Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners and Open Problems*

Š. J. Raudys

Institute of Mathematics and Cybernetics
Lithuanian Academy of Sciences
Vilnius, 232600, USSR

A. K. Jain

Department of Computer Science
Michigan State University
East Lansing MI 48824, USA

Abstract

During the last two decades a considerable amount of effort has been made in analyzing the influence of both training and testing sample size on the design and performance of pattern recognition systems. These questions are interesting to practitioners as well as theoreticians, because the small-sample effects can easily contaminate the design and evaluation of a proposed system. For applications with a large number of features and a complex classification rule, the training sample size must be quite large. A large test sample is required to accurately evaluate a classifier with a low error rate. The design of a pattern recognition system consists of several stages: data collection, formation of the pattern classes, feature selection, specification of the classification algorithm, and estimation of the classification error. In this paper, we will discuss the effects of sample size on feature selection and error estimation for several types of classifier. In addition to surveying prior work in this area, our emphasis is on giving practical advice to today's designers and users of statistical pattern recognition systems.

1 Introduction

During the last two decades a considerable amount of effort has been made in analyzing the influence of both training (also called design or learning) and testing sample size on the design and performance of pattern recognition systems (see *e.g.* reviews [1, 3, 5, 9, 14, 16, 33, 37]). These questions are interesting to practitioners as well as theoreticians, because the small-sample effects can easily contaminate the design and evaluation of a proposed system. For applications with a large number of features and a complex classification rule, the training sample size must be quite large. A large test sample is essential to accurately evaluate a classifier with a very low error rate.

The design of a pattern recognition system consists of several stages: data collection, formation of the pattern classes, feature selection, specification of the classification algorithm, and estimation of the classification error. In this paper, we will discuss the effects of sample size on feature selection and error estimation for several types of classifier. In addition to surveying prior work in this area, our emphasis is on giving practical advice to today's designers and users of statistical pattern recognition systems. The paper is organized as follows. Section 2 introduces the classifiers we will focus on in this paper. In Section 3, we explore classifier design in the context of small design sample size. The estimation of error rates under small test sample size follows in Section 4. Section 5 investigates sample size effects in feature selection. Section 6 analyzes the problem of determining learning and test sample sizes. Section 7 contains discussion and highlights some open research problems.

2 Classification Algorithms

In the pattern recognition literature, there are a large number of ways to use sample observations to design a classification rule. One can use a statistical decision function approach with the Gaussian or exponential family of distributions along with a dozen of structural forms for the covariance matrices. One can further assume the covariance matrices to be equal or different for the various pattern classes. Classical maximum likelihood approaches can be used to estimate the parameters of the probability density functions corresponding to the pattern classes. In case of complex multimodal pattern classes, one can use a number of modifications of piecewise linear, piecewise quadratic classification rules, nonparametric Parzen window, or K-NN classifiers. The latter two can differ in the metric used to define the distance between two pattern vectors, and in methods used to edit the learning sample. There are several versions of classifiers based on potential functions [6] differing in a family of transformations of the pattern vector **X**, and in the optimization criteria. There are at least eight types of pattern error functions used to evaluate an empirical risk function in order to design linear discriminant functions. Nearly two dozen methods exist for finding classification rules using heuristics when different similarity measures are applied to define the similarity between vector **X**, and the class π_i. A number of statistical models and expansions are known for approximating discrete distributions which can be used to design the classification rule for observations characterized by discrete or mixed variables. Therefore, the total number of classification methods which have been proposed in the pattern recognition literature exceeds two hundred.

We describe several important classifiers which have seen practical use. We will concentrate on the two-class problem in this paper. An unclassified p-dimensional multivariate feature vector **X** is allocated to the class π_1 if the discriminant function (DF) $g(\mathbf{X})$ is positive and to the class π_2 otherwise.

The quality of a classification rule will be characterized by its probability of misclassification (PMC). In the following definitions of PMC, we are assuming that the number of test samples is infinite. In Section 4, we discuss the estimation of PMC when only a finite number of test samples is available.

