Experts' Boasting in Trainable Fusion Rules

Sarunas Raudys

Abstract—We consider the trainable fusion rule design problem when the expert classifiers provide crisp outputs and the behavior space knowledge method is used to fuse local experts' decisions. If the training set is utilized to design both the experts and the fusion rule, the experts' outputs become too self-assured. In small sample situations, "optimistically biased" experts' outputs bluffs the fusion rule designer. If the experts differ in complexity and in classification performance, then the experts' boasting effect and can severely degrade the performance of a multiple classification system. Theoretically-based and experimental procedures are suggested to reduce the experts' boasting effect.

Index Terms—Fusion rule, expert classifiers, generalization error, resubstitution error, complexity.

1 Introduction

Multiple classifier systems (MCS) became a popular approach for designing complex pattern recognition systems [1], [2], [3], [4]. In this approach, initially a number of "simple" expert (base) classifiers categorize unknown pattern vectors. A fusion rule aggregates the outputs of the first-level experts and makes a final decision. Like feature extraction and feature selection, MCS is an approach designed to utilize additional informal designer's information [5], [6]. Splitting the decision-making procedure into two stages, by designing separately the experts and the fusion rule, changes training set size/complexity relations.

A great deal of research in the pattern recognition community focused on fusion rules [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Fixed fusion rules (majority voting, sum, product, etc.) are very popular. Such rules are obviously based on quite strong assumptions, such as comparable performances of all members in the MCS, statistical independence between solutions of the experts, etc., [2], [3], [7]. Fusion of the experts' outputs can be regarded as a problem of statistical pattern recognition. Linear weighted voting, the naive Bayes classifiers, the kernel function approach, potential functions, the behavior-knowledge space method, decision trees, and multilayer perceptrons are among the most popular techniques used for experts fusion [4], [5], [6], [7], [8], [9], [10], [11]. Special approaches such as bagging, boosting and arcing classifiers, mixture of experts, stacked generalization have been suggested [1], [12], [13], [14], [15], [16]. In many practical problems, simple, fixed, nontrainable rules compete or even outperform trainable ones [17], [18]. It means that certain difficulties arise in finite sample size situations. Sample size effects can be divided into three types:

1. Generalization errors of the expert classifiers increase due to imperfect training.
2. Generalization errors of the fusion rule increase due to imperfect training.
3. If the training set is used twice, to train the experts and the combiner, the fusion rule designer is being bluffed since she/he utilizes biased resubstitution error estimates of quality of each single expert.

The first effect necessitates the utilization of simple base classifiers as possible. The second effect requires that one has to adapt the complexity of the fusion rule to the sample size: In the small sample case, one needs to use only simple fusion rules. Only for large sample sizes should one work with complex combiners. The third effect requires that the fusion rule designer has to distrust experts' "self-evaluations" if the expert classifiers are complex and training set sizes are too small [3], [11], [19]. This paper deals with the third problem that was almost unconsidered in the literature.

In [16], leaving-one-out estimates were used to design the combiner. In [11], Euclidean distance and standard linear Fisher classifiers were utilized as experts in a linearly weighted sum type of fusion. To improve linear fusion rule, correction terms to evaluate the experts' boasting were derived. These corrections helped to improve the accuracy of MCS, however, the gain in classification error reduction was not appreciable. The objective of the present paper is to consider the much more complex behavior-knowledge space (BKS) fusion method when linear Fisher classifiers are used as experts. The BKS method is nonlinear and, if sample size/complexity relations are satisfied, it can give acceptable results. Moreover, a pruned BKS method makes up a decision tree classifier that also can be used as fusion rule. In contrast to the correction terms derived in [11], standard formulae to evaluate bias of the resubstitution error estimate (see e.g., Section 6.3.1.2 in [20]) are used and adapted to case where nonlinear method is used for fusion. It enables *a better understanding* of the problem of combining classifiers, especially when the base classifiers are overtrained. It gives *two useful procedures* for minimizing the undesired effects when this is the case.

2 The Multinomial Classifier as a Fusion Rule

If the experts provide crisp outputs (class labels), then, as the sample size increases, the asymptotically optimal statistical decision rule is provided by the multinomial classifier [20], [21] usually referred to by MCS proponents as the BKS method [21], [22]. In the pattern recognition literature, the use of the BKS method as the fusion rule was found very promising, but seriously limited when the training data set size was small [8], [23].

