Multicategory Nets of Single-Layer Perceptrons: Complexity and Sample-Size Issues
Sarunas Raudys, Rimantas Kybartas, and Edmundas Kazimieras Zavadskas

Abstract—The standard cost function of multicategory single-layer perceptrons (SLPs) does not minimize the classification error rate. In order to reduce classification error, it is necessary to: 1) refuse the traditional cost function, 2) obtain near to optimal pairwise linear classifiers by specially organized SLP training and optimal stopping, and 3) fuse their decisions properly. To obtain better classification in unbalanced training set situations, we introduce the unbalance correcting term. It was found that fusion based on the Kulback-Leibler (K–L) distance and the Wu–Lin–Weng (WLW) method result in approximately the same performance in situations where sample sizes are relatively small. Comprehensive comparative investigations of six real-world pattern recognition (PR) problems demonstrated that employment of SLP-based pairwise classifiers is comparable and as often as not outperforming the linear support vector (SV) classifiers in moderate dimensional situations. The colored noise injection used to design pseudovalidation sets proves to be a powerful tool for facilitating finite sample problems in moderate-dimensional PR tasks.

Index Terms—Classification, complexity, generalization, multiclass, perceptrons, support vectors (SVs).

I. INTRODUCTION

The standard $K$-category net of single-layer perceptrons (SLPs) is composed of $K$ SLPs [1]–[3]. At times such multicategory scheme does not work successfully. A powerful stream of research aimed at improving multicategory classification by pairwise classifications has been pursed in the trends of two-stage decision making. Here one actually splits a multidimensional input feature space into regions. Most often the support vector (SV) or decision-tree-based classifiers are used in the first stage [4]–[8]. In the second stage, one is trying to assign a class label to each region. Special fusion rules are developed for processing $K(K-1)/2$ pairwise outputs and for the final decision making [9]–[12]. The Kulback–Leibler (K–L) distance [6], diverse variants of the sum of squares of pairwise 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 (see Section II).

Another modification of $K$-category SLPs consists of $K$ separate units. Each of them is trained to discriminate one class from other ones (i.e., one-against-all approach). Many investigations compare the accuracy of diverse methods by means of simulation. It was found that the two-stage pairwise decision making is more promising [5]. For that reason, we are investigating this approach in our paper. In the majority of the investigations, differences between fusion methods applied were minor. It is worth mentioning that degrees of the experimental comparisons were insufficient very often. The researchers paid most attention to a number of benchmark data sets, however, they often forgot about a reliability of their experimental evaluations. If the data size is not large, a random partition of the data that are put into the training and test sets formation yields diverse results. For this reason, the estimates become unreliable. In the best cases, cross-validation experiments have been performed 20 times after reshuffling the data of each class every time [7]. We reshuffled each class data $N_c = 250$, 500, or even 1000 times.

A better elucidation of the existing theories sometimes may be more valuable than the proposal of new ones without proper theoretical explanation. The two-stage decision making is one of such areas. The two-stage algorithms are perspective since researchers may adapt a number of features and complexities of the binary classifiers to the training set size. In addition, it becomes easier to solve problems with imbalanced training sets. Therefore, we need theoretical clarification of reasons why the two-stage decision making schemes are promising. This paper is aimed at elucidating the complexity and sample size issues of known procedures, as well as effects of inaccurate criteria employed to construct the fusion rules. In Section II, we review the single and two-stage classifiers, which will be considered in further analysis. We also present comprehensive experimental comparisons. Section III investigates the effects of inaccuracies of the K–L distance and a sum-of-squares error functions. Section IV considers small sample complexity problems. The concluding remarks are in Section V.

II. METHODS INVESTIGATED

A. Statistical Classifiers and Their Implementation

Statistical decision theory proposes a method to design optimal $K$-category classifiers, provided that density functions and prior probabilities $q_1, q_2, \ldots, q_K$ of the classes are known. Assuming that distribution densities of the $p$-dimensional input feature vector $\mathbf{x}$ are multivariate Gaussian characterized by different mean vectors $\mu_1, \mu_2, \ldots, \mu_K$ and
common covariance matrix (CM) $\Sigma$, we obtain an optimal $K$-category linear classifier [1], [14]. In practice, unknown parameters $\mu_1, \ldots, \mu_K$ and $\Sigma$ are replaced by training-set-based estimates $\bar{x}_1, \ldots, \bar{x}_K$ and $\hat{\Sigma}$. In the $K$-category case, a standard Fisher linear discriminant function (DF) actually is a set of $K$ ones

$$g_i^{DF}(x) = x^T \mu_i + \nu_0, \quad i = 1, 2, \ldots, K \quad (1)$$

where weight vector $\mu_i$ is $S^{-1} \bar{x}_i$, bias term $\nu_0 = -\bar{x}_i^T S^{-1} \bar{x}_i + \ln \eta_i$, and an allocation of vector $x$ is performed according to a maximum of $g_i^{DF}(x), g_j^{DF}(x), \ldots, g_k^{DF}(x)$.

If the dimensionality $p$ is very high, and training set sizes $N_1, N_2, \ldots, N_K$ are too small to obtain a reliable estimate of the $p \times p$ matrix $\Sigma$, then small positive constants $\lambda$ are added to the diagonal elements of matrix $\Sigma$. In such a way, we obtain a regularized discriminant analysis (RDA; see [3], [14]). If $\lambda \to \infty$, one ignores the matrix $\Sigma$. So we have 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, instead of the linear, quadratic DFs should be used. Quadratic DF is very sensitive to training set size; covariance matrices are supposed diverse and we need a large number of samples to obtain estimates $\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_K$. One of the practically useful remedies is to use pairwise classifiers with pooled sample estimates of the covariance matrices. For pair $\Pi_i$ and $\Pi_j$, we use $S_{ij}(\pi_{ij}) = S_i + S_j - \pi_{ij}$ and determine the bias term in a special way (the Anderson–Bahadur (A–B) linear DF [3], [14]). In this type of linear classifiers, classes $\Pi_i$ and $\Pi_j$ use “common” CM. For that reason, in comparison with quadratic DF, the A–B rule has much better small sample properties [15].

The SLP-based classifier can realize the Fisher DF and in further training, the A–B DF. Most often, it performs well even when $S_j \neq S_i$.

**B. The K-Category SLP-Based Neural Net for Classification**

A network of $K$ SLPs [1]–[3], [14], [16]–[18] has $K$ outputs $o_i = f(v^T x + u_0) = f(\bar{u}^T z)$ (i = 1, 2, …, K), where $\bar{u}$ is a $p$-dimensional weight vector, $u_0$ is a bias weight, $\bar{u} = (u_0, u_1^T, \ldots, u_p^T)^T$, $z = (1, x)^T$, and $f(s)$ is a nonlinear sigmoid activation function. We assume $f(s) = 1/(1+ \exp(-s))$.

