analysis of learning dynamics is impossible. For that reason, *a random search optimization procedure* was investigated. Relatively small learning sets ($N_3 = N_2 = N_1 = N = 50$) were used. In 600 independent runs of the experiment, 600 random learning sets $S^1_{Lpw}, S^2_{Lpw}, \ldots, S^{600}_{Lpw}$ of size $3\times N$ were generated and used to train 600 triplets of pair-wise SLPs. 600 3×51-dimensional matrices of the weights were obtained as a result. A large test set S_{Test} ($n_3 = n_2 = n_1 = 2000$) was

used to achieve accurate estimates of the classification error rates.

To find the parameters of 600 H-T fusion rules in the *first experiment*, an extra learning set, $S_{LFusion}$ (50×3 vectors) was generated. This set was used to estimate sample means $\hat{m}^{R}_{i/ij}$, $\hat{m}^{R}_{j/ij}$ and standard deviations $\hat{s}_{i/ij}$, $\hat{s}_{j/ij}$. 600 K-L distances D^{t}_{KL} were estimated from set $S_{LFusion}$, every time. A single test set S_{Test} was used to estimate generalization errors P^{t}_{gen} (t = 1, 2, …, 600). In **Figure 9(a), a scatter diagram of the distribution of 600 vectors ($-D^{t}_{KL}$, P^{t}_{gen}) (t = 1, 2, …, 600) is shown. This diagram was obtained after minimizing the K-L** distances evaluated from the learning set.



(a) (b)

Figure 9. Generalization errors of 3-category pair-wise classifier fusion by H-T method as functions of averaged K–L distances based on learning (a) and test (b) sets.

Each triplet corresponds to one of the 600 points in 2D scatter diagram in Figure 9(a). Here the estimation of K-L distances was based on the learning set and the generalization errors were evaluated from the test set. Experiments show that the generalization errors and K-L distances are weakly correlated (ρ = 0.263). The low correlation, observed in Figure 9(a), is caused by factors (a) and (b) that were mentioned at the beginning of this section.

In order to elucidate the effect of factor (a), the influence of factor (b) has to be eliminated. In order to get rid of inexact estimation due to finiteness of

the sample size the *second artificial experiment* was carried out: the test set vectors were used to estimate the means $m_{i/ij}$, $m_{j/ij}$ and standard deviations $s_{i/ij}$, $s_{j/ij}$ (Figure 9(b)). The generalization errors and K-L distances are considerably more correlated (ρ=0.757) in this case. This fact hints that low correlation is evoked due to inexact parameter estimation from small samples. This experiment confirms that the influence of finiteness of the sample size is a more essential factor than the influence of differences between the generalization errors and K-L distances in the 50-dimensional PR task with relatively small sizes of the learning set (N = 50).

The effect of the sample size finiteness may be proved more forcefully after a scrupulous analysis of the scatter diagrams in Figures 9. The best model according to approximate measures, the K-L distances, is selected in realistic model selection procedures. Figure 9(a) shows that, the classifier **A** has to be chosen. The generalization error of model **A** is: P_{genA}= 0.04. A hypothetical procedure, an *ideal model selection*, is performed according to the test set estimates of the generalization error. In ideal model selection, the best model (classifier) is **B** with P_{genB} = 0.034. Thus, the generalization error increased by ΔP_{genL}= 0.04 – 0.034 = 0.006 because of the usage of imperfect model selection measure. Similar analysis of Figure 9(b) shows that this time (with the influence of the of sample size finiteness artificially eliminated), the generalization error increases from P_{genB} = 0.0324 (*ideal classification error in model selection*) up to P_{genA} = 0.0335 (*true classification error in model selection*). This time, the difference ΔP_{genT} =P_{genA} – P_{genB}= 0.0011 is much smaller. The comparison of ΔP_{genL} and ΔP_{genT} confirms that the finiteness of the sample size was the main factor in this particular simulation study.

The effect of K-L criteria inexactness is arising together with finiteness of the sample size. In hypothetical situations with very large sample sizes, the inexactness of performance measure would be the only source of errors. With an increase in sample size, the difference P_{genA} - P_{genB} declines. This difference is approaching a certain constant that depends on the accuracy of performance measure. The rapidity of the decrease depends on the data and the accuracy of

model selection criterion. But the point of such type of research is left outside the scope of this thesis. Such investigation has been done analytically for the cross-validation error counting classification error estimate (see [3], Section 6.5.2). The joint impact of the sample size and the inexactness of the model selection criteria is an important unexplored problem.

In order to gain more reliable, *"averaged" evaluations of the influence of the both factors*, all possible selections of l models out of M = 600 are considered. If the case of two models (l = 2) selected out of M = 600 ones, $r = M!/(l!\times(M-l)!) = 179700$ selections may be formed. In case of l = 10, $r \approx 1.5453\times10^{21}$. The averages of true generalization errors on the number l (triplets of the pair-wise SLP classifiers) are presented in Figure 10. The fusion rules were based on learning set S_{Fusion} data (1 – SV classifiers were used, red solid line marked by squares; the default value, C = 1, was used this time), 3 – SLPs). Curves 2 (SV classifiers, red dotted line marked by circles) and 4 (SLPs, black dotted line marked by crosses) show the idealized situation when the test set was used to design the fusion rules. Each single point of the curve is an average of r estimations calculated according to V. Pikelis combinatory equations ([3], Appendix A4).

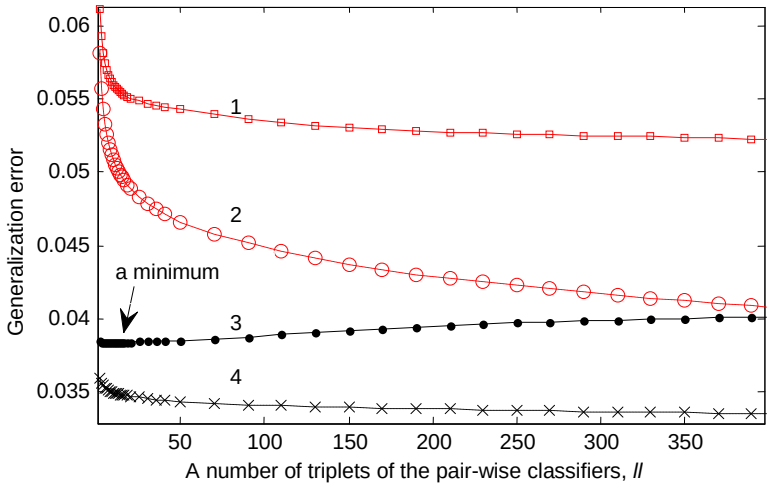


Figure 10. Generalization errors as functions of m, the number of K-category classifiers considered while mimicking training process as the model selection: fusion based on the learning set (1 – SV, 2 - SLP) and the test set (3 – SV, 4 – SLP).

The differences between the "true" (upper red) and "ideal" (lower black) curves characterize an increase in the generalization error when the fusion rules are based on inaccurate estimates of the classification error rate. The differences between points A and B, obtained from data as depicted in Figure 9(a), were also much higher than those of the data in Figure 9(b). These facts corroborate once again that the finiteness of training set size was the main factor that influenced the accuracy of model selection (training of the fusion rule) and increased the generalization error for K-L criterion, the data dimensionality and the learning sample sizes investigated.

The generalization errors obtained by using fusion rules based on the K-L distances (H-T method) and the WLW sums were in fact the same. Figure 11 plots the generalization errors obtained from the WLW sum of squares (*x* axis) versus that from K-L distances (*y* axis).