Consider K pattern classes and L expert classifiers. Denote the decision made by jth expert by e_j. Suppose e_j can take one of the labels $\{0, 1, \dots, K-1\}$. Thus, for the design of the fusion rule, we have a discrete-valued vector $\boldsymbol{E} = (e_1, e_2, \dots, e_L)^T$. The total number of possible combinations of L outputs (states) $e_1, e_2, \dots, e_L$ is $m = K^L$. Each vector $\boldsymbol{E}$ can assume only one state, s_r, from the m possible ones, $s_l, s_2, \dots, s_{m-1}, s_m$. In a statistical approach, it is supposed that values $s_1, \dots, s_m$ follow multinomial distribution. The conditional distribution of the ith class vector $\boldsymbol{E}$, taking one of m "states" is characterized by m probabilities

$$P_1^{(i)}, P_2^{(i)}, \dots, P_{m-1}^{(i)}, P_m^{(i)}, \text{ with } \sum_{r=1}^{m} P_r^{(i)} = 1, (i = 1, \dots, K).$$

Let P_i be a prior probability of the ith pattern class, π_i. Then, *Bayes rule* should allocate vector $\boldsymbol{E}$, falling into the rth state, according to maximum of the products

$$P_1 P_r^{(1)}, P_2 P_r^{(2)}, \dots, P_K P_r^{(K)}. \tag{1}$$

To use the allocation rule, we have to know $K \times (m-1)$ probabilities $P_1^{(1)}, P_2^{(1)}, \dots, P_{m-1}^{(K)}$ and the class priors $P_1, P_2, \dots, P_{K-1}$. If the fusion rule makes its prediction based only on the class labels $e_1, e_2, \dots, e_L$ supplied by the expert classifiers and if all probabilities in (1) are known, it is the ***optimal classifier***. No other fusion rule can perform better. It is worth noting that the multinomial classifier based fusion rule will fall against an ***oracle***, an ideal fusion rule. The oracle is a hypothetical rule. It makes the correct classification if at least one expert is exact. It makes an error, however, if all experts classify

- *The author is with the Institute of Mathematics and Informatics, Akademijos 4, Vilnius 2021, Lithuania. E-mail: raudys@das.mii.lt.*

Manuscript received 27 Sept. 2001; revised 19 Sept. 2002; accepted 9 Dec. 2002.
Recommended for acceptance by M. Pietikainen.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115084.

0162-8828/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

vector $\boldsymbol{x}$ incorrectly. In comparison with the multinomial fusion rule, the oracle utilizes an additional information (vector $\boldsymbol{x}$).

In practice, the $K \times m$ probabilities $P_1^{(1)}, P_2^{(1)}, \dots, P_m^{(K)}$ are unknown and must be estimated from the data. One can use the maximum likelihood (ML) estimates (sample frequencies)

$$\hat{P}_r^{(i)} = n_r^{(i)}/N_i, \tag{2}$$

where N_i is a priori fixed number of training vectors from the ith pattern class and, among those, $n_r^{(i)}$ is a number of vectors falling into the state s_r.

In this case, we have a sample-based multinomial classifier (BKS method). We will use the two names interchangeably. If the training set is used twice, to train the experts and the combiner, the ML estimates (2) are optimistically biased. In order to better understand the problem of combining classifiers, decompose the biased ML estimate $\tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(i)}$ into an unbiased part $\hat{\hat{\boldsymbol{P}}}{}_r^{(i)}$ (it would be the ML estimate if an independent set would be used to train the fusion rule) and the expert's bias Δ_{ir}, i.e., $\tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(i)} = \hat{\hat{\boldsymbol{P}}}{}_r^{(i)} + \Delta_{ir}$. If inaccurate estimates $\tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(1)}$ and $\tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(2)}$ are used, a generalization error occurs

$$\begin{aligned} EP_{gen}^{\mathrm{BKS}} = &\sum_{r=1}^{m} Prob\Big\{P_1 \tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(1)} = P_2 \tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(2)}\Big\}(P_1 P_r^{(1)} + P_2 P_r^{(2)})/2 \\ &+ \sum_{j=1}^{m} Prob\Big\{P_1 \tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(1)} < P_2 \tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(2)}\Big\} P_1 P_r^{(1)} \\ &+ \sum_{j=1}^{m} Prob\Big\{P_1 \tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(1)} > P_2 \tilde{\hat{\hat{\boldsymbol{P}}}}{}_r^{(2)}\Big\} P_2 P_r^{(2)}. \end{aligned} \tag{3}$$

The above equation makes clear that, in order to calculate generalization error, one has to know entire set of $2m-2$ probabilities, $P_1^{(1)}, P_2^{(1)}, \dots, P_{m-1}^{(2)}$.