To obtain the weight vectors, we minimize the cost function of the sum of squares

$$c(X) = \frac{1}{N_1 + N_2 + \cdots + N_K} \sum_{h=1}^{K} \sum_{j=1}^{N_h} \sum_{i=1}^{N_j} \left( t_{jhi} - f(x_{jhi}^T \bar{u}_i) \right)^2 \quad (2)$$

where $t_{jhi}$ stands for desired output. For sigmoid activation function, we choose $t_{jhi} = 1$ if $h = j$, and $t_{jhi} = 0$ if $h \neq j$. In multicategory case, we minimize $K$ terms

$$\sum_{h=1}^{K} \sum_{i=1}^{N_h} \left( t_{jhi} - f(x_{jhi}^T \bar{u}_i) \right)^2, \quad j = 1, 2, \ldots, K$$

of loss function (2) independently. In pairwise (binary) classification of classes $\Pi_i$ and $\Pi_j$, we search for a $(p + 1)$-dimensional weight vector $\bar{u}_{ij}$ taking into account training vectors of two pattern classes. If we used the result of minimization of loss (2) to perform binary classification of classes $\Pi_i$ and $\Pi_j$, we would have to employ $\bar{u}_{ij} = \bar{w}_i - \bar{w}_j$. While finding $\bar{w}_i$ and $\bar{w}_j$, training vectors of all other classes affect components of these two vectors. For that reason, discrimination of separate pairs of the classes by standard net of $K$ SLPs can become notably worse than that performed by individually trained pairwise perceptrons.

In Fig. 1(a), we demonstrate such situation with a 2-D example. Each single decision boundary (a line) is formed by outputs of two SLPs. In the 2-D feature space, three pairwise discrimination lines intersect at a point O. An implicit requirement concerning the intersection is a severe constraint. This restriction follows from the criterion of the sum-of-squared errors (2). Further, this restriction results in that at times the hyperplanes classify the pairs of classes unsatisfactorily. The area AOC is attributed to class $\Pi_1$. The area COB is attributed to class $\Pi_2$, and the remaining area BOA is attributed to class $\Pi_3$.

Consider a two-stage decision making procedure where one performs a pairwise classification in the first stage and assigns a class label to each single region formed by the pairwise classifiers. In Fig. 1(b), three pairwise SLP-based linear classifiers, 1(1/2), 2(1/3), and 3(2/3), split the 2-D space into seven regions. 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 $\Pi_1$ or $\Pi_2$), etc.

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

A proper initialization and early stopping are powerful tools to 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 sample mean vector of the pair of the classes is moved to a zero point, and training starts with initial weight vector composed of zero components, one obtains EDC 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, after suitable training, one may approach even the maximum margin
(SV) classifier [3], [17]. Successfully defined initial weight vector contains useful information which could be saved if to stop training in time [16]. If prior to training the perceptron, one would perform a whitening data transformation based on specially constructed sample estimates of CM of the input data, then after the first iteration, one would obtain valuable initial weight vector [3]. In principle, by prudent controlling parameters of the CM 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 employment of a validation set.

1) Use of a Noise Injection: One possible solution to bypass small sample size complication is to use all design set vectors as the training set and form a virtual validation set from the training vectors by means of a noise injection. A 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.

Injection of a white noise distorts the data geometry. A colored k-NN noise injection was suggested for reducing the data distortion [19]. To generate such a noise, for each single training vector $x_{js}$, we find its k-nearest neighbors of the same pattern class and add an artificially generated noise only in a subspace formed by the vector $x_{js}$ and k neighboring training vectors $x_{i1}, x_{i2}, \ldots, x_{ik}$. Random Gaussian $N(0, \sigma_{\text{noise}}^2)$ variables are added $n_{i\text{rand}}$ times along the k lines connecting vector $x_{js}$ and $x_{i1}, x_{i2}, \ldots, x_{ik}$ in $x_{i}$-vector noise injection procedure: 1) k, the number of neighbors, 2) $n_{i\text{rand}}$, the number of new, artificial vectors generated around each single training vector $x_{js}$, and 3) $\sigma_{\text{noise}}$, the noise standard deviation. Optimal values of these parameters depend on the data.

2) Support Vector Classifier: To make the first stage pairwise decisions we also used linear SV classifiers realized in LIBSVM [20]. We used default parameters proposed in LIBSVM, except the penalty parameter of the error term C and the class weighting parameter $\text{Cw}_{ij}$. Parameter C controls a complexity (in the sense of training sample size/complexity considerations) of the SV classifier. As recommended in [21], the C values were selected from set $[2^{-7}, 2^{-6}, \ldots, 2^{10}]$. To accomplish fair assessment of diverse two-stage and benchmark decision making schemes, we tried to equalize the number of adjustable global parameters of the methods.

The pseudovalidation data sets (see previous paragraph) were used to find the best C values for each of the pairs. We applied a simple and often used way [22] to define class importance parameters: $Cw_{ij}$ was proportional to $1/N_i$. Preliminary data transformations for SV- and SLP-based classifiers were performed in the same way: the input features were roughly decorrelated, and standard deviations of each them were set to 1 and means to 0.

C. Fusion of Pairwise Classifiers

In order to fuse the pairwise decisions, a number of methods have been developed. The voting rule does not require training. It classifies the $K(K - 1)/2$-dimensional vector according to the majority of class labels for this vector.

1) Hastie-Tibshiranie (H-T) Method: To classify the input vector $x_d$, one needs to estimate posterior probabilities $p_1, \ldots, p_K$ of the classes $\Pi_1, \ldots, \Pi_K$. The H-T method uses an information contained in $K(K - 1)/2$ conditional probabilities $\mu_{ijd} = \text{Prob}(x_z \in \Pi_i|x_d \in \Pi_j)$ or $x_d \in \Pi_j) = p_i/(p_i + p_j)$, $\mu_{ijd} = 1 - \mu_{ijd}$ and minimizes the sum of K-L distances between estimates $\hat{r}_{ij}$ and true probabilities $\mu_{ij}$. In the case of K pattern classes [7]

$$D_{\text{K-L}}(p_1, p_2, \ldots, p_K, x_d) = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} n_{ij} \left[ \hat{r}_{ij} \log \frac{\hat{r}_{ij}}{\mu_{ij}} + (1 - \hat{r}_{ij}) \log \frac{1 - \hat{r}_{ij}}{1 - \mu_{ij}} \right]$$

subject to $\sum_{i=1}^{K} p_i = 1$, $p_i \geq 0$.

The probabilities $\mu_{ijd}$ and $\mu_{ijd}$ are conditioned to the concrete vector $x_d$ and classes $\Pi_i, \Pi_j$.