- *Bayes PMC*: P_B is the PMC of an optimal Bayes classifier.
- *Conditional PMC*: P_N^α is the PMC of the classifier α trained on a given training sample of size N. We assume that the training samples are labeled.
- *Expected PMC*: EP_N^α is the expectation of P_N over all random training samples of size N_i from class π_i, $N = N_1 + N_2$.
- *Asymptotic PMC*: the probability of misclassification under the classifier α designed with an infinite number of training samples.

$$P_\infty^\alpha = \lim_{N_1, N_2 \to \infty} EP_N^\alpha \qquad (1)$$

Note that P_∞^α can be different for different classifiers α. Also, $P_\infty^\alpha \geq P_B$ for all classifiers α.

*This work was supported in part by NSF grant CDA-8806599.

CH2898-5/90/0000/0417$01.00 © 1990 IEEE

We now briefly review six commonly used classifiers and provide the corresponding discriminant functions.

2.1 Euclidean distance classifier

This classifier makes classifications only according to sample means, $\bar{\mathbf{X}}^{(1)}$ and $\bar{\mathbf{X}}^{(2)}$ of the two classes. Its discriminant function is written as

$$\begin{aligned}\hat{g}^E(\mathbf{X}) &= (\mathbf{X}-\bar{\mathbf{X}}^{(2)})^T(\mathbf{X}-\bar{\mathbf{X}}^{(2)}) - (\mathbf{X}-\bar{\mathbf{X}}^{(1)})^T(\mathbf{X}-\bar{\mathbf{X}}^{(1)}) \\ &= 2\cdot[\mathbf{X}-\frac{1}{2}(\bar{\mathbf{X}}^{(1)}+\bar{\mathbf{X}}^{(2)})]^T(\bar{\mathbf{X}}^{(1)}-\bar{\mathbf{X}}^{(2)}) \qquad (2)\end{aligned}$$

The Euclidean distance classifier can be used when the pattern classes are well separated or when we want to implement a simple decision rule.

2.2 Fisher's linear discriminant

This is perhaps the most commonly used classification rule. The discriminant function is given by

$$\hat{g}^F(\mathbf{X}) = [\mathbf{X}-\frac{1}{2}(\bar{\mathbf{X}}^{(1)}+\bar{\mathbf{X}}^{(2)})]^T\mathbf{S}^{-1}(\bar{\mathbf{X}}^{(1)}-\bar{\mathbf{X}}^{(2)}) + \ln\frac{q_1}{q_2}, \qquad (3)$$

where q_1 and q_2 are the prior probabilities of the classes π_1 and π_2, respectively, and S is the sample covariance matrix (assumed to be common to both classes). It is an asymptotically optimal rule for the classification of Gaussian populations with a common covariance matrix.

2.3 Quadratic discriminant function

$$\begin{aligned}\hat{g}^Q(\mathbf{X}) = &\ (\mathbf{X}-\bar{\mathbf{X}}^{(2)})^T\mathbf{S}_2^{-1}(\mathbf{X}-\bar{\mathbf{X}}^{(2)}) \\ &-(\mathbf{X}-\bar{\mathbf{X}}^{(1)})^T\mathbf{S}_1^{-1}(\mathbf{X}-\bar{\mathbf{X}}^{(1)}) \\ &+\ln\frac{|\mathbf{S}_2|q_1}{|\mathbf{S}_1|q_2}, \qquad (4)\end{aligned}$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ are the sample estimates of the class-conditional covariance matrices, and $|\mathbf{S}_i|$ denotes the determinant of $\mathbf{S}_i$. The quadratic DF (Equation (4)) is a plug-in rule obtained from an optimal quadratic DF for two Gaussian populations where sample means and sample covariances have replaced the true parameters. But the resulting sample-based DF is not 'optimal' in the Bayesian sensé [6]. It is important to note that, when training sample sizes N_1 and N_2 from the two classes are unequal, then the performance of the plug-in discriminant functions is further degraded [4]. The main reason for the nonoptimality of the plug-in discriminant function is its bias [4, 16]. Grabauskas [13] found the expected value of Equation (4) and proposed an unbiased quadratic discriminant function.