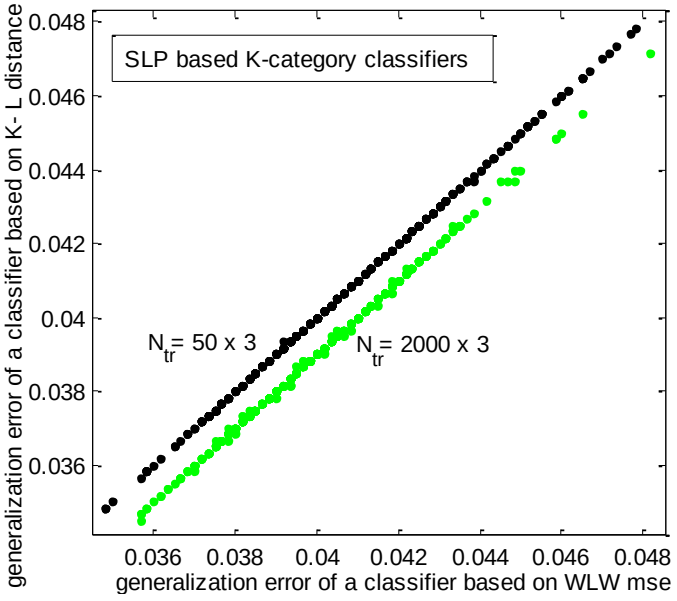


Figure 11. Comparison of fusion rules based on the K-L distances and WLW mean square errors. A set of points marked Ntr=50x3 (left plot, in black) are the results for fusion rules trained on the training set, and Ntr=2000x3 points (right plot, in green) are the results when for fusion rules designed by using the test set data. The plots are undistinguishable. To avoid overlapping, the green plot was shifted by 0.001 to the righ.

The fusion rules were designed either on the training sets (black points scattered on the upper line) or the test sets (green points scattered on the lower line, *green points were shifted by* 0.001 *to the right for display clarity*). This conclusion agrees with the results reported in Table 3 (see Section 5.3.4) where the fact that the H-T and WLW methods gave almost the same accuracy was also observed. The conclusions that followed from Figure 9 and Figure 11 are also valid for WLW estimates.

The curves $P_{gen} = f(l)$ often "peak" with an increase in the number of models, m. T*he overfitting phenomenon* in the experiments where SLPs were used as the pair-wise classifiers was observed in Figure 10. Peaking behavior is widespread in tasks where optimization is based on inexact criteria. An origin of this behavior is the same as in feature over-selection [25] or overtraining of neural networks, where the training process is based on minimization on the learning set based cost function (see [3], Sections 4.5 and 6.5). The inexactness of the K-L distance and WLW sum measures, the difference of the cost function (5) and empirical classification error actually are the causes of the correspondence of effectiveness of H-T and WLW methods, and the peaking phenomena.

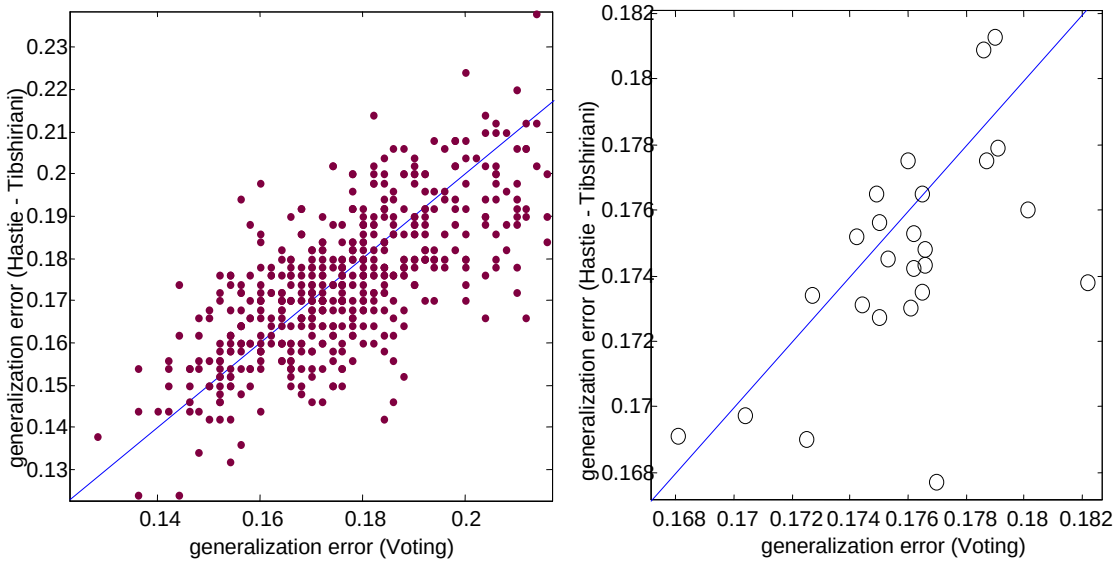
5.2 The Importance of the Number of Experiments

As it was shown in the previous chapters, a plenty of multi-class classification, classifier fusion and other approaches of classification was proposed. Some methods are difficult because of their non-linear expressions while others are rather simple but based only on empirical basis. So in some cases it is impossible to theoretically demonstrate the classification method accuracy. Therefore, the accuracies usually are investigated by means of simulation. The differences between classification methods detected in many experimental investigations are small. Very often the scale of the experiments are insufficient. Some researchers pay main attention to a number of data sets

investigated, forgetting however about the reliability of their experimental evaluations (e.g. [7]). In some cases, only a single leave-one-out or cross-validation experiment is performed with each data. This is especially a case in researches where classification methods are applied to particular data in order to solve some a particular task [36], [37]. Some of researchers probably do not bother mentioning the way they performed their experiments – only the amounts of training and testing data are mentioned, while the most important, i.e. the number of the experiments, is missing. This makes the reliability of results very doubtful. In better cases, the cross-validation experiments are performed 10, 20 or more times after reshuffling each class data every time.

The random splits of the data into training and test sets give diverse results, especially when the size of design set is not large. For that reason the estimates from just one or a few experiments become unreliable. The diverse publications on the comparison of methods often reveal certain contradictions in the results of experimental comparisons of competing algorithms. The differences between performance evaluations of diverse methods vary with the data. As a result, it is impossible for the end users to resolve which method to use in their practical tasks. This imperfection is explained by an example below.

Two scatter diagrams of cross-validation error estimates of two-stage classifiers obtained by the Hastie-Tibshiriani (H-T) [6], and voting methods with six category Satimage data [23] are presented in Figure 12. 500 independent cross-validation estimates are shown in diagram (a) and 25 averaged estimates from 20 subsequent reshuffling in diagram (b). To reduce the computer time necessary for 500 repetitions of the experiments only four features (17 to 20, following the recommendations of the data providers) were used. The experiments were performed the same way as in the Wu, Lin and Weng paper [7]: 300 randomly selected vectors were used for training and 500 vectors were used for testing after multiple reshuffling of each class data.



(a) (b)

Figure 12. Scatter diagrams of cross-validation error estimates in: a) –500 cross-validation estimates, b) after 25 averaging of 20 subsequent trials.

The left scatter diagram (a) shows that both fusion methods in a two stage *K*-category pattern recognition task visually look equivalent. The same result may be seen after calculating the means of the error rates of both classification methods: the voting results in an error rate of 0.176, while the H-T method in an error rate of 0.175. After averaging the same results for 20 subsequent experiments, 25 dissimilar averages are obtained. One of the averages (the point on the blue line in diagram (b)) says that the performance of both methods is similar. Two averages (the ones closest to the bottom right corner) advocate that the H-T method is more successful. However, the majority of the averages (14 points closer to the blue line in the bottom part of the triangle in diagram (b)) confirm that the H-T method is slightly better. Nevertheless, eight *averages* advocate the opposite – that the voting outperforms the H-T method. It means that the comparison of the two methods according to 20 independent experiments may become unreliable in this situation.