We assume that in high-dimensional cases densities of the weighted sums $g(x_d) = x_d^T + \hat{n}_{ij}$ and $\hat{n}_{ij}$ are Gaussian, $\phi(g(x_d)|m_{ij}, s_{ij})$, with means $m_{ij}, m_{ij}$, and standard deviations $s_{ij}, s_{ij}$. Then

$$\mu_{ijd} = \frac{\phi(g(x_d)|m_{ij}, s_{ij})}{\phi(g(x_d)|m_{ij}, s_{ij}) + \phi(g(x_d)|m_{ij}, s_{ij})}. \quad (3)$$

In the standard version [6], $\hat{n}_{ij}^R, \hat{n}_{ij}^R$, and $\hat{s}_{ij}, \hat{s}_{ij}$ are training-set-based means and standard deviations of $N_i + N_j$ values $g(x_d)$. We will also estimate these parameters on the pseudovalidation or test sets in several experiments. The estimates of pairwise conditional probabilities are

$$\hat{p}_{ij} = \frac{\phi(g(x_d)|\hat{n}_{ij}^R, \hat{s}_{ij}^R)}{\phi(g(x_d)|\hat{n}_{ij}^R, \hat{s}_{ij}^R) + \phi(g(x_d)|\hat{n}_{ij}^R, \hat{s}_{ij}^R)}, \quad (4)$$

subject to $\sum_{i=1}^{K} p_i = 1$, $p_i \geq 0$.

2) Wu-Lin-Weng (WLW) Sum-of-Squares Method: Wu, Lin, and Weng introduced two new algorithms for finding $p_i$ (note that $p_i = \phi_i(g(x_d)|m_{ij}, s_{ij})$ in the Gaussian distribution case) and $\mu_{ijd}$. We have chosen the second algorithm, since in empirical comparisons [7], it worked better in 11 cases out of 14. This algorithm minimizes the sum-of-squared differences between the weighted estimates of the pairwise conditional probabilities

$$D_{\text{WLW2}}(p_1, p_2, \ldots, p_K, x_d) = \sum_{i \neq j} (p_i \hat{r}_{ij} - p_j \hat{r}_{ij})^2$$

subject to $\sum_{i=1}^{K} p_i = 1$, $p_i \geq 0$. \quad (6)

To find $p_1, \ldots, p_K$ for each $x_d$, we used an algorithm developed in [7].
3) The Fuzzy Templates (FT) Method [13]: It is aimed at fusing continuous outputs of several classifiers. We adapted this algorithm for $L = K(K - 1)/2$ pairwise classifiers. First, the fuzzy template of class $\Pi_k$ was converted into a vector $F_i = \{f_i(l)\}$ with $K - 1$ attributes, where

$$f_i(l) = \frac{\sum_{m=1}^{N_k} C_{m,n}(x_d)}{N_k}$$

(7)

for all $l = 1, 2, \ldots, K - 1$ classifiers $C_{m,n}$; $m = i$ or $n = i$. In the description of FT method, $\{x_d\}$ are crisply labeled training vectors and $C_{m,n}(x_d)$ are outputs of the pairwise classifiers. Only the classifiers with

$$(i, j) = (1, 2), (1, 3), \ldots, (K - 1, K)$$

(8)

are considered, because $C_{i,j}(x_d) = 1 - C_{j,i}(x_d)$. So, for each pattern class, we have $K - 1$ associated pairwise classifiers. The components of the FT vector were ordered as described in (8). Then, having the vector $x_d$, we calculated its decision profiles for each class: $DP_m(x_d) = \{C_{m,n}(x_d)\}$, where $m = i$ or $n = i$, ordered as described in (8).

**Final decision was made according to**

$$\max_i(S(F_i, DP_m(x_d)))$$

where

$$S(F_i, DP_m(x_d)) = 1 - \frac{1}{K - 1} \sum_{l=1}^{K-1} (f_i(l) - DP_m(x_d))^2.$$  

(9)

4) Directed Acyclic Graph (DAG) Method [11]: The idea of this method is to organize the pairwise SV classifiers into a rooted binary directed acyclic graph for making the final decision. The decision making process in DAG may be considered as some sort of the decision tree. When an unknown vector is submitted for classification, first it is 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 the $K$ nodes with no child is reached. This node labels the new vector. In our experiments (Section IV-D), we also used the binary SLPs instead of the SV classifiers. In such a way, we generalized the DAG algorithm to other types of the pairwise classifiers.

D. Two Benchmark Multiclass Allocation Rules

1) Kernel Discriminant Analysis (KDA): Here, we apply nonparametric kernel-based local estimates of conditional probability density functions of input vectors $f_{KDA}(x|\Pi_k)$. In classification phase, independent decisions are performed in each point of the feature space [3], [14]. We used the Gaussian kernel and classified according to the maximum of products

$$q_i \times f_{KDA}(x|\Pi_k) = \frac{q_i}{N_i} \sum_{j=1}^{N_i} \exp(-h^{-1}(x - x_{ij})^T(x - x_{ij})),$$

$$i = 1, 2, \ldots, K, \quad j = 1, 2, \ldots, N_i$$

(10)

where $h$ is a smoothing parameter. To have truthful comparison of the KDA-based algorithm with the other methods, after training-set-based decorrelation of the data, we normalized standard deviations $s_i$ of the features at first $(s_i \approx 1, 0)$, and used default value $h = 1.0$.

2) The Radial Basis Function (RBF) Neural Network: The RBF-based classifier consists of three layers: input layer, a hidden radial basis function layer, and a linear output layer [1]–[3], [14]. Radial basis layer is composed of $G$ radial basis neurons that calculate $y_i = \text{rad}(\|C_{i1} - x\|/H_i)$, $i = 1, \ldots, G$, $\text{rad}$ is a transfer function for radial basis neuron (in our experiments, we used the model of multivariate Gaussian distribution), $C_{i1}$ is the $i$th “center” of the radial basis neuron, and $H_i$ is the smoothing parameter. Output layer is linear: $o_k = w_{k2}y + b_{k2}$, where $y = (y_1, \ldots, y_G)^T$, $w_{k2}$ is the weight vector, and $b_{k2}$ is the bias term. The newly classified vector $x$ is classified according to the maximum of outputs. We used the Matlab neural network (NN) toolbox to form the RBF networks. We used artificial pseudovalidation sets to select parameters $g$ and $H_i$.

E. Experimental Comparison

1) Data: Six real-world data sets were used for 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] describes three pattern classes of Iris plants by their sepal and petal size (50 4-D vectors in each class). The Realty data set comprises 392 17-D vectors describing constructional, ecological, and market characteristics of realty. The data were grouped into three categories (118, 160, and 94 vectors) by experts and was used in an econometric analysis in one private Lithuanian company. The Satimage data set [23] describes multispectral values of pixels in a satellite image. The six classes contain 1072, 479, 961, 415, 470, and 1038 36-D vectors, respectively. The Wheat data set describes 12 geometrical wheat grain measurements (see [24]). This data set contains five pattern classes, with 80 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. Originally we had 1500 spectral features. In order to reduce dimensionality, we employed a ten-class Euclidean distance classifier. The 10-D space of ten EDC outputs formed nine discriminative features. Before training, all data sets were normalized and rotated according to their eigenvalues [3].