An example. In order to show the effect of the expert boasting bias on the generalization error, consider MCS with L statistically independent experts performing binary classification $(K=2)$. After receiving vector $\boldsymbol{x}$, let the experts produce binary (0 or 1) outputs,$e_1, e_2, \dots, e_L$. We assumed that the outputs $e_j (j = 1, \dots, L)$ are independent binomial variables. Therefore, conditional probability $P_j^{(i)}$of the ith class, rth cell $(0 < r = 1 + e_1 + 2e_2 + 2^2 e_3 + \dots + 2^{L-1} e_L \leq m)$ can be expressed as the product

$$P_r^{(i)} = \prod_{j=1}^{L} (P_{ij})^{e_j} (1 - P_{ij})^{1-e_j}, \tag{4}$$

whereP_{ij} is the probability that the jth expert assigned the ith class vector to the first class. P_{2j} and $1 - P_{1j}$ are conditional probabilities of misclassification of the first and second sorts.

Consider artificial Gaussian data model **A**, with 2,600 features divided into seven blocks (one block for each base classifier). Let the blocks be mutually independent and let the pattern classes share common covariance matrix. Assume that individual asymptotic errors of each of five blocks be $P_\infty^{\mathrm{F}} = 0.3535$ (Mahalanobis distances $\delta = 0.752$, for an introduction to statistical pattern recognition see [24], [25]), number of features in each of them, $n = 500$; asymptotic errors of the last two blocks $P_\infty^{\mathrm{F}} = 0.1172 (\delta = 2.378)$, $n = 50$.

In finite training set situations, generalization errors of the expert classifiers increase due to imperfect training. We have $\bar{N} = 500$ training vectors of each class. Exploitation of formulae for standard linear Fisher classifier (Section 6.3.1.2 in [20], see also the equations at the beginning of Section 3) gives $EP_{gen}^{\mathrm{F}} = 0.4503$ (for first five experts), $EP_{gen}^{\mathrm{F}} = 0.1274$ (for last two experts). Thus, for data model **A** one can use $P_{2j} = 1 - P_{1j} = 0.4503 (j = 1,2,3,4,5), P_{2j} = 1 - P_{1j} = 0.1274 (j = 6,7)$ and (4) to calculate true cells' probabilities $P_1^{(1)}, \dots, P_{128}^{(2)}$. If the fusion rule designer would know unbiased probabilities of incorrect classification of each single expert (0.4503 and 0.1274), she/he could design an ideal *fusion* rule with an asymptotic classification error $P_\infty^{\mathrm{BKS}} = \frac{1}{2}\sum_{r=1}^{m} \min\{P_r^{(1)}, P_r^{(2)}\} = 0.1068$ (we assumed $P_2 = P_1 = \frac{1}{2}$). If the designer utilizes the training set twice, instead of generalization error estimates, EP_{gen}^{F}, she/he has to use resubstitution estimates, $\hat{P}_R^{\mathrm{F}}$. When the training set size is $\bar{N} = 500$, exploitation of formulae presented in the literature (Section 6.3.1.2 in [20], see also the equations at the very beginning of Section 3) gives expected resubstitution errors $E\hat{P}_R^{\mathrm{F}} = 0.1287$ (for first five experts) and $E\hat{P}_R^{\mathrm{F}} = 0.1072$ (for last two experts). Suppose that the fusion rule designer utilizes the biased probabilities, $E\hat{P}_R^{\mathrm{F}}$, and (4) to evaluate cell probabilities and to build BKS fusion rule. Calculation according to (4) and (3) gives the generalization error of this fusion rule, $EP_{gen}^{\mathrm{BKS}} = 0.2028$. Here, for calculations, we used theoretically calculated (nonrandom) estimates of the cell probabilities. Therefore, in (3), the probabilities $Prob\{P_1 \hat{\hat{\boldsymbol{P}}}{}_r^{(1)} > = < P_2 \hat{\hat{\boldsymbol{P}}}{}_r^{(2)}\}$ were equal either to 0, $\frac{1}{2}$, or 1. We see biased estimates lead to error almost twice larger as unbiased ones ($P_\infty^{\mathrm{BKS}} = 0.1068$).