The shortcoming of performing only a few experiments may also be seen in the of inaccuracy estimation expression [3]

$$inaccuracy = \sqrt{P(1-P)/N_{ts}} \quad (32)$$

Equation (32) shows the less is the number of testing vectors, the higher the inaccuracy. But only testing vectors from independent experiments are considered here. While performing experiments on the same reshuffled data, the independence of experiments disappears. So the actual inaccuracy is between $\sqrt{P(1-P)/(N_{ts}N_e)}$ and $\sqrt{P(1-P)/N_{ts}}$ where N_e is the number of performed experiments. A more exact estimation requires thorough analysis of the data participating in experiments.

Some data set designers prepare their data by providing two subsets: the training data subset and the testing data subset (e.g. [23] Satimage, Blood data sets). This should not be used as a rule, since by using the data in this way the researchers obtain performance estimation of their algorithms only on the particular test data, not the overall data. Such procedure is good only when training data is selected randomly, while testing data is selected to represent the statistical characteristics of the overall data. Usually the data set providers do not provide any arguments on dividing data into training and testing subsets.

5.3 Experimental Comparison of Fusion Rules

5.3.1 Data

Eight real world data sets were used for the comparison of the methods considered.

The Chromosomes data set is based on 30 geometrical measurements and describes 24 classes of chromosomes. Each class contains 500 data vectors.

The Iris data set [23] is probably the best known data set in pattern recognition world. It was presented by Fisher who is one of pioneers in the

field of statistical pattern recognition. The data describes three types of Iris plant (Setosa, Versicolour and Virginica) by their sepal and petal length and width measured in centimeters. I.e. the data contains three pattern classes with 50 4-dimensional vectors in each class.

The Realty data comprises of 392 17-dimensional vectors describing constructional, ecological, and market characteristics of realty. The data was grouped into three categories (118, 160 and 94 vectors) by experts and was used in an econometric analysis in a private Lithuanian company.

The Satimage data set [23] describes multi-spectral values of the pixels in a satellite image. The values of attributes are between 0 and 255. The six classes representing red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble and very damp grey soil contain 1072, 479, 961, 415, 470, and 1038 36-dimensional vectors respectively. The authors of the data set describe it as data containing seven classes of different scenes, but there is actually no data representing the mixture (all types present) class.

The Wheat data set was obtained by scanning eighty kernels of five wheat varieties. Then digital color images were then transformed to gray level digital images. The converting and segmentation algorithms were applied to get nine geometrical features and three color information features for each kernel (see [24] for more). Thus this data set contains five pattern classes describing different types of wheat with 80 12-dimensional vectors in each of them.

The Yeast data set describes ten types of yeast infections. The classes contain 113, 84, 116, 83, 120, 56, 90, 97, 113, and 129 vectors respectively. The data was composed of 1500 spectral features originally. In order to reduce the dimensionality, a ten-class Euclidean distance classifier was employed. The 10D space of ten EDC outputs formed nine discriminative features.

The Wine data [23] is the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents such as alcohol, flavanoids, color intencity, ash, alkalinity of ash

etc. found in each of the three types of wines. Wine data consists of three data classes containing correspondingly 59, 71 and 48 13-dimensional instances.

The Ecoli data was taken from the UCI repository. It is created and maintained by the Institute of Molecular and Cellular Biology in Osaka University. The data represent protein localization sites in the gram-negative bacteria. The vectors from 7 attributes compose 8 classes: cytoplasm (143 vectors), inner membrane without signal sequence (77 vectors), perisplasm (52 vectors), inner membrane (uncleavable signal sequence 35 vectors), outer membrane (20 vectors), outer membrane lipoprotein (5 vectors), inner membrane lipoprotein (2 vectors), inner membrane (cleavable signal sequence 2 vectors). Only the first five classes were used in the experiments. The data of the last three classes was omitted due to the small number to divide it into training and testing data and the peculiarities of noise injection technique.

5.3.2 Data Whitening

Before training, the data whitening transformation [3] was applied to all the data sets separately. Data whitening decorrelates and scales input variables in order for them to have the same variances. The representation of the data transformation matrix is:

$$F = \Lambda^{1/2}\Phi^{T} \tag{33}$$

where Λ and Φ are determined by Equation (34)

$$\Sigma = \Phi\Lambda\Phi^{T} \tag{34}$$

and Σ represents the covariance matrix of the data. But this value is not known in real world situations thus the estimated sample covariance matrix S is used instead of it.

The SLP obtains seven different statistical classifiers during its training

after such a transformation. The data whitening helps to reduce generalization error and speeds up training.

It should be noticed that in order to get all the data in the same transformation space, the transformation matrix has to be applied both to the training data and to the testing data.

5.3.3 Procedure of Experiments

The experiments were carried out 250×2 times to obtain reliable estimates: after reshuffling each class of data separately, half of the vectors were used for training, and the other half for testing. In subsequent trial, the training and testing data sets were interchanged. This procedure was repeated $N_e = 250$ times (only 25×2 experiments were performed with the Chromosome data set). Prior to training the classifiers, the data whitening transformations (33) applied. Moreover, prior to training the pair-wise classifiers, the two-class mean vectors were moved to zero point every time. Pseudo-validation sets were used to determine the SLP stopping moments and regularization parameters C of the SV classifiers. The "default" values for pseudo-validation set generation (using noise injection technique) were selected to be $k = 2$; $\sigma_{\text{noise}} = 1.0$; $ni_{\text{nn}} = 2$ based on experience

5.3.4 Results

The averages of generalization errors of three standard K-category classifiers and five two-stage algorithms (linear SV or SLP as the pair-wise classifiers were used in the first stage of decision making) are presented in Table 3. The last row (marked as σ^N) shows standard deviations σ of the fusion methods with the smallest error rate (printed in bold) divided by $\sqrt{N_e}$ (N_e = 250; N_e = 25 for the Chromosome data).

Methods	Chromo-somes	Iris	Realty	Satim	Wheat	Yeast	Wine	Ecoli
KSLP	0.262	0.097	0.078	0.177	0.088	0.136	0.025	**0.125**
KDA	0.270	0.033	0.074	0.214	0.098	0.131	**0.023**	0.136
RBF	0.202	0.031	0.053	0.149	0.073	0.190	0.024	0.129
+Vot.	0.195	0.042	0.056	0.147	0.069	0.149	0.032	0.138
	0.198	0.035	0.045	0.142	0.071	0.137	0.032	0.131
+H-T	**0.193**	0.038	0.056	0.144	0.067	0.146	0.032	0.147
	0.200	**0.027**	**0.043**	0.147	**0.063**	**0.129**	0.024	0.136
+WLW	**0.193**	0.038	0.056	0.145	0.067	0.147	0.032	0.148
	0.194	**0.027**	**0.043**	0.147	**0.063**	0.130	0.024	0.136
+Fuzz.	0.197	0.040	0.055	0.144	0.069	0.144	0.031	0.138
	0.200	0.038	0.098	**0.141**	0.197	0.150	0.095	0.348
+DAG	0.197	0.042	0.056	0.148	0.069	0.150	0.031	0.140
	0.199	0.036	0.045	0.143	0.072	0.140	0.034	0.133
σ^N	$8*10^{-4}$	10^{-3}	$7*10^{-4}$	$4*10^{-4}$	$9*10^{-4}$	10^{-3}	$9*10^{-4}$	10^{-3}

Table 3. Average generalization errors of the three benchmark one-stage and five two-stage SVM (the upper part of row) and SLP (the lower part of row) based classifiers.

The results in Table 3 show that the *K*-category net of SLPs was outperformed by other techniques in almost all the eight multi-class classification or pattern recognition (PR) tasks. No method proved to be the best one. The local nonparametric classifiers (KDA or RBF) were the best methods in two PR tasks. Diverse two-stage decision making methods were the best in six PR tasks. The pair-wise SLP with H-T fusion method could be considered to be statistically best. But actually the efficiency of the methods actually highly depends on the data as was shown in analysis of methods.