2) Procedure: To obtain reliable estimates, the experiments were carried out 250 x 2 times: after reshuffling each class of data separately, a half of vectors were used for training, and the other half for testing. In subsequent trial, the training and testing data sets were interchanged. Such a procedure was repeated $N_c = 250$ times (only $25 \times 2$ experiments were performed with Chromosome data set). Prior to training the classifiers, we normalized the data by standard deviations of each single feature.
Then, we used principal components and eigenvalues of pooled sample CM to transform the data prior to train the perceptrons and SV classifiers. Moreover, prior to training the pairwise classifiers, each time, we moved two-class mean vectors to zero point. To determine stopping moments of SLPs and regularization parameters $C$ of SV classifiers we used pseudovalidation sets. Based on our experience, “default” values were used: $k = 2$, $\sigma_{\text{noise}} = 1.0$, $m_{\text{ini}} = 2$.

3) Results: In Table I, we present averages of generalization errors of three standard $K$-category classifiers and five two-stage algorithms (in the first stage of decision making we used linear SV as the pairwise classifiers). The last column (marked as $\sigma_N$) shows standard deviations $\sigma$ of the fusion methods with smallest error rate (printed in bold) divided by $\sqrt{N_c}$ ($N_c = 250$; $N_c = 25$ for the Chromosome data).

We see that in all the six PR tasks, the $K$-category net of SLPs were outperformed by other techniques. No method proved to be best. 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 four PR tasks. This means that the efficiency of the methods highly depends on the data.

III. EFFECT OF SIMPLIFIED PERFORMANCE MEASURES AND SAMPLE SIZE ON FUSION ACCURACY OF THE PAIRWISE CLASSIFIERS

In the H–T fusion method, we minimize the sum of $K$–L distances (3) between the estimates $\hat{p}_{ij}$ and true probabilities $\mu_{ij}$. In the WLW method, we employ the sum-of-squares differences of pairwise conditional probabilities (6). In the FT method, other sum-of-squared differences (9) is used. Two sources of errors affect the accuracy of the performance estimates:

1) the simplified performance measures are related to classification error only approximately;

2) the classifiers are based on the training data, while for estimating of the generalization error we use the test data.

The complexities of the SVs and SLPs used in our study as the linear pairwise classifiers are governed by 1) regularization parameter $C$ or 2) the number of training epochs. Positive feature of the SLPs is the fact that the complexity of a classifier gradually increases during its training process [3], [17]. In this way, SLP may adapt its complexity to particular design data set. Artificial pseudovalidation data sets generated by means of a noise injection were used to find best values of the parameters mentioned.

A. Analysis of the Single Learning Set

The difference between 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, we have to take into account the complexity of the PR task and effects of high dimensions. Hence, we generated three-category 50-D Gaussian data. Mean vectors and covariance matrices of first 17 features were taken from the Realty data set. To care about the higher dimensional situations, we added 33 noninformative features with low dispersions $\sigma^2 = 0.1^2$ (standard deviations of the first two features were approximately 1).

Due to nonlinearity of activation function, exact nonasymptotic analysis of learning dynamics is impossible. For that reason, we investigated a random search optimization procedure. We used relatively small learning sets ($N_3 = N_2 = N_1 = N = 50$). In 600 independent runs of the experiment, we generated 600 random learning sets $S_{l, \text{Lwin}}$, $S_{l, \text{Lwin}}$, $S_{l, \text{Lwin}}$, $S_{l, \text{Lwin}}$, $S_{l, \text{Lwin}}$ of size $3 \times N$ and used them to train 600 triplets of pairwise SLPs. As a result, we obtained 600 $3 \times 51$-dimensional matrices of the weights. To achieve accurate estimates of the classification error rates, we used a large test set $S_{\text{Test}} (n_3 = n_2 = n_1 = 2000)$.

To find parameters of 600 H–T fusion rules in the first experiment, we generated an extra learning set $S_{l, \text{Fusion}}$ ($50 \times 3$ vectors). This set was used to estimate sample means $\bar{m}_{l, \text{Fusion}}$, $\bar{m}_{l, \text{Fusion}}$, $\bar{m}_{l, \text{Fusion}}$, and standard deviations $s_{l, \text{Fusion}}$, $s_{l, \text{Fusion}}$, $s_{l, \text{Fusion}}$. Each time, 600 K–L distances $D_{l, \text{Fusion}}$ were estimated from set $S_{l, \text{Fusion}}$. A single test set $S_{\text{Test}}$ was used to estimate generalization errors $P_{\text{Test}} (t = 1, 2, \ldots, 600)$. In Fig. 2(a), we have a scatter diagram of the distribution of 600 vectors ($D_{l, \text{Fusion}}$, $P_{\text{Test}}$) ($t = 1, 2, \ldots, 600$), obtained after minimizing the K–L distances evaluated from learning set.

Each triplet corresponds to one of the 600 points in 2-D scatter diagram in Fig. 2(a). Here estimation of K–L distances was based on learning set and the generalization errors were evaluated from test set. We see that the generalization errors and K–L distances are weakly correlated ($\rho = 0.263$). The low correlation, observed in Fig. 2(a), is caused by factors 1) and 2) that were mentioned at the beginning of Section III.

B. The Second Experiment

In order to elucidate the effect of factor 1), we have to eliminate the influence of factor 2). In order to get rid of inexact estimation due to finiteness of sample size, we carried out an artificial experiment: the test set vectors were used to estimate the means $m_{l, \text{Fusion}}$, $m_{l, \text{Fusion}}$, $m_{l, \text{Fusion}}$ and standard deviations $s_{l, \text{Fusion}}$, $s_{l, \text{Fusion}}$, $s_{l, \text{Fusion}}$ [Fig. 2(b)]. In this case, the generalization errors and K–L distances are considerably more correlated ($\rho = 0.757$). This fact hints that low correlation is evoked due to inexact parameter estimation from small samples. This experiment confirms that in the 50-dimensional PR task with relatively small sizes of the learning set ($N = 50$), the influence of finiteness of the sample size is more essential factor than the influence of differences between the generalization errors and K–L distances.