Suppose now that $500 + 500$ *additional* independent learning vectors are used to obtain cell estimates $\hat{\hat{\boldsymbol{P}}}{}_r^{(i)} (r = 1, \dots, 128; i = 1, 2)$. Estimates $\hat{\hat{\boldsymbol{P}}}{}_r^{(i)}$ are binomial random variables. Therefore, to evaluate the generalization error, one can use (3) with

$$\begin{aligned} &P\Big\{P_1 \hat{\hat{\boldsymbol{P}}}{}_r^{(1)} < P_2 \hat{\hat{\boldsymbol{P}}}{}_r^{(2)}\Big\} P_1 P_r^{(1)} = \\ &\sum_{l=0}^{\bar{N}-1} \sum_{v=j+1}^{\bar{N}} \frac{P_1 \bar{N}!}{l!(\bar{N}-l)!} \left(P_r^{(1)}\right)^{l+1} \\ &\left(1 - P_r^{(1)}\right)^{\bar{N}-l} \frac{\bar{N}!}{v!(\bar{N}-v)!} \left(P_r^{(2)}\right)^{v+1} \left(1 - P_r^{(2)}\right)^{\bar{N}-v}. \end{aligned}$$

Other terms in (3) can be calculated in a similar way (see also (3.46) in [20]). For $\bar{N} = 500$, we calculate $P_{gen}^{\mathrm{BKS}} = 0.1201$. It is notably closer to the asymptotic error, $P_\infty^{\mathrm{BKS}} = 0.1068$, as the classification error of the fusion rule is based on nonrandom, however, *biased* expert estimates (in the later case, we found that $EP_{gen}^{\mathrm{BKS}} = 0.2028$). It allows us to predict that the second utilization of the learning set (to build BKS fusion rule) increased the generalization error by 8 percent.

3 Fighting the Bias in Error Rate Estimation

Correction term. When we train the expert classifiers and use the training set to evaluate the cells' probabilities $P_1^{(1)}, P_2^{(1)}, \dots, P_m^{(K)}$, we deal with biased (apparent) error estimates. Consider a standard two-category example where the pattern classes are Gaussian and share a common covariance matrix. The expectation of the resubstitution error estimate of the linear Fisher classifier is [20] $E\hat{P}_R^{\mathrm{F}} = \Phi\{-\frac{1}{2}\delta\sqrt{T_M T_\Sigma}\}$, where $T_{\mathrm{M}} = 1 + \frac{2n}{\delta^2 \bar{N}}, T_\Sigma = 1 + \frac{n}{2\bar{N} - n}$, n is the input dimensionality (in an MCS design, n can be different for each expert), and δ is the Mahalanobis distance. The expected value of the generalization error is $EP_{gen}^{\mathrm{F}} = \Phi\{-\frac{1}{2}\delta/\sqrt{T_M T_\Sigma}\}$. For a multivariate Gaussian data model, these equations can be used to calculate expectations of resubstitution and generalization errors.

In the example considered in the previous section, we had the MCS with seven expert classifiers. The asymptotic errors of the first five experts were $P_\infty^{\mathrm{F}} = 0.3535$. For training set $\bar{N} = 500$ vectors from each of the classes and dimensionality $n = 500$ for the first five experts, we calculated resubstitution error $E\hat{P}_R^{\mathrm{F}} = 0.1287$ and generalization error $EP_{gen}^{\mathrm{F}} = 0.4503$. For the other two experts with asymptotic error $P_\infty^{\mathrm{F}} = 0.1172$ and input feature dimensionality $n = 500$, we found $E\hat{P}_R^{\mathrm{F}} = 0.1072$ and $EP_{gen}^{\mathrm{F}} = 0.1274$.

In order to reduce the effect of the expert's boasting, the fusion rule designer ought to use unbiased generalization error estimates. In the two-category case, for the linear Fisher classifier, the following almost unbiased estimate of generalization error was recommended ([20], ((6.30)):

$$\begin{aligned}\hat{P}_R^{\mathrm{F}*} &= \hat{P}_R^{\mathrm{F}} + \left(EP_{gen}^{\mathrm{F}} - E\hat{P}_R^{\mathrm{F}}\right)\\ &= \hat{P}_R^{\mathrm{F}} + \left(\Phi\left\{-\frac{\hat{\delta}}{2}\frac{1}{\sqrt{T_M T_\Sigma}}\right\} - \Phi\left\{-\frac{\hat{\delta}}{2}\sqrt{T_M T_\Sigma}\right\}\right),\end{aligned} \tag{5}$$

where $\hat{\delta}$ is the sample estimate of distance δ. It can be obtained from resubstitution classification error estimate by means of interpolation the equation $\hat{P}_R^{\mathrm{F}} = \Phi\{-\frac{\hat{\delta}}{2}\sqrt{T_M T_\Sigma}\}$.