WLW performed also very well next to H-T fusion method compared to other pair-wise classifier fusion strategies. Both methods gave almost identical classifications in many cases, similarly as to the experiments with the 50D

Gaussian data (see Section 5.1). *After successful selection of the fusion rule* the employment of SLP based pair-wise classifiers was more beneficial than use of the SV ones in almost all experiments. The experiment with the K=24 class chromosome data set and a small number of cross-validation trials (N_e = 25) was a minor exception. Empirical study revealed that major attention should be paid to the performance of the base classifiers while designing the two-stage multi-category classifiers based on the pair-wise decisions,.

The superiority of SLP against SVM may be especially well seen with the *Iri*s data set. The optimal classifier for *Iris* data is the standard Fisher classifier. The error rate of 0.027 is obtained while employing Fisher classifier as a pair-wise classifier.

The differences between modern decision making schemes are usually rather small. Therefore it is worth stressing for a second time that the experimental evaluations highly depend on the exact split of available data into the training and test sets. For that reason, N_e = 20 or 100 cross-validation trials with reshuffled data are frequently insufficient to reliably reveal the differences between the methods. Accordingly, many more reshufflings have to be performed.

5.4 Bias Reduction in Fusion of Pair-wise Decisions

The training set based (re-substitution) estimates of the classification error rate are optimistically biased [3], [14], [27]. For that reason, usage of the training set based re-substitution estimates $\hat{m}^{R}_{i/ij}$, $\hat{m}^{R}_{j/ij}$ of $m_{i/ij}$, $m_{j/ij}$ and $\hat{s}_{i/ij}$, $\hat{s}_{j/ij}$ of $s_{i/ij}$, $s_{j/ij}$, can lead to incorrect estimates $\hat{r}^{R}_{ij}$ and $\hat{r}^{R}_{ji}$ of probabilities μ_{ij} and μ_{ji}. The over-adapted estimates worsen the H-T and WLW fusion rules. In an attempt to investigate the possibility to improve fusion rules, let us consider a theoretical way to reduce the optimistic bias of estimates $\hat{r}^{R}_{ij}$ and $\hat{r}^{R}_{ji}$ when binary Fisher classifiers are used. Denote

$$\hat{\delta}^{R}_{i/ij} = \hat{m}^{R}_{i/ij} / \hat{s}_{i/ij}\,,\ \hat{\delta}^{R}_{j/ij} = \hat{m}^{R}_{j/ij} / \hat{s}_{j/ij} \tag{35}$$

Distances $\hat{\delta}^{R}_{i/ij}$ and $\hat{\delta}^{R}_{j/ij}$ characterize pair-wise *re-substitution* classification error estimates of the Π_i, Π_j pair. The expected values of sample Mahalanobis distances are optimistically biased since

$$\mathrm{E}\,\hat{\delta}^{R}_{i/ij} = \delta_{ij}\ T_{\mathbf{m}i/ij}\ T_{\mathbf{\Sigma}} \ \geq\ \delta_{ij} \tag{36}$$

Terms $T_{\mathbf{m}i/ij}$, $T_{\mathbf{m}j/ij}$ and $T_{\mathbf{\Sigma}}$ in Equation (36) have been defined by Equations (28). Let us remember that specific distances should be used for each pair Π_i, Π_j. Suppose that $\hat{\delta}^{R}_{i/ij}$ and $\hat{\delta}^{R}_{j/ij}$ values for the pair Π_i, Π_j were already calculated from training data. Then "unbiased estimates" $\tilde{\delta}_{i/ij}$ and $\tilde{\delta}_{j/ij}$ of distances $\delta_{i/ij}$ and $\delta_{j/ij}$ may be obtained. To do this, Equation (36) has to be applied and the result in a certain interval of $\mathrm{E}\,\hat{\delta}^{R}_{i/ij}$ values has to be interpolated. Having the estimates of $\tilde{\delta}_{i/ij}$ and $\tilde{\delta}_{j/ij}$, the relationship (27) could be used in order to calculate the generalization errors:

$$P_{ij} = \Phi\{\ -\frac{1}{2}\ \hat{\tilde{\delta}}^{N}_{i/ij}\ \},\ \text{and}\ \ P_{ji} = \Phi\{\ -\frac{1}{2}\ \hat{\tilde{\delta}}^{N}_{j/ij}\ \},$$

where $\hat{\tilde{\delta}}^{N}_{i/ij} = \tilde{\delta}_{i/ij} / (\ T_{\mathbf{m}i/ij}\ T_{\mathbf{\Sigma}})$ and $\hat{\tilde{\delta}}^{N}_{j/ij} = \tilde{\delta}_{j/ij} / (\ T_{\mathbf{m}j/ij}\ T_{\mathbf{\Sigma}})$.

The distances $\hat{\tilde{\delta}}^{N}_{i/ij}$ and $\hat{\tilde{\delta}}^{N}_{j/ij}$ are expressed as fractions $\hat{\tilde{\delta}}^{N}_{i/ij} = \hat{\tilde{m}}^{N}_{i/ij}\ /\ \hat{s}_{i/ij}$, $\hat{\tilde{\delta}}^{N}_{j/ij} = \hat{\tilde{m}}^{N}_{j/ij}\ /\ \hat{s}_{j/ij}$ where $\hat{\tilde{m}}^{N}_{i/ij} = \hat{\tilde{\delta}}^{N}_{i/ij}\ \hat{s}_{i/ij}$, and $\hat{\tilde{m}}^{N}_{j/ij} = \hat{\tilde{\delta}}^{N}_{j/ij}\ \hat{s}_{j/ij}$ similarly to Equation (35).

After the above manipulations, the unbiased (corrected) estimates of conditional *aposteriori* probabilities μ_{ijd} and μ_{jid} to be used in the Hastie-Tibshiriani fusion rule design instead of Equation (11) are obtained:

$$\hat{\tilde{r}}_{ijd} = \frac{\phi(g(\boldsymbol{x}_d)\,|\,\hat{\tilde{m}}^{N}_{i/ij}, \hat{s}_{i/ij})}{\phi(g(\boldsymbol{x}_d)\,|\,\hat{\tilde{m}}^{N}_{i/ij}, \hat{s}_{i/ij}) + \phi(g(\boldsymbol{x}_d)\,|\,\hat{\tilde{m}}^{N}_{j/ij}, \hat{s}_{j/ij})}, \hat{\tilde{r}}_{jid} = 1 - \hat{\tilde{r}}_{ijd}$$

A similar method could be used to obtain unbiased (corrected) estimates of conditional *aposteriori* probabilities μ_{ij} and μ_{ji} for the Euclidean distance pair-wise classifiers. Instead of δ_{ij} and d, one has to use δ^*_{ij} and d^*_{ij} defined in Equation (30), and omit the term $T_{\mathbf{\Sigma}}$.

Apart from the theoretical methods, an *extra pseudo-validation set* to evaluate the over-adaptation bias of the H-T parameters $m_{i/ij}$, $m_{j/ij}$ and $s_{i/ij}$, $s_{j/ij}$ also may also be used. Artificial 50D Gaussian data (see Section 5.1) was used to perform a simulation study aimed to verify the usefulness of the following bias correction methods:

a) the EDC analytically based correction of the H-T parameters (it is formally assumed that the pair-wise classifiers are EDC ones (E correction)),

b) the same estimation method as the E correction, however, this time the Fisher DFs were supposed to be used as the pair-wise classifiers (F correction),

c) the estimates of the H-T parameters were obtained from the first pseudo-validation data, V1, already used for determination of the SLP's optimal stopping moment (V1 correction), and

d) the estimates of H-T parameters were obtained from an extra (second) pseudo-validation data, V2, generated to estimate the H-T parameters (V2 correction). This experiment was done in order to exclude the adaptation to V1 data while determining the optimal number of iterations.