The effect of finiteness of sample size may be proved more forcefully after a scrupulous analysis of the scatter diagrams in Fig. 2. In realistic model selection procedures we are selecting

---

<table>
<thead>
<tr>
<th>Meth.</th>
<th>Chrom</th>
<th>Iris</th>
<th>Realty</th>
<th>Satim</th>
<th>Wheat</th>
<th>Yeast</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSLP</td>
<td>0.262</td>
<td>0.097</td>
<td>0.078</td>
<td>0.177</td>
<td>0.088</td>
<td>0.136</td>
</tr>
<tr>
<td>KDA</td>
<td>0.270</td>
<td>0.033</td>
<td>0.074</td>
<td>0.214</td>
<td>0.098</td>
<td>0.131</td>
</tr>
<tr>
<td>RBF</td>
<td>0.202</td>
<td>0.031</td>
<td>0.053</td>
<td>0.149</td>
<td>0.073</td>
<td>0.190</td>
</tr>
<tr>
<td>+Tot.</td>
<td>0.195</td>
<td>0.030</td>
<td>0.056</td>
<td>0.147</td>
<td>0.069</td>
<td>0.149</td>
</tr>
<tr>
<td>+H-T</td>
<td>0.193</td>
<td>0.030</td>
<td>0.056</td>
<td>0.144</td>
<td>0.067</td>
<td>0.146</td>
</tr>
<tr>
<td>+WLW</td>
<td>0.193</td>
<td>0.030</td>
<td>0.056</td>
<td>0.145</td>
<td>0.067</td>
<td>0.147</td>
</tr>
<tr>
<td>+Fuzz.</td>
<td>0.197</td>
<td>0.029</td>
<td>0.055</td>
<td>0.144</td>
<td>0.069</td>
<td>0.144</td>
</tr>
<tr>
<td>+DAG</td>
<td>0.197</td>
<td>0.030</td>
<td>0.056</td>
<td>0.148</td>
<td>0.069</td>
<td>0.150</td>
</tr>
<tr>
<td>$\sigma_N$</td>
<td>0.0008</td>
<td>0.0010</td>
<td>0.0011</td>
<td>0.0004</td>
<td>0.0008</td>
<td>0.0011</td>
</tr>
</tbody>
</table>
the best model according to approximate measures, the K–L distances. Fig. 2(a) shows that we have to choose the classifier A. Generalization error of model A is $P_{genA} = 0.04$. A hypothetical procedure, an ideal model selection, is performed according to test set estimates of the generalization error. In ideal model selection, the best model (classifier) is B with $P_{genB} = 0.034$. Thus, due to usage of imperfect model selection measure, the generalization error increased by $\Delta P_{genL} = 0.04 - 0.034 = 0.006$. Similar analysis of Fig. 2(b) shows that this time (here the influence of the finiteness of sample size was artificially eliminated), the generalization error increased 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, difference $\Delta P_{genT} = P_{genA} - P_{genB} = 0.011$ is much smaller. The comparison of $\Delta P_{genL}$ and $\Delta P_{genT}$ confirmed that the finiteness of the sample size was the main factor in this particular simulation study.

The effect of inexactness of K–L criteria is arising together with finiteness of the sample size. In hypothetical situations with very large sample sizes, the only one source of errors would be the inexactness of performance measure. With an increase in sample size, difference $P_{genA} - P_{genB}$ declines. This difference is approaching a certain constant that depends on the accuracy of performance measure. The decrease rapidity depends on the data and the accuracy of model selection criterion. We do not point toward such type of research in this paper. For the cross-validation error counting classification error estimate, such investigation has been done analytically (see [3, Sec. 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, we considered all possible selections of $m$ models out of $M = 600$ ones. In case of two models ($m = 2$), selected out of $M = 600$ ones, we can form $r = M!/(m! \times (M - m)!) = 179700$ selections. In case of $m = 10$, $r \approx 1.5453 \times 10^{21}$. In Fig. 3, we present averages of true generalization errors on the number $m$ (triplets of the pairwise SLP classifiers). The fusion rules were based on learning set $S_{LPfusion}$ data (1-SV classifiers were used, solid line marked by squares; the default value $C = 1$ was used this time), 3-SLPs). Curves 2 (SV classifiers, dotted line marked by circles) and 4 (SLPs, 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 the Pikelis combinatory equations [3, App. A4].

The differences between the “true” (upper) and “ideal” (lower) curves characterize an increase in the generalization error when the fusion rules were based on inaccurate estimates of the classification error rate. The differences between points A and B, obtained from data as depicted in Fig. 2(a), also were much higher than that of the data in Fig. 2(b). These facts corroborate once again that for K–L criterion, the data dimensionality, and the learning sample sizes investigated, 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.

The generalization errors obtained by using fusion rules based on the K–L distances (H–T method) and the WLW sums in fact were the same. Fig. 4 plots the generalization errors
obtained from the WLW sum of squares (x-axis) versus that from K–L distances (y-axis).

The fusion rules were designed either on the training sets (points scattered on the upper line) or the test sets (points scattered on the lower line; for display clarity lower points were shifted by 0.001 to the right). This conclusion agrees with the results reported in Table I where we also observed that the H–T and WLW methods gave almost the same accuracy. So, the conclusions that followed from Figs. 2 and 3 also are valid for WLW estimates and SV- or SLP-based pairwise rules (Section IV-D).

The curves $P_{gen} = f(m)$ often “peak” with an increase in the number of models $m$. In Fig. 3, we watched the overfitting phenomenon in the experiments where SLPs were used as the pairwise classifiers. Peaking behavior is widespread in tasks where optimization is based on inexact criteria. An origin of this behavior is the same as in feature overselection [25], overtraining of NNs, where the training process is based on minimization of the learning-set-based cost function (see [3, Sec. 4.5 and 6.5]). The inexactness of the K–L distance and WLW sum measures, the difference of the cost function (2), and empirical classification error actually are the causes of the correspondence of effectiveness of H–T and WLW methods, and the peaking phenomena.

IV. THE UNBALANCED SAMPLE SIZE/COMPLEXITY ISSUES

A. SINGLE-STAGE K-CATEGORY SLP-BASED NEURAL NET FOR CLASSIFICATION

In Section II, we witnessed nonoptimality of the multicategory neural network in the cases where learning set sizes were sufficiently large. Additional difficulties arise if design sample sizes are small and imbalanced [4]. In the asymptotic learning theory, the generalization error is related to the classifier’s complexity and the size of the training set $n = N_1 + N_2 + \cdots + N_K$ [26], [27]. The use of a solitary sample size parameter $n$ is appropriate if we use the optimal sample-based classification rules. Nevertheless, classification rules without theoretical proof of their optimality are often applied in practice. It has been shown analytically that 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], [28]. No theoretical results exist for multicategory situations.

As an illustration, in Fig. 5, we present experimentally evaluated relationships between the sample size $N_3$ and the generalization error of the Fisher linear DF (F) and the three-category network of SLPs in a case where $N_2 = N_3$. The 50-D three-category Gaussian classes, considered in Section III, were used. The curves are averages of 1000 runs of the experiment, where $N_2 = N_1 = 50$, and test set sizes $n_1 = n_2 = n_3 = n_{test} = 2000$ (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 $50 \times 50$-dimensional CM 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 $N_3$, we also observe a similar behavior of the net of SLPs. The red solid curve 1 marks the result obtained for the optimal number of training epochs (in this experiment, $n_{test} = 2000$; so, generalization errors were evaluated rather exactly), and curve 2 marks the result after optimal stopping determined by the artificial pseudovalidation set. When $N_3$ approaches $N_1 = N_2 = 50$, the generalization error of the network diminishes up 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 nonoptimally designed classification rule depends also on the balance of the data sizes. Curves 1 and 2 demonstrate that generalization error increases more than two times if $N_3 \rightarrow 1000$.