Let L statistically independent experts provide binary (0 or 1) crisp outputs, $e_1, e_2, \ldots, e_L$. Then, according to (4), the conditional probability $P_r^{(i)}$ of the $r\mathrm{th}$ cell is a function of the probabilities of incorrect classification, P_{2j} and $1 - P_{1j}$. Replacing P_{2j} and $1 - P_{1j}$ in (4) by the *generalization error* estimates of each expert, $\hat{P}_{Rj}^{\mathrm{F}*}$, we obtain an "unbiased" estimate of a conditional generalization error in the $r\mathrm{th}$ cell, $\hat{P}_{Gr}^{(i)}$. Replacing P_{2j} and $1 - P_{1j}$ in (4) by the *resubstitution error* estimates $\hat{P}_{Rj}^{\mathrm{F}}$ $(j = 1, 2, \ldots, L)$, we construct an almost "unbiased" estimate of conditional resubstitution error in the $r\mathrm{th}$ cell, $\hat{P}_{Rr}^{(i)}$ (for the sake of simplicity, we assume $P_{2r} = 1 - P_{1r}$). The modified term, $\hat{\Delta}_{ir} = \hat{P}_{Gr}^{(i)} - \hat{P}_{Rr}^{(i)}$, can be used to reduce the bias of the $r\mathrm{th}$ cell's probability estimate:

$$\hat{P}_{r\,\mathrm{unbiased}}^{(i)} = \hat{P}_r^{(i)} + \left(\hat{P}_{Gr}^{(i)} - \hat{P}_{Rr}^{(i)}\right). \tag{6}$$

Equation (6) is valid if the experts' solutions are mutually statistically independent. Moreover, the data have to be Gaussian, with a pooled two-pattern class covariance matrix. In other cases (non-Gaussian data, another type of expert classification rule, many-pattern classes, etc.), correction term $\hat{\Delta}_{ir} = \hat{P}_{Gr}^{(i)} - \hat{P}_{Rr}^{(i)}$ is not based theoretically any more.

Noise injection. One of the possible strategies to reduce the resubstitution error bias, in the general case, is to create a pseudovalidation set by means of a noise injection. In the noise injection technique, we form a pseudovalidation set by adding many (say, ni_{nn}) randomly generated zero mean vectors to each training pattern vector. Spherical Gaussian (white) noise, however, can distort the intrinsic dimensionality of the data. To reduce data distortion, colored noise injection [20], [26], [27] can be used. In k-nearest neighbor-directed noise injection, for each single training vector, $\boldsymbol{x}_{is}$, one finds its k nearest neighbors, $\boldsymbol{x}_{is1}, \boldsymbol{x}_{is1}, \ldots, \boldsymbol{x}_{isk}$, from the same pattern class. Then, one adds random Gaussian $\mathrm{N}(0, \sigma_n^2)$ noise ni_{nn} times along the k lines connecting $\boldsymbol{x}_{is}$ and $\boldsymbol{x}_{is1}, \boldsymbol{x}_{is1}, \ldots, \boldsymbol{x}_{isk}$.

Three parameters have to be defined to realize a noise injection procedure: k, the number of neighbors, ni_{nn}, the number of new, artificial vectors generated around vector $\boldsymbol{x}_{is}$, and σ_n^2, the noise variance. In our experiments, we used: $k = 2$ (two nearest neighbors), $ni_{nn} = 10, \sigma_n^2 = 1$. For the MCS fusion rule design, we classify vectors of artificial pseudovalidation set by L expert classifiers and use the classification results to estimate unknown probabilities, $P_1^{(1)}, P_2^{(1)}, \ldots, P_{m-1}^{(K)}$. Just as in kernel discriminant analysis, *the noise smoothes the sample estimates of the cell probabilities*. In fact, *noise injection introduces additional information*, by filling the space between nearest vectors of one pattern class with vectors of the same category. Similarly to smoothing in the kernel discriminant analysis, a noise injection technique is effective if the intrinsic dimensionality of the data is low [20].

4 Simulation Experiments

Three artificial multivariate Gaussian data models sharing a common covariance matrix for two-pattern classes were used to verify the efficacy of our theoretical estimates. Strictly speaking, correction term becomes exact if the sample size is increasing without bound (the dimensionality can increase too). Then, variances of conditional classification error are small ([20], Section 3.4.4). Therefore, we intentionally set the input dimensionality high and the sample size large in order to have correct estimates of classification errors. Large differences between performances and dimensionalities of different experts were chosen to reveal the effectiveness of the BKS method.