A standard training set based H-T procedure was used as a benchmark method. To verify the potential abilities of the bias elimination, an "ideal" H-T procedure where parameters (12) would be evaluated exactly was used. The fusion rules where the H-T parameters were estimated on a very large test set composed of 10,000 vectors, were designed for this purpose. The average values of the classification errors obtained in N_e=1,000 independent cross-validation trials are shown in Table 4. Parameter C of SV and stopping of SLPs were accomplished on the basis of generalization errors estimated on pseudo-validation sets generated with parameters $ni_{\mathrm{nn}} = 50$, $k = 2$, and $\sigma_{\mathrm{noise}} = 1.0$.

Fusion method	SLP	SVC
learn set based H-T	0.0446	0.0793
H-T with E correction	0.0443	0.0787
H -T with F correction	0.0440	0.0783
H -T with V1 correction	0.0439	0.0772
H -T with V2 correction	0.0439	0.0769
test set based H-T	0.0424	0.0741

Table 4. Generalization errors of the voting and H-T fusion methods using diverse correction terms

All four bias diminishing schemes gave a gain in comparison with the benchmark method in both situations where SV or SLPs were used as the pair-wise binary classifiers. The empirical bias correction methods much more outperformed the analytical ones. The differences between the benchmark and ideal H-T methods, the *test set based H-T* procedure (the hypothetical limit value), were 0.0446-0.0424 = 0.0022 for SLP and 0.0793-0.0741 = 0.0052 for SV binary classifiers. Empirical *V2 correction* method reduced the classification error rate of the benchmark fusion rule by 0.0446-0.0439 = 0.0007 for SLPs and 0.0793-0.0769 = 0.0024 for SVs, i.e. 32% and 42% of the possible ideal error rate decrease. Though the nominal increase is not considerable, the relative increase is worth of attention. The SV based scheme was improved much more than the SLP based. This fact advocates that the SLP classifiers demonstrated good generalization properties in this case. The gain was based on the fact that *the colored noise injection introduces useful supplementary information into the decision making algorithm* (see Section 3.3.4).

5.5 Concluding Remarks

The experimental results presented in this section confirmed assumptions and analytical results of the proposed two-stage classification based on pair-

wise classifiers. Since the results are not considerably better in some cases, or in a few cases even a little bit worse, the main target was achieved – it was shown that proposed two-stage strategy in multi-class classification task solving is not worse that the known standard multi-class classifiers providing opportunities for further enhancement.

The analysis of methodical issues showed that different methods relaying on different optimization criteria should be estimated carefully. It appeared that the two analyzed classifier fusion methods, H-T and WLW, are approximately the same for moderate and small sample size data in the context of classification error rate. It was also shown that the number of experiments sometimes may be critical for not exact conclusions of classifier comparison.

The analysis of classifiers' fusion correction due to their bias shows, that the method based on introduction of new pseudo-validation data is more promising than the well known analytical methods based on statistical assumptions.

Chapter 6

Application to Mineral Classification

After obtaining main results of these theses a new challenge was presented from Selçuk University. They provided complicated imbalanced mineral data with a small sample. The multi-class classification approach with data modification was successfully applied.

The data and the relevance of research in the domain and the data are described in Section 6.1 and Section 6.2. The results of research and its analysis are presented in Section 6.3. The practical aspects of the research are discussed in Section 6.4.

6.1 Domain Task

Color is a fundamental physical property of image processing. It is widely used in physical analysis [31], [32], [33]. In optical mineralogy, color is used for the recognition of minerals in order to identify the rock names. Color is useful for the recognition of minerals under microscopes with polarized light. Hence, microscopes with polarized light capabilities are used for optical mineralogy [34], [35], [36], [37].

Microscopes are commonly used for manual mineral identification in thin sections. But there are some problems concerning color that depend on a variety of factors including illumination, mineral type, the thickness of the thin section etc. Thus, automated mineral identification systems [35], [36] are based on scanned or plane and polarized images and use the natural color of the mineral.

Today, many vision systems appear for the quality control of products appear in all areas [38]. They have been applied for boundary detection, segmentation, feature extraction and identification of objects. Because of these varieties of

applications, image vision is getting popular and is also used in different fields [39], [40], [41]. In this study, thin section images were analyzed by using image processing in order to identify minerals.

The obtained color parameters of the minerals have to be passed to a classification system in order to know which class of minerals they represent. A lot of research has been done on mineral exploration, e.g. [35], [36] [37]. Most of researchers paid principal attention to obtaining the data, its preprocessing and proper feature selection. But they missed a thorough analysis of experiment performance. That could lead to inaccurate results. The most frequently used method of mineral classification is an artificial neural network (ANN) with one hidden layer [35], [37]. But its shortcoming is that the selection of proper architecture (i.e. selecting the best amount of neurons in each layer) is time consuming.

The objective of this study is to find a simple and reliable method suitable for mineral data classification. Two types of classification methods were considered in this study: standard multiple class classifiers and two-stage classifiers based on simple pair-wise classifiers. The latter were selected due to their lightweight and fixed architecture and proven ability to perform not worse or even better than that standard ANNs solutions.

The mineral data distributions analyzed in this study are rather complicated – classes have highly diverse sizes and covariance matrixes. Besides, classes are overlapping and located near each other. Thus similarity features were employed in order to separate data and to get higher classification performance.

6.2 Data

In this study thin sections were observed using the James Swift microscope. Images were taken by a digital camera in a rotating experimental stage instead of a fixed stage. The experimental stage can be rotated from 0 to 180 degrees by 1 degree increments, while polarizer and analyzer remain

crossed to each other in a vertical direction during the analysis. Illumination source was a 12V/100W halogen light. Thin section images were taken with a Videolab camera mounted on the microscope. Images were transmitted to a computer by Inca software.

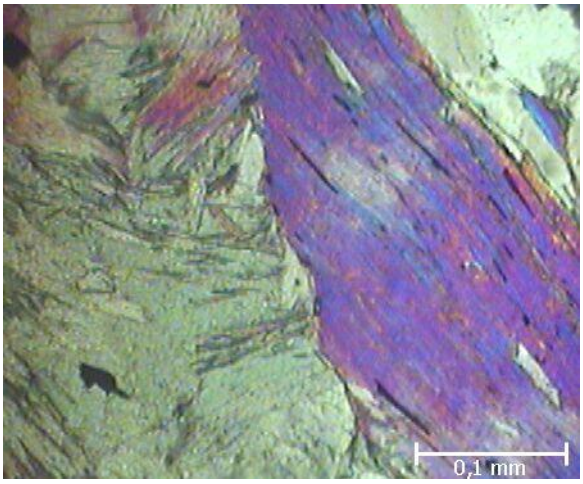


Figure 13. Maximum intensity image of minerals

Images obtained both under plane-polarized and cross-polarized light contain the maximum intensity values (Figure 13). The images were first captured at every 10^{th} increment and then compared to the previous images for maximum intensity. All images were stored in RGB format with the dimensions of 450x370 pixels and the resolution of 150 dpi.

Twenty-two digital images were taken from nine thin sections. Thin sections were taken from the department of geological engineering in Selçuk University, Turkey. In this research, a total of 5 common minerals – quartz (110 samples), muscovite (110 samples), biotite (60 samples), chlorite (60 samples) and opaque (60 samples) – were used. For image quantization, first a median filter was applied to images for noise reduction and then the histogram was equalized. Thus 6 features of each mineral image pixel were obtained. The first three color parameters were extracted from cross-polarized light, and the other three from plane-polarized light.

Prior to training the classifiers, the data was normalized by standard deviations of each single feature. Then the principal components and eigenvalues of pooled sample covariance matrix were used to transform the data prior to training the SLPs and SVCs. Moreover, the two-class mean vectors were moved to the zero point prior to training the pair-wise classifiers, each time.

6.2.1 Similarity Features

Similarity features were employed in this study. For any original data vector x_i the similarity feature vector consists of N_{tr} (the number of training vectors) components:

$$s_i^j = \exp(-\alpha * \sqrt{\sum_{k=1}^{d} (x_i^k - x_j^k)^2}) \qquad (37)$$

where d is the dimensionality of data, j is the index of the j^{th} similarity feature, vector superscript k denotes the k^{th} element of the original data vector and α is the normalization coefficient. This way the new dimension becomes N_{tr}. In this research half of the 400 data vectors were used for training. Thus a 200-dimensional similarity feature vector s_i was obtained for each data vector x_i.