A question arises: Can we avoid this undesirable effect of the standard $K$-category network? To find an answer, we need to
pay an attention to the cost function (2). Theoretically, while increasing \(N_3\), we aim at evaluating the third class influence on the cost function more precisely. If \(N_3\) is increased two times, the contribution of the third class training vectors increases two times. 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, we need to restore prior probabilities of each class. Hence, we introduce the unbalance correcting terms \(q_i/N_i\). Instead of cost (2), we use a modified variant
\[
\text{cost} = \sum_{h=1}^{K} \sum_{i=1}^{K} q_i \sum_{j=1}^{N_i} \left[ h_{i,j} - f(x_i^h w_i + w_{h0}) \right]^2. \tag{11}
\]

The modified cost is important when values \(q_i\), genuine prior probabilities of the classes, are not proportional to \(N_i\). In Fig. 5 lower curves, 3 (ideal stopping, based on generalization errors estimated from the test set) and 4 (pseudovalidation-set-based stopping), show the generalization error of “the modified network of \(K\)-category SLPs.” We see that the generalization error curves almost stop increasing with an excessive increase in \(N_3\). Moreover, if we stop training at the right time, we achieve a notable improvement even if \(N_3\) is very small.

B. Complexity of Pairwise SV- and SLP-Based Classifiers

1) Generalization Error of the Fisher Classifier: In the two-stage approach, we acquire the possibility to choose the feature subset and the type of classifier that is individual for each pair of the classes. In this context, the perceptron’s quality that “while training the nonlinear SLP, the classifier’s complexity is gradually increasing” (see Section II) becomes very attractive. We present a theoretical justification of this opinion by analyzing a hypothetical situation, where the classes are multivariate Gaussian 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. We consider this classifier because during one of the phases of its evolution, the SLP-based classifier behaves like the Fisher DF. The generalization error of binary Fisher DF may be calculated by following asymptotic formula [3, 27]:

\[
EP_N^F \approx q_i \Phi \left\{ -\frac{1}{2} \delta_{ij} T_{mi/ji} T_{Se} \right\} + q_i \Phi \left\{ -\frac{1}{2} \delta_{ij} T_{mi/ji} T_{Se} \right\} \tag{12}
\]

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

\[
T_{mi/ji} = \frac{1 - \frac{p}{(N_i \delta_{ij})} + \frac{p}{(N_j \delta_{ij})}}{\sqrt{1 + \frac{p}{(N_i \delta_{ij})} + \frac{p}{(N_j \delta_{ij})}}}
\]

\[
T_{mi/ji} = \frac{1 + \frac{p}{(N_i \delta_{ij})} - \frac{p}{(N_j \delta_{ij})}}{\sqrt{1 + \frac{p}{(N_i \delta_{ij})} + \frac{p}{(N_j \delta_{ij})}}}
\]

\[
T_{Se} = \sqrt{\frac{N_i + N_j - p}{N_i + N_j}} \tag{13}
\]

\(\delta_{ij}\) is a Mahalanobis distance between pattern classes \(\Pi_i\) and \(\Pi_j\)

\[
\delta_{ij} = ((\Delta_{ij})^T \Sigma^{-1} \Delta_{ij})^{1/2} \tag{14}
\]

\(\Delta_{ij} = m_i - m_j\), \(m_i\), \(m_j\) are mean vectors of the classes, and \(\Sigma\) is their “common” CM.

Terms \(T_{mi/ji}\) and \(T_{mi/ji}\) in (12) emerge due to inexact sample estimation of the mean vectors of the classes. Term \(T_{Se}\) appears due to inexact sample estimation of the CM that is supposed to be common to \(\Pi_i\) and \(\Pi_j\). In EDC design, we do not estimate the CM. Hence, for EDC, we have to omit term \(T_{Se}\) in (12). In the latter case, standard Mahalanobis distance (14) no longer determines the asymptotic probability of misclassification. Instead, we have an “effective distance” and “effective dimensionality” [3, Ch. 3]

\[
\delta_{ij}^e = ((\Delta_{ij})^T \Sigma^{-1} \Delta_{ij})^{1/2}
\]

\[
p_{ij}^e = ((\Delta_{ij})^T \Delta_{ij})^{1/2} \tag{15}
\]

The Mahalanobis distance \(\Delta_{ij}\) and effective parameters \(\delta_{ij}^e\), \(p_{ij}^e\) are specific to each pair of the classes. Usually, \(\delta_{ij}^e < \delta_{ij}\) and \(1 < p_{ij}^e < \infty\) [3].

2) Case \(N_2 = N_1\): If sample sizes \(N_2 = N_1 = N\) and \(p_2 = p_1 = 0.5\), expressions (13) become more simple. In the following illustration, we used the first \(p = 30\) features of the modified three-category Gaussian data considered in the previous examples (see the beginning of Section III). Due to particularities of the real-world Realty data set used to construct the artificial data, three classes are located approximately on an arched curve (see Fig. 1). Thus, for the class pairs \(\Pi_1\), \(\Pi_2\) or \(\Pi_2\), \(\Pi_3\), the asymptotic classification errors of the Euclidean distance and Fisher classifiers \(p_{\infty}^E\) and \(p_{\infty}^F\) are almost equal. To increase the difference between asymptotic errors of EDC and Fisher classifiers, we diversified covariance matrices: \(\Sigma_{\text{new}} = \Sigma_4/\theta, \Sigma_{2\text{new}} = \Sigma_2 \times \theta\) in novel experiment. In calculations, we used \(\Sigma = 1/2(\Sigma_{\text{new}} + \Sigma_{2\text{new}})\). Table II shows parameters \(\delta_{ij}, \delta_{ij}^e, p_{ij}^e\), and \(p_{\infty}^E, p_{\infty}^F\). In our computational example, effective dimensionality \(p_{12}^e = 1.90\). It is much smaller than \(p = 30\), formal dimensionality of the data. This fact advocates that sensitivity of EDC to training set size is low in this particular case. In small learning set situations, however, generalization error rate of Fisher DF is much higher as the asymptotical one [curves 1 and 2 (dots) in Fig. 6].

In Fig. 6, we also present generalization errors (evaluated experimentally in 1000 runs) of EDC, the pseudovalidation set stopped SLP (squares) and SV classifier (crosses) with parameter \(C\) evaluated using generalization errors estimated from pseudovalidation set.
This data model is unfavorable for SV classifier. Exploitation of 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 pseudovalidation set stopped SLP was the best classifier.