TABLE 1
Means of Generalization Errors and Their Standard Deviations of MCS in the Experiments with Three Sets of Artificial and a Real-World Satellite Data

Data	**BKSIdeal**	**BKStand**	**BKSNoise**	**BKSModi**	**MajorVot**
A	0.099/.003	0.223/.020	0.193/.021	0.111/.006	0.203/.007
B	0.087/.003	0.162/.043	0.144/.037	0.114/.018	0.301/.008
C	0.086/.002	0.169/.030	0.105/.013	0.109/.014	0.297/.010
Experts	**BKSIdeal**	**BKStand**	**BKSNoise**	**BestValid**	**MajorVot**
Generic MLP	0.049/.003	0.083/.006	0.057/.004	0.067/.006	0.059/.004
Specialized MLP	0.055/.004	0.348/.109	0.060/.003	0.069/.007	0.077/.020
Single Experiment	0.0615	0.3182	0.0726	0.0748	0.0980

The data model **A** consists of 2,600 *independent* features divided into seven blocs: 500, 500, 500, 500, 500, 50, 50 features. All 2,600 features have unit variance, all features in one block have the same mean value, selected to have the Bayes error and Mahalanobis distances defined at the end of Section 2. In data models **B** and **C**, we have 1,000 features. The first five experts use 500 overlapping features, the last two experts use 50 features each. In model **A**, the experts are statistically independent; in models **B** and **C**, they are dependent. The difference between models **B** and **C** is the intrinsic dimensionality of the data: In model **B**, all 1,000 variances are equal to 1. In model **C**, the variances of 11 informative features are unity; standard deviations of the remaining features were set to 0.01. The experts' complexity and the asymptotic errors are the same in all three data models: Five experts (Fisher classifiers) operate in 500-variable feature space and are rather "weak." The last two experts work in 50-variable space and are considerably more powerful. In artificial data models, **B** and **C**, we had the same correlations between the experts' outputs. For each model, 10 independent experiments with training sets of size $N = 500 + 500$ were performed.

A real world, two-category **satellite data** was composed of 15,787 eight-dimensional vectors. The entire training data, $4,384 + 4,242$ vectors, was split randomly into five training sets $(876 + 848$ vectors each). Experiments with each training set were performed two times starting form different initial weights to train multilayer perceptrons (MLP) with four hidden units used as expert classifiers; 10 independent experiments in total. The perceptrons were trained with 35 training epochs by the Levenberg-Marquardt algorithm. In each of the first 10 independent experiments, 13 MLPs were trained starting from different random initial weights. Seven "best" experts were selected. The experts' selection was performed according to classification error estimates obtained from the artificial pseudovalidation set. We refer to these experts as "*generic*" (nonspecific) ones. The BKS fusion rules were trained using all $876 + 848$ vectors in the particular learning set. The test set comprised $3,555 + 3,606$ independent vectors.

In the second part of the study (an additional 10 independent experiments), we used five *specialized* (with four hidden units) and two generic MLPs. The specialized experts were trained on different training subsets. Five nonintersecting subsets were formed of $876 + 848$ vectors by means of cluster analysis. Therefore, specialized MLPs were experts in different regions of input feature space.

Mean values and standard deviations of generalization error obtained in 10 experiments are presented in Table 1. Abbreviations: **BKStand** is the standard and **BKSIdeal** is the ideal BKS fusion rule (test set vectors were used to evaluate the cells' probabilities $P_1^{(1)}, \ldots, P_{m-1}^{(2)}$). **BKSNoise** denotes the BKS rule when the pseudovalidation set was utilized to evaluate the cells' probabilities.

MajorVot stands for fixed majority voting fusion rule. **BKSModif** denotes the modified BKS rule, when the ML sample-estimated cell probabilities (2) were corrected by (4), (5), (6), and the Fisher classifiers were used as experts. **Best Valid** designates that the best expert was selected from the pseudovalidation set error estimates.

Experiments with the Gaussian data set **A** confirm the theoretical estimates presented at the end of Section 2. The increase in classification error *due to the expert's bias* is much greater than the increase *due to the imperfectly trained fusion rule*. The mean experimental generalization error, 0.223 (for joint expert bias and imperfect fusion rule training), is only a little bit greater than the theoretically calculated classification error, 0.2028, found from (3) for the case when only the experts' boastings were taken into account. For data model **A**, the **BKSModif** rule completely compensates the experts' boasting effect: The experiments give the same ratio of the generalization error of **BKSModif** to the asymptotic error, $0.111/0.099 = 1.121$, as the theoretical evaluation. In the latter case, we used ratio $0.1201/0.1068 = 1.1245$, calculated above assuming that *additional* learning set was used to train the fusion rule.