6.3 Practical Results

In order to get a rather high reliability of results, 250 2-fold cross-validation generalization estimation procedures were performed for all the data and all the methods. The data was permuted 250 times, dividing it into two equal pieces for training and testing. The same permutations of data were used in all kinds of experiments (different data and classification methods), i.e. all the experiments were performed with exactly the same data sets.

First all the classification algorithms were employed with original data. The classification error rates were rather high. It was 0.211 (see Table 5) for

the best pair-wise method (DAG + SVC as a pair-wise classifier). The multi-class RBF method with an error rate of 0.189 showed the best results among all methods.

As in the studies done by other researchers, e.g. [35], [37], ANN with one hidden layer was also used for the original mineral data. Neurons in hidden layer were selected (by employment of pseudo-validation data) from an empirically predefined set. The error rate obtained with such an ANN was 0.25 – the worst out of the used methods. Besides, as it was already mentioned before, the selection of neurons in hidden layer was highly time consuming. Thus the use of this method in further study was eliminated.

The effect of similarity feature employment may be seen in Figure 14. The classes became "*C*" shaped and more separable.

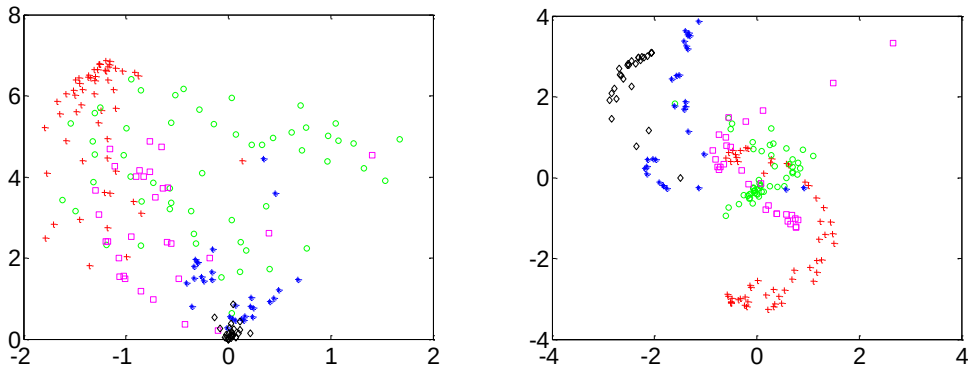


Figure 14. Original data of five different minerals (left) and the same data after employment of similarity features (right). Dimensions were reduced according to eigenvalues of covariance matrixes (principal component analysis method). Same shapes and colors mean the same minerals.

Parameter α was selected for each classification method from an empirically formed set of values [0.0001, 0.001, 0.01, 0.1, 0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 1.8, 2] using validation data generated by noise injection. The α was selected a new every time a new data permutation was used in training, i.e. 250x2 times.

	One stage classifiers			Two stage classifiers		
Data parameters	RBF	KDA	K-SLP	Voting	DAG	H-T
Original data	0.189	0.226	0.252	0.212/0.211	0.215/0.211	0.226/0.218
Similarity features	0.177	0.212	0.174	0.227/0.174	0.238/0.174	0.173/0.183
Best α	10^{-4}	0.1	0.5	0.1	0.1	0.5
Best α rate	0.63	0.83	0.47	0.86	0.86	0.54

Table 5. Results of mineral data classification. The left value in the cells of two-stage algorithm results show the average error rate with SLP as a pair-wise classifier, and the right one shows the average error rate with SVC as a pair-wise classifier. The most often selected similarity features' parameter □□ and its selection rate (from all experiments) are presented in the last two lines.

The results presented on the second line of Table 5 show that the multi-class one stage methods and pair-wise methods based on SVC showed better results than using original data despite the increase of dimensionality.

The best method in the experiments was a two-stage method based on SLP as pair-wise classifiers and using Hastie-Tibshirani fusion method (SLP+HT). For the estimation of inaccuracy the Equation (32) may be used. Actually, all of the data is considered in one experiment (in one of the 2 cross validation sets) in 2-fold cross validation. Thus despite having performed 250 experiments, generalization error was estimated on the same 400 data vectors. So the inaccuracy of estimation of generalization error of SLP+HT method is $\sqrt{0.173(1-0.173)/400} = 0.019$. It may be seen that many other methods also performed well since their error rate fits within this accuracy interval (i.e. up to 0.173 + 0.019 = 0.192). After employing similarity features, some multi-class methods performed better as well as pair-wise ones, but the latter ones are recommended for practical use due to their vast ability for further improvement.

The results with employment of pair-wise SLP in voting-based fusion methods (Voting and DAG) are not as good because of their sensitivity to the

sample size and dimensionality ratio. This behavior is due to their similarity to statistical methods [17] which work well if the sample size is much greater than dimensionality because of problems arising with covariance matrix estimation (see e.g. [42] for more). It may be also observed from the results that probability estimation based H-T fusion method overcomes the shortcomings of SLP classifier and exploits its results best.

In order to overcome high dimensionality problems for SLP as pair-wise classifier in voting based fusion methods simple linear dimension reduction using eigenvalues of covariance matrixes - principal component analysis – was used. Dimensionalities from a vast set of different values from 2 to 195 (the total dimensionality of similarity features was 200) were reselected. The experiments showed that better results may be obtained with dimensionality reduced to proper size. E.g. the results may become 1.5 times worse when using dimensionality equal to 2 because of the loss of some information. While using dimensionality between approximately 40 and 50, SLP as pair-wise and Voting as fusion method (SLP + Voting) may perform with a generalization error rate of 0.176, that is much better than without the dimensionality modification (error rate of 0.227). If dimensionality is further increased, then classification error increases again due to redundant additional information. The generalization errors of other methods also decreases while reducing dimensionality, but only within the above mentioned accuracy.

The results of this study also showed that the parameter α used to obtain similarity features highly depends on both the classification method (see the best values in line 3 of Table 5) and the data permutation (see the percentage of use of the best α value in line 4 of Table 5).

6.3.1 Reliability of Results

The number of experiments with different data permutations plays a great role in the reliability of results. If we take a single experiment out of 250 2-fold experiments done using similarity features (without dimensionality reduction),

the best and the worst ones – which are highly different from the means of all experiments listed in Table 5 – may be selected. E.g., an error rate of even 0.125 was obtained in one of the experiments with SLP+Voting method, while an error rate of up to 0.25 was obtained in the worst experiment for the best averaged methods (e.g. SLP+HT or K-SLP). So it is necessary to perform a rather large amount of experiments in order to obtain reliable results.

Although the average results of the experiments do not seem perfect, they are satisfactory for the mineral classification, since an error rate of even 0.25-0.3 is acceptable for this task. Besides, the accuracy of classification may be increased by obtaining more pixels from the same mineral to be classified.

6.4 Discussions on Practical Application

Similarity features were used in order to simplify the complexity and obtain non-linear decision boundaries in the complex shaped input feature space. This data transformation makes data more separable. On the other hand it enlarges the dimension count and makes it hard for SLP pair-wise classifiers to learn due to their similarity to classical statistical classifiers. Thus using similarity features and two-stage classification methods with pair-wise classifiers or fusion rules which are less sensitive to dimensionality increase or K-SLP multiple class classification method would be recommended for mineral classification.

Since the data describing six different mineral color features was very complicated, the simple SLP fusion using voting techniques provides worse results than with SVM. But when the fusion is done in a more advanced manner (estimating probabilities) it outperforms SVM and other methods. The outperforming is not statistically considerable, but it shows that wisely fused SLPs may produce results not worse than in other classification methods. Thus the main issue in classification of such complicated data is the proper selection of parameters (both the classifier and the data). In order to get a precise

estimation of methods, a lot of experimental results should be applied on as much data as possible.