3) Case $N_2 \neq N_1$. When prior probabilities of the classes are different and the training set sizes are imbalanced, terms $p/(N_2 \xi_{i,j}^2)$, $p/(N_1 \xi_{i,j}^2)$ in (13) and nonlinearity of cumulative distribution function $\Phi(\alpha)$ in (12) cause that EDC and Fisher classifiers are not optimal. The nonoptimality is inherent also to the SW- and SLP-based classifiers.

While analyzing the imbalance problem in the multicategory situation (Section IV-A), we have seen that with an increase in the training set size of one of the classes, the generalization error of the Fisher DF diminishes permanently. Consider now a two category PR task, where the true learning set sizes $N_1$, $N_2$ do not reflect $q_1$ and $q_2$. We assume that $q_2 = 0.75$, however, $N_2$ is varying. The total learning set size remains unchanged: $n = N_1 + N_2 = 100$.

In Fig. 7, we present generalization errors for the class $\Pi_1$, $\Pi_2$ pair ($\Sigma_2 \neq \Sigma_1$ here) as functions of $N_2$ for EDC, Fisher DF, SV, and the modified SLP-based classifiers [see (11)]. We plot averages of 1000 experimental runs performed with the data model considered in previous experiment. This time, the covariance matrices diversity parameter $\theta = 0.4$ (Table II). The optimally stopped SLP-based classifier was almost insensitive to imbalance of $N_1$, $N_2$. In all cases, it was the best choice to apply in the classification task. If instead of a priori probabilities we used ratios of $N_i$ to total sample size $n$, classification results were worse.

C. Bias Reduction in Fusion of Pairwise Decisions

The training set based (resubstitution) estimates of the classification error rate are optimistically biased [3], [14], [27]. For that reason, usage of the training-set-based resubstitution estimates $\hat{\theta}_{i,j}^R$, $\hat{\theta}_{j,i}^R$ of $\mu_{i,j}$, $\mu_{j,i}$ and $\hat{\delta}_{i,j}^R$, $\hat{\delta}_{j,i}^R$ of $\delta_{i,j}$, $\delta_{j,i}$ can lead to incorrect estimates $\hat{\theta}_{i,j}^R$ and $\hat{\theta}_{j,i}^R$ of probabilities $\mu_{i,j}$ and $\mu_{j,i}$. The overadapted estimates worsen the H–T and WLW fusion rules. In an attempt to investigate a possibility to improve fusion rules, let us consider a theoretical way to reduce the optimistic bias of estimates $\hat{\theta}_{i,j}^R$ and $\hat{\theta}_{j,i}^R$ when binary Fisher classifiers are used. Denote

$$\hat{\theta}_{i,j}^N = \frac{\hat{m}_{i,j}^R}{\hat{s}_{i,j}^R}, \hat{\theta}_{j,i}^N = \frac{\hat{m}_{j,i}^R}{\hat{s}_{j,i}^R}. \tag{16}$$

Distances $\hat{\delta}_{i,j}^N$ and $\hat{\delta}_{j,i}^N$ characterize pairwise resubstitution classification error estimates of pair $\Pi_i$, $\Pi_j$. Expected values of sample Mahalanobis distances are optimistically biased since

$$\hat{E}_{i,j}^N = \hat{\delta}_{i,j}(T_{\mu_{i,j}}^N T_{\Sigma}) \geq \hat{\delta}_{i,j}. \tag{17}$$

In (17), terms $T_{\mu_{i,j}}$, $T_{\mu_{j,i}}$, and $T_{\Sigma}$ have been defined by (13), (14), and (15). We remind that, for each pair $\Pi_i$, $\Pi_j$, specific distances should be used. Suppose that for pair $\Pi_i$, $\Pi_j$, $\hat{\delta}_{i,j}^N$, and $\hat{\delta}_{j,i}^N$ values were already calculated from training data. Then, we may obtain “unbiased” estimates $\tilde{\delta}_{i,j}^N$ and $\tilde{\delta}_{j,i}^N$ of distances $\delta_{i,j}$ and $\delta_{j,i}$, respectively. To do this, we have to apply (17) and interpolate the result in a certain interval of $E_{i,j}^N$ values. Having the estimates of $\tilde{\delta}_{i,j}^N$ and $\tilde{\delta}_{j,i}^N$, we could use relationship (12) in order to calculate generalization errors

$$P_{i,j} = \Phi \left\{ -\frac{1}{2} \tilde{\delta}_{i,j}^N \right\} \quad \text{and} \quad P_{j,i} = \Phi \left\{ -\frac{1}{2} \tilde{\delta}_{j,i}^N \right\}$$

where $\tilde{\delta}_{i,j}^N / (T_{\mu_{i,j}}^N T_{\Sigma})$ and $\tilde{\delta}_{j,i}^N = \delta_{j,i} / (T_{\mu_{j,i}}^N T_{\Sigma})$.

Analogously as in (16), we express the distances $\hat{\delta}_{i,j}^N$ and $\hat{\delta}_{j,i}^N$ as fractions

$$\hat{\delta}_{i,j}^N = \frac{\hat{m}_{i,j}^N}{\hat{s}_{i,j}^N}, \hat{\delta}_{j,i}^N = \frac{\hat{m}_{j,i}^N}{\hat{s}_{j,i}^N}$$

where $\hat{m}_{i,j}^N = \delta_{i,j}^N \hat{s}_{i,j}^N$, and $\hat{m}_{j,i}^N = \delta_{j,i}^N \hat{s}_{j,i}^N$. 

\[Fig. 6. \] Generalization errors as functions of sample size: 1—Fisher DF (experiment), 2—Fisher DF (theory, (12), (13)), 3—SV classifier, 4—EDC classifier, 5—pseudovalidation set stopped SLP.

\[Fig. 7. \] Generalization errors as functions of sample size: 1—Fisher (experiment), 2—Fisher DF (theory, (11), (12)), 3—SV classifier, 4—EDC classifier, 5—pseudovalidation set stopped novel SLP.
Final Result: After the above manipulations, we obtain unbiased (corrected) estimates of conditional a posteriori probabilities $\mu_{ij|d}$ and $\mu_{ij|d}$ to be used instead of (4) in the H–T fusion rule design

$$\hat{\mu}_{ij|d} = \frac{\phi(g(x_d)|\mu_{ij|d}, \hat{y}_{ij|d})}{\phi(g(x_d)|\mu_{ij|d}, \hat{y}_{ij|d}) + \phi(g(x_d)|\mu_{ij|d}, \hat{y}_{ij|d})}$$

$$\hat{\mu}_{ij|d} = 1 - \hat{\mu}_{ij|d}$$

A similar method could be used to obtain unbiased (corrected) estimates of conditional a posteriori probabilities $\mu_{ij}$ and $\mu_{ij}$ for the Euclidean distance pairwise classifiers. Instead of $\delta_{ij}$ and $p$, one has to use $\delta_{ij}^2$ and $p_{ij}^{\text{corr}}$ defined in (15), and omit term $T_E$.