For data sets **B** and **C**, for which the experts' outputs were statistically dependent, the analytical boasting correction **BKSModif** was also highly effective. The 10-fold noise injection **BKSNoise** was effective only in experiments with data set **C**, which has *low intrinsic dimensionality*. If the intrinsic dimensionality is high, $5,000 + 5,000$ new artificial vectors are insufficient to fill the 500-dimensional feature space and to smooth the sample estimates of the cell probabilities $P_1^{(1)}, P_2^{(1)}, \dots, P_{m-1}^{(2)}$. For data set **C**, method **BKSNoise** was as effective as the analytically-based corrections **BKSModif** and outperformed both the weighted voting rule and the best single expert (12.74 percent of errors, on average).

In the experiments with the satellite data, and MLPs as expert classifiers, we could not use the theoretically corrections derived for linear Fisher classifier and multivariate Gaussian data model. Table 1 shows that the **BKSNoise** procedure was effective: it helped to reduce the generalization error and outperformed other fusion methods. In addition, for a single experiment with specialized experts (last row in Table 1), we utilized all 15,787 vectors to calculate the cell probabilities $P_1^{(1)}, P_2^{(1)}, \dots, P_{m-1}^{(2)}$ and found $P_B^{\mathrm{BKS}} = 0.0712, EP_{gen}^{\mathrm{BKS}} = 0.0773$ (calculations for $\bar{N} = 794$ from (3.46) in [20]). This outcome denotes that the expected probability of misclassification increases $0.0773/0.0712 = 1.086$-fold. In a single experiment with one training set and a noise injection, we got the generalization error increase ratio $0.0726/0.0615 = 1.181$. The average obtained in 10 experiments is $0.060/0.055 = 1.091$. These results suggest that a noise injection approach practically eliminated the expert bias effect.

While processing the results obtained in the simulation experiments, we designed moderately pruned *binary decision tree* classifiers to be used as the fusion rule. A moderate complexity reduction of the fusion rule helped in reducing the generalization error. However, for such simpler classifiers, an additional noise injection was less effective. When comparing the efficacy of the much more complex trainable BKS fusion rule with that of simple fixed fusion rules, (e.g., the majority voting rule), we observed that a nontrainable rule is more successful if the experts were "generic," i.e., not specialized in certain regions of the input feature space, or if the classification errors of all expert classifiers were comparable (row "*Generic MLP*" in Table 1). If the experts are specialized and their performances differ substantially, then majority voting loses against the multinomial rule with noise injection (rows **A**, **B**, **C**, and "*Specialized*" in Table 1).

5 Discussion

When designing trainable expert decision fusion rules, sample size effects can be divided into three components: 1) an increase in the generalization error due to imperfect training of the expert classifiers, 2) an increase in the generalization error due to insufficient training of the fusion rule, and 3) the experts' boasting. The expert boasting effect is present in all trainable fusion rules if the training set is used twice, to train both the experts and the fusion rule.

We have shown both theoretically and experimentally that expert boasting can become extremely harmful. If for high-dimensional Gaussian data the standard linear Fisher classifiers are used as experts and a multinomial classifier is used for fusion, (4), (5), and (6) compensate the increase in generalization error and gives *theoretical explanation of the expert boasting phenomenon*. The correction terms cannot be used when the data are non-Gaussian, when more complex types of the expert classification rules are employed, or when the number of pattern classes exceeds two. Due to large variances of statistical estimates of classification error, the correction terms are also ineffective when the training set size is too small.

A more general technique, the k-nearest neighbor-directed noise injection may be recommended. If the intrinsic dimensionality of the data is not too high, noise injection helps in smoothing the cell probability estimates of the multinomial classifier, simplifies the fusion rule, and reduces the experts' boasting effect. This technique can be used even for selecting the experts in a fixed, nontrainable majority voting procedure. In this case, artificial validation set could be used to select a fixed number of best experts.

Our earlier experiments with spherical noise injection [19], indicated that with increasing the noise variance σ_n^2, there is a peaking effect for the generalization error as a function of σ_n^2. Thus, one of the problems for future research is to find a way to control the σ_n^2 value. An important unsolved problem is determining what is the minimal sample size for which the expert boasting effect can still be reduced, at least in principle.

Acknowledgments

The author would like to thank Dr. Krystyna Stankiewicz for providing the satellite data for the experiments, Professors Fabio Roli and Ray Somorjai and Arūnas Janeliūnas for their help and challenging and fruitful discussions. Special thanks to anonymous reviewers for benevolent remarks and suggestions.

References

[1] "Multiple Classifier Systems," *Lecture Notes in Computer Science,* J. Kittler and F. Roli, eds., vols. 1857, 2096, 2364, Springer, 2000, 2001, 2002.