The proposed two-stage classification methods are much better in calculation speed than multilayer artificial neural networks since they do not require a special training procedure to obtain proper architecture (input values, the number of hidden neurons).

The novelties of this research are 1) using two-stage classification methods to classify multi-class mineral data, 2) using similarity features to make mineral data more non-linearly separable, and 3) using a pseudo-validation data set to select the parameters for the decision making algorithms.

The straightforward way of using similarity features and dimension reduction by principal component analysis was used in this research. The decision making strategy for mineral data presented in this thesis could be improved by using advanced similarity feature selection techniques. It also gives an opportunity to introduce straightforward pricing of incorrect classification, which is an important issue in industrial applications.

6.5 Concluding remarks

The main results of the thesis were applied on complicated multi-class mineral data in this chapter. The results confirmed that two-stage classification strategy based on pair-wise classifiers provides not worse results than other multi-class classifiers, if properly used and leaves opportunities for further enhancement. Similarity features were employed in order to transform the data space to a more separable one. It appeared that even dimensionality-sensitive classifiers such as SLP may get considerable gain in employment of similarity features if properly fused, which considerably increases dimension when data sample size is much larger than dimensionality.

The obtained experimental results showed that the proposed strategy performs in generalization error which is enough in order to apply it in industry.

Chapter 7

Conclusions

The recommendations for multi-class classifier designers who face data with unknown distributions, are presented in this final chapter. The conclusions of this thesis are divided into core (Section 7.2) and additional not so significant (Section 7.3) conclusions.

7.1 Recommendations for Multi-class Classification Task Designers

The data should be at least normalized – the values of all data vector attributes should have the same value range. Then other proper data transformation has to be performed. In this thesis the data transformations using singular value decomposition and similarity features were proposed.

If the distributions of classes are known, special classifiers should be used. If not, SLPs as pair-wise classifier s should be used after the data transformation since they are able to stop their learning at the moment they reach the optimal statistical classifier for particular data.

The experiments showed that the H-T fusion rule performed the best in most cases. But despite that, the use of *Pair-wise Fuzzy Templates* fusion rule is proposed when the statistical characteristics of class pairs are very different.

It is very important to perform a rather large amount of experiments in order to get reliable results. Due to the small number of experiments, the estimation of generalization error is not exact.

7.2 Main Conclusions

The standard cost function of the multi-category nets of SLPs does not directly minimize the classification error. As a consequence, it does not allow to always obtain optimal pair-wise classifiers even in the cases when prior probabilities of the classes are the same and the training sets are balanced. The only classifiers aimed to obtaining decision making rules that minimize the classification error rate (supposing that sufficient sample data is obtained) involve methods based on statistical decision theory. To allow the minimization of the classification error rate, the task optimization properties of two-stage decision making scheme (where the optimal pair-wise classifiers are obtained in the first phase and the decisions of the pair-wise classifiers are are fused with proper fusion rule in the second phase) were analyzed. The main results of this thesis are:

1. It was theoretically shown why the two-stage neural network based decision making procedures may outperform the single-stage ones. It was found out that this is due to the following reasons: (a) refusing the traditional *K*-class cost function, (b) allowing to obtain near to optimal pair-wise linear classifiers by specially organized SLP or SV training, and (c) ability to save useful discriminative information contained in the first stage classifiers by prudent fusion of the pair-wise decisions. Actually this decision making scheme transfers the difficulties of multi-class classification task (e.g. imbalanced classes, imperfect cost function) to the better explored and better performing two-class classification case.
2. It was shown that successfully stopped pair-wise SLP based classifiers are a useful option in the first stage of decision-making. This is due to the SLP feature to evolve through seven different statistical classifiers and to stop at the moment, when optimal statistical classifier is reached. If a prudently trained fusion rule is used, the SLPs are comparable and often outperform the linear SV classifiers in moderate dimensional

situations.

3. The presented pair-wise Fuzzy Templates method is a proper classifier fusion method when statistical characteristics of class pairs are rather different. Experimental results showed that SVM as a pair-wise classifier is better than SLP with such method when standard parameters are used. But introducing a new scaling parameter α in transfer function makes SLP not worse than SVM.

7.3 Other Results

1. It was demonstrated that two fusion strategies – the K-L distance based Hastie-Tibshirani and the WLW method – result in approximately the same performance when the training sample sizes are small. This may actuality be explained by the theoretical fact that an excessive minimization of inexact criteria may become harmful (see the NN overtraining and feature over-selection in [25]). After performing the numerical analysis of the simultaneous effect from two sources of inaccuracies – (a) the simplification of performance measures and (b) the finiteness of sample size – it was found that the sample size was a major source of the increase of the classification error in the fusion rule design.
2. The repeated employment of training data to design fusion rules leads to optimistic bias and deterioration of the two-stage decision-making system. The empirical, pseudo-validation set based, bias diminishing technique appeared to be more effective than the theoretical, multi-dimensional Gaussian distribution model based methods (e.g. corrections for Fisher classifier). In this context a colored noise injection once more proved to be a powerful tool to facilitate finite sample size based model selection problems in moderate-dimensional pattern recognition tasks.

3. Experimental results showed that a directed acyclic graph is a proper fusion method not only with SVM, but with SLP as well.

Some experiments showed that the performance difference between the proposed two-stage classification and one stage classification is not statistically significant. Despite that, the pair-wise classifiers are more promising due to the possibility to use different features and methods best suited for each pair. Besides, when new training data of a particular class is introduced for training, there is no need to train the overall network – only K-1 pair-wise classifiers.

Since it was proved that the approach of using simple pair-wise classifiers as the first stage classifiers in two-stage multiclass classification is a good alternative to complex one-stage classification methods, it opens an area for further research with more complex pair-wise classifiers which could better employ the features of statistical class pair features.

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford Univ. Press, 1995.

[2] S. Haykin, *Neural Networks: A comprehensive foundation*. 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1999.

[3] S. Raudys, *Statistical and Neural Classifiers: An integrated approach to design*. Springer-Verlag, NY, 2001.

[4] Z.-H. Zhou and X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. on Knowledge and Data Engineering*, 18(1): 63-77, 2006.

[5] Hsu, C. W., Lin C. J., A comparison on methods for multi-class support vector machines. *IEEE Trans. on Neural Networks*, 13: 415-425, 2002

[6] T. Hastie and R. Tibshirani, Classification by pair-wise coupling, *The Annals of Statistics*, 26(1):451-471, 1998.

[7] T.-F. Wu, C.-J. Lin, R. C. Weng, Probability estimates for multi-class classification by pair-wise coupling, *Journal of Machine Learning Research*, 5: 975–1005, 2004.

[8] A. H.R. Ko, R. Sabourin, A. de Souza Britto Jr., and L. Oliveira, Pair-wise fusion matrix for combining classifiers. *Pattern Recognition* 40, 2198–2210, 2007.

[9] M. Gonen, A.G. Tanugur, and E. Alpaydm, Multiclass posterior probability support vector machines. *IEEE Tr. Neural Networks*, 19: 130–139, 2008.

[10] J. Cid-Sueiro, J.I. Arribas, S. Urban-Munoz, and A. R. Figueiras-Vidal, Cost functions to estimate *a posteriori* probabilities in multi-class problems. *IEEE Trans. on Neural Networks*, vol. 10 (3): 645- 656, 1999.

[11] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAG's for multi-class classification", in: *Advances in Neural Information Processing Systems.* Cambridge, MA: MIT Press, vol. 12 pp. 547-553, 2000.

[12] E. Gelenbe and K.F. Hussain, Learning in the multiple class random neural network, *IEEE Trans. Neural Networks* 13 (6): 1257–1267, 2002.

[13] L. Kuncheva, J. C. Bezdek, M. A. Sutton, *On combining multiple classifiers by fuzzy templates*, 1998.