Apart from the theoretical methods, an extra pseudovalidation set to evaluate the overadaptation bias of the H–T parameters $m_{ij|d}$, $m_{ij|d}$ and $s_{ij|d}$, $s_{ij|d}$ also may be used. Artificial 50–D Gaussian data (see Section III–A) was used to perform a simulation study aimed to verify usefulness of following bias correction methods.

1) The EDC analytically based correction of the H–T parameters [formally, we assume that the pairwise classifiers are EDC ones ($E$ correction)].

2) The same estimation method as the $E$ correction, however, this time the Fisher DFs were supposed to be used as the pairwise classifiers ($E$ correction).

3) The estimates of the H–T parameters were obtained from the first pseudovalidation data V1 already used for determination of the SLP’s optimal stopping moment (V1 correction), and

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

As a benchmark method we used standard training-set-based H–T procedure. To verify potential abilities of the bias elimination, we used an “ideal” H–T procedure where parameters (5) would be evaluated exactly. For this purpose, we designed fusion rules, where the H–T parameters were estimated on a very large test set composed of 10 000 vectors. In Table III, we show average values of the classification errors obtained in $N_e = 1000$ independent cross-validation trials. Parameter $C$ of SV and stopping of SLPs were accomplished on a basis of generalization errors estimated on pseudovalidation sets generated with parameters $m_{ij|d} = 50$, $k = 2$, and $\sigma_{\text{noise}} = 1.0$.

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 pairwise binary classifiers. The empirical methods became the winners. 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 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%. Though nominal increase is not considerable, the relative increase is worth of attention. The SV-based scheme was improved much more than the SLP-based scheme. 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” (Section II-B).

### D. Results With Pairwise SLPs

In the experiments with SLP-based pairwise classifiers, we used the same $N_e = 250$ data reshufflings as in the experiments with SV classifiers (Table I). The training sets were employed to generate pseudovalidation sets. In Table IV, we present average values of generalization errors, standard deviations of H–T method, $\sigma_{\text{HT}}$, and standard deviations of the averages $\sigma_{\text{HT}} / \sqrt{N_e}$ (marked as $\sigma_{\text{HT}}^*$).

Most often the best fusion rules were created when using H–T and WLW fusion. In many cases, both methods gave almost identical classifications, similarly as in the experiments with the 50–D Gaussian data (see Section III). Almost in all the experiments, after successful selection of the fusion rule, the employment of SLP-based pairwise classifiers was more beneficial than use of the SV ones. The experiment with the $K = 24$ class Chromosome data set and small number of cross-validation trials ($N_e = 25$) was a minor single exception. Empirical study revealed that while designing the two-stage multiclass category classifiers, based on the pairwise decisions, a major attention should be paid to the performance of the base classifiers.

Usually, the differences between modern decision making schemes are rather small. Therefore, it is worth to stress for a second time that the experimental evaluations highly depend on concrete split of available data into training and test sets. For
that reason, \( N_c = 20 \) or 100 cross-validation trials with reshuffled data frequently are insufficient to reveal the differences between the methods reliably. Accordingly, we have to perform much more reshufflings.

V. CONCLUDING REMARKS

The standard cost function of the multicategory nets of SLPs does not minimize the classification error directly. Habitually, it does not allow obtaining optimal pairwise classifiers even in the cases, where prior probabilities of the classes are the same and the training sets are balanced. The only classifiers aimed for obtaining decision making rules that minimize the classification error rate involve methods based on statistical decision theory. To allow minimization of the classification error rate in multiclass neural network training, we attempted to improve the two-stage decision making scheme, where in the first phase, we perform the pairwise classifications, and in the second one, we fuse the pairwise decisions. The main results are itemized below.

1) The two-stage neural-network-based decision making procedures may outperform the single-stage ones because they: a) refuse the traditional \( K \)-class cost function, b) allow obtaining near to optimal pairwise linear classifiers by specially organized SLP or SV training, and c) can save useful discriminative information contained in the first stage classifiers by prudent fusion of the pairwise decisions. Introduction of special correcting terms into the traditional cost function reduces the classification error rate in unbalanced training set size situations.

2) Two fusion strategies, the K–L distance based and the WLW method, result in approximately the same performance in situations where the training sample sizes are small. This actuality may be explained by the theoretical fact that excessive minimization of inexact criteria may become harmful (the NN overtraining and feature oversampling [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, we found that the sample size was a major source of an increase in the classification error in the fusion rule design.

3) Comparative investigations of six real-world pattern classification problems demonstrated that successfully stopped pairwise SLP-based classifiers are useful options in the first stage of decision making. If the prudent fusion rule is used, the SLPs were comparable and time and again, outperformed the linear SV classifiers in moderate dimensional situations.

4) 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, pseudovalidation-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). The colored noise injection proves to be a powerful tool to facilitate finite sample-size-based model selection problems in moderate-dimensional pattern recognition tasks.

ACKNOWLEDGMENT

The authors would like to thank Prof. A. Kaklauskas, R. P. W. Duin, R. Somorjai, O. K. Baykan, and A. Babalik for data provided for the experiments and their useful discussions, and Mrs. N. Bizys who edited the text of this paper.

REFERENCES


Sarunas Raudys received the M.S. degree and Ph.D. degrees in computer science from Kaunas University of Technology, Kaunas, Lithuania, in 1963 and 1969, respectively, and USSR doctor of science (Habil.) degree from the Institute of Electronics and Computer Science, Riga, Latvia, in 1978.

In the past he was a part time Researcher at Vilnius Gediminas Technical University (VGTU), Vilnius, Lithuania. Currently, he is a Professor at the Department of Informatics, Vilnius University, Vilnius, Lithuania. His research interests are multivariate analysis, data mining, statistical pattern recognition, artificial neural networks, machine learning, evolvable multiagent systems, and artificial life.

Rimantas Kybartas received the M.S. degree in computer science from Vilnius University, Vilnius, Lithuania, in 2003.

He is a former Research Assistant at Vilnius Gediminas Technical University (VGTU), Vilnius, Lithuania. Currently, he is a Research Assistant at Vilnius University and Senior Software Developer at the Bank of Lithuania. His current research interests are analysis of financial time series, economy and financial data mining, changing environments, statistical pattern recognition, artificial neural networks, and multiagent systems.

Edmundas Kazimieras Zavadskas received the M.S. degree in civil engineering from Vilnius branch of Kaunas University of Technology, Kaunas, Lithuania, in 1967, the Ph.D. degree from Vilnius Gediminas Technical University (VGTU), Vilnius, Lithuania in 1973, and the USSR doctor of science degree from Moscow State University of Civil Engineering, Moscow, Russia, in 1987.

Currently, he is the Head of the Department of Construction Technology and Management at VGTU. His research interests are building technology and management, decision-making theory, automated design, and decision support systems.

Dr. Zavadskas is a member of the Lithuanian Academy of Sciences.