[2] J. Kittler, "Combining Classifiers: A Theoretical Framework," *Pattern Analysis and Applications,* vol. 1, pp. 18-27, Springer, 1998.

[3] J. Kittler, "A Framework for Classifier Fusion: Is Still Needed?" *Lecture Notes in Computer Science,* vol. 1876, pp. 45-56, 2000.

[4] J. Gosh, "Multi-Classifier Systems: Back to the Future," *Lecture Notes in Computer Science,* vol. 2364, pp. 1-15, Springer, 2002.

[5] T.K. Ho, "Data Complexity Analysis for Classifier Combination," *Lecture Notes in Computer Science,* vol. 2096, pp. 53-67, Springer, 2001.

[6] S. Raudys, "Multiple Classifier Systems in the Context of Feature Extraction and Selection," *Lecture Notes in Computer Science,* vol. 2364, pp. 27-41, Springer, 2002.

[7] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On Combining Classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, pp. 226-239, 1998.

[8] L.I. Kuncheva, J.C. Bezdek, and R.P.W. Duin, "Decision Templates for Multiple Classifier Fusion: An Experimental Comparison," *Pattern Recognition,* vol. 34, pp. 299-314, 2001.

[9] L. Xu, A. Krzyzak, and C.Y. Suen, "Methods of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 22, pp. 418-435, 1992.

[10] S. Hashem, "Optimal Linear Combination of Neural Networks," *Neural Networks,* vol. 19, pp. 599-614, 1997.

[11] A. Janeliunas and S. Raudys, "Reduction of Boasting Bias' of Linear Expert," *Lecture Notes in Computer Science,* vol. 2364, pp. 242-251, Springer, 2002.

[12] L. Breiman, "Bagging Predictors," *Machine Learning J.,* vol. 24, pp. 123-140, 1996.

[13] L. Breiman, "Arcing Classifiers," *Annals of Statistics,* vol. 26, pp. 801-849, 1998.

[14] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and Systems Science,* vol. 55, pp. 119-139, 1997.

[15] M. Jordan and R. Jakobs, "Hierarchical Mixture of Experts and the EM Algorithm," *Neural Computation,* vol. 6, pp. 181-214, 1994.

[16] D. Wolpert, "Stacked Generalization," *Neural Networks,* vol. 5, pp. 241-260, 1992.

[17] F. Roli and G. Fumera, "Analysis of Linear and Order Statistics Combiners for Fusion of Imbalanced Classifiers," *Lecture Notes in Computer Science,* vol. 2364, pp. 252-261, Springer, 2002.

[18] F. Roli, S. Raudys, and G.L. Marcialis, "An Experimental Comparison of Fixed and Trained Rules for Crisp Classifiers Outputs," *Lecture Notes in Computer Science,* vol. 2364, pp. 232-241, Springer, 2002.

[19] C. Güler, B. Sankur, Y. Kahya, M. Skurichina, and S. Raudys, "Classification of Respiratory Sound Patterns by Means of Cooperative Neural Networks," *Proc. Eighth European Signal Processing Conf.,* G. Ramponi, G.L. Sicuranza, S. Carrato, and S. Marsi, eds., 1996.

[20] S. Raudys, *Statistical and Neural Classifiers: An Integrated Approach to Design,* p. 312, London: Springer, 2001.

[21] P.A. Lachenbruch and M. Goldstein, "Discriminant Analysis," *Biometrics,* vol 5, pp. 9-85, 1979.

[22] Y.S. Huang and C.Y. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, pp. 90-93, 1998.

[23] L.I. Kuncheva and C.J. Whitaker, "Feature Subsets for Classifier Combination: An Enumerative Experiment," *Lecture Notes in Computer Science,* vol. 2096, pp. 228-237, Springer, 2001.

[24] K. Fukunaga, *Introduction to Statistical Pattern Recognition,* second ed. New York: Academic Press, 1990.

[25] R.O. Duda, P.E. Hart, and D.G. Stork., *Pattern Classification,* second ed. New York: Wiley, 2000.

[26] R.P.W. Duin, "Nearest Neighbour Interpolation for Error Estimation and Classifier Optimisation," *Proc. Eighth Scandinavian Conf. Image Analysis,* K.A. Hogda, B. Braathen, and K. Heia, eds., pp. 5-6, 1993.

[27] M. Skurichina, S. Raudys, R.P.W. Duin, "K-Nearest Neighbors Directed Noise Injection in Multilayer Perceptron Training," *IEEE Trans. Neural Networks,* vol. 11, pp. 504-511, 2000.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.