[14] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. 2nd edition. Academic Press, NY, 1990.

[15] S. Raudys. Integration of statistical and neural methods to design classifiers in case of unequal covariance matrices. *Lecture Notes in Computer Science*, Springer, Vol. 3238, pp. 270-280, 2004.

[16] S. Raudys and S. Amari, Effect of initial values in simple perception, In *Proc.1998 IEEE World Congress on Comp. Intelligence.* IEEE Press, CNN'98: 1530–1535, 1998.

[17] S. Raudys, Evolution and generalization of a single neurone. I. SLP as seven statistical classifiers, *Neural Networks*, 11: 283–96, 1998.

[18] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Berlin. 1995.

[19] M. Skurichina, S. Raudys, and R.P.W .Duin, K-NN directed noise injection in multilayer perceptron training, *IEEE Trans. on Neural Networks*, 11(2): 504–511, 2000.

[20] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[21] C.-W. Hsu, C.-C.. Chang, and C.-J. Lin, *A Practical Guide to Support Vector Classifcation*, 2009; http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[22] K.-P. Wu and S.-D.Wang, A weight initialization strategy for weighted support vector machines. *Lect. Notes in Computer Science,* Springer. Vol. 3686, pp. 288-296, 2005.

[23] A. Asuncion, and D.J. Newman, *UCI Machine Learning Repository* Irvine, CA: [http://www.ics.uci.edu/~mlearn/MLRepository.html], 2007.

[24] A. Babalik, Ö.K. Baykan, and F.M. Botsali, Determination of wheat kernel type by using image processing techniques and ANN. *Proc. of Int. Conf. on Modeling and Simulation*, Konya/Turkey, vol. 1: 531-534, 2006.

[25] S. Raudys, Feature over-selection. *Lect. Notes in Computer Science*, Springer. Vol. 4109, pp. 622-631, 2006.

[26] S. Amari, N. Fujita, S. Shinomoto, Four types of learning curves, *Neural Computation* 4:605–618, 2002.

[27] S. Raudys and D.Young, Results in statistical discriminant analysis: A review of the former Soviet Union literature, *Journal of Multivariate Analysis*, Vol. pp. 89, 1-35, 2004.

[28] Hamamura T., Mizutani H., and Irie B. 2003. A Multiclass Classification Method Based on Multiple Pairwise Classifiers. In *Proceedings of the Seventh international Conference on Document Analysis and Recognition - Volume 2* (August 03 - 06, 2003). ICDAR. IEEE Computer Society, Washington, DC, 809

[29] B. Boser, I. Guyon and V. Vapnik, A Training Algorithm for Optimal Margin Classifiers. In Proceedings of the Fith Annual Workshop on Computational Learning Theory, pages 144-152. ACM Press, 1992

[30] S. Raudys, Generalization error of multinomial classifier *Lecture Notes in Computer Science*, Springer, Vol. 4109, pp. 503 – 511, 2006.

[31] Segnini, S., Dejmek, P., Öste, R., 1999. A low cost video technique for colour measurement of potato chips. Lebensmittel-Wissenschaft und-Technologie 32 (4), 216-222

[32] Yam, K.L., Papadakis, S.E., 2004. A simple digital imaging method for measuring and analyzing color of food surfaces. Journal of Food Engineering 61 (1), 137-142

[33] Gökay, M.K., Gundogdu, I.B., 2008. Color identification of some Turkish marbles. Construction and Building Materials 22 (7), 1342-1349

[34] Fueten, F., 1997. A computer-controlled rotating polarizer stage for the petrographic microscope. Computers & Geosciences 23 (2), 203-208

[35] Marschallinger, R., 1997. Automatic mineral classification in the macroscopic scale. Computers & Geosciences 23, 119–126

[36] Thompson, S., Fueten, F., Bockus, D., 2001. Mineral identification using artificial neural networks and the rotating polarizer stage. Computers & Geosciences 27 (9), 1081-1089

[37] Fueten, F., Mason, J., 2007. An artificial neural net assisted approach to editing edges in patrographic images collected with rotating polarizer stage. Computers & Geosciences 33 (9), 1176-1188

[38] Bombardier, V., Schmitt, E., Charpentier, P., 2009. A fuzzy sensor for color matching vision system. Measurement 42 (2), 189-201

[39] Komenda, J., 2001. Automatic recognition of complex microstructures using the Image Classifier. Materials Characterization 46(2-3), 87-92

[40] Akesson, U., Stigh, J., Lindqvist, J.E., Göransson, M., 2003. The influence of foliation on the fragility of granitic rocks, image analysis and quantitative microscopy. Engineering Geology 68(3-4), 275-288

[41] Forero, M.G., Sroubek, F., Cristobal, G., 2004. Identification of tuberculosis bacteria based on shape and color. Real-Time Imaging 10(4), 251-262

[42] Friedman, J. H., 1989. Regularized discriminant analysis. Journal of the American Statistical Association 84 (405), 165-175

[43] Wald, P.W., Kronmal, R.A., 1977, Discriminant function when covariances are unequal and sample sizes are moderate. Biometrics, 33, 479-484.

[44] Raudys, S. 2006. Trainable fusion rules. I. Large sample size case. Neural Netw. 19, 10 (Dec. 2006), 1506-1516

[45] James, W. and Stein, C., 1961, Estimation with Quadratic Loss, Proc. Fourth Berkley Symp. Math. Stat. Prob., 1, 361-379, UNversity of California Press.

[46] Stein, C., Efron, B. and Morris, C., 1972, Improving the Usual Estimator of a Normal Covariance Matrix, Departament of Statistics, Stanford University Report No. 37

[47] Titterington, D.M., 1985, Common Structure of Smoothing Techniques in Statistics, Int. Statist. Review, 53, 141-170

[48] O'Sullivan, F., 1986, A Statistical Perspective on Ill-Posed Inverse Problems, Statistical Science, 1, 502-527

[49] Park., S-H., Furnkranz, J., 2007, Efficient Pairwise Classification, Proc. of the 18th European conference on Machine Learninig, 658 – 665

[50] Goldberg, D. E., Genetic Algorythms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley, 1989

[51] Ishibuschi H., Nozaki K., Tanaka H., Distributed Representation of Fuzzy Rules and its Application to Pattern Classification, Fuzzy Sets and Systems, North – Holand, pp. 21-32, 1992

[52] Chieh-Neng Young, Chen-Wen Yen, Yi-Hua Pao and Mark L. Nagurka, One-Class-at-a-Time Removal Sequence Planning Method for Multiclass Classification Problems IEEE Transactions on Neural Networks 17(6): 1544-1549, 2006

[53] Moshe Leshno, Vladimir Lin, Alan Pinkus and Shirnon Schocken, Multilayer Feedforward Networks With Non-Polynomial Activation Functions Can Approximate Any Continuous Function. Journal of Neural Networks, 6(3), 1993

[54] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification. 2nd ed. Wiley, NY, 2000.

[55] N. Japkowicz, "Learning from imbalanced data sets: a comparison of various strategies," in Working Notes of the AAAI'00 Workshop on Learning from Imbalanced Data Sets, Austin, TX, pp.10–15, 2000.

[56] Sarunas Raudys, Rimantas Kybartas, Edmundas Kazimieras Zavadskas. *Multicategory Nets of Single-Layer Perceptrons: Complexity and Sample-Size Issues*. IEEE Transactions on Neural Networks, Vol. 21, No. 5, p. 784 – 795, 2010

[57] Arthur Earl Bryson, Yu-Chi Ho (1969). Applied optimal control: optimization, estimation, and control. Blaisdell Publishing Company or Xerox College Publishing. pp. 481

[58] Y. S. Huang and C. Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. IEEE Trans. PAMI, 17(1):90 – 94, 1995.

[59] J. R. Quinlan. Induction of Decision Trees. Machine Learning 1: 81 – 106, 1986.