2.4 Parzen window classifier

The Parzen window classifier does not assume a particular form for the class-conditional densities. Its discriminant function is

$$\hat{g}^P(\mathbf{X}) = \frac{1}{N_1}\sum_{i=1}^{N_1}K\left(\frac{\mathbf{X}-\mathbf{X}_i^{(1)}}{\lambda}\right) - \frac{1}{N_2}\sum_{i=1}^{N_2}K\left(\frac{\mathbf{X}-\mathbf{X}_i^{(2)}}{\lambda}\right), \qquad (5)$$

and depends on the window function $K(\cdot)$ and on the value of the smoothing parameter λ. The most popular window functions is the exponential window. Grabauskas [13] has tested 20 types of window functions (including Gaussian, logistic, trapezoidal, triangular, rectangular, and sinusoidal) and found that with the proper selection of smoothing parameters, all 20 classifiers had nearly equal error rates. However, the value of the smoothing parameter is very important. It has been proved theoretically [6, 41] that the value of λ should decrease with an increase in the design sample size N. An optimal value of λ which minimizes classification error depends both on the design sample size and on the distribution of the pattern vectors $f_i(\mathbf{X})$. When we have two Gaussian populations with equal covariance matrices, the optimal decision boundary is a hyperplane and λ should be very large. When we have complex multimodal distributions and the decision boundary is extremely nonlinear, then even for small design sample sizes we have to use a small value of the parameter λ. Many different criteria have been used to find λ_{opt} [17]. Theoretical considerations can show only the qualitative characteristics of the dependence of the optimal value of the smoothing parameter on dimensionality and sample size. It is impossible to find an optimal λ, λ_{opt}, which minimizes the error rate for all class-conditional densities. In order to find λ_{opt} for a particular problem, we recommend evaluating the classifier's performance for several values of λ and choosing that value which provides the best performance.

2.5 K-nearest-neighbor (K-NN) classifier

In the K-NN rule, the class of the input pattern $\mathbf{X}$ is chosen as the class of the majority of its K nearest neighbors. The Euclidean metric is commonly used for distance calculations; however, the Mahalanobis metric can sometimes lead to better performance. The K-NN and Parzen window classifiers have many similar characteristics. It can be said that the K-NN classifier is the Parzen window classifier with a hyper-rectangular window function in the p-dimensional feature space. Both classifiers allow us to obtain complex nonlinear decision boundaries. The curvature of the boundary depends on the value of the smoothing parameters K and λ. When $K = 1$ or $\lambda \to 0$, the curvature is maximum; it diminishes with the increase of K or λ. The performance of K-NN classifier in finite design sample case significantly depends on the number K of nearest neighbors. Analogous to the Parzen window classifier, we recommend estimating the classification error for several values of K simultaneously, to find an optimal value of K.

2.6 Multinomial classifier

This classifier is used for the recognition of patterns described by discrete variables. Let the jth variable take m_j distinct values. The p-variate vector $\mathbf{X}$, can therefore take one of $m = m_1m_2\dots m_p$ values (states). Let $p_{ij}, i = 1,2; j = 1,...,m$ be the probability that a pattern from the class π_i takes on the jth state. Then the optimal Bayes discriminant function is given by

$$g^M(\mathbf{X}) = q_1p_{1s(\mathrm{X})} - q_2p_{2s(\mathrm{X})}, \qquad (6)$$

where $s(\mathrm{X})$ denotes the label of the state corresponding to $\mathbf{X}$. In practice, instead of the true probabilities p_{ij}, one uses sample estimates

$$\hat{p}_{ij} = \frac{n_{ij}}{N_i} \qquad (7)$$

where n_{ij} stands for the number of cases when the pattern vectors from the design sample of the class π_i have taken on the jth state, resulting in the sample-based multinomial discriminant function $\hat{g}^M(\mathbf{X})$.

Sometimes the multinomial classifier is applied to continuous variables after making them discrete. One example of such a classifier is the *histogram classifier*, where for each class we design a histogram containing m bins. Such a classifier is very similar to the Parzen window classifier with rectangular windows in *a priori* fixed positions.

When the dimensionality p is not small, then even in the binary case ($m_i = 2$), the total number of states ($m = 2^p$) is very large and it is difficult to obtain reliable estimates p_{ij}. In such a case, one needs to introduce some additional information in order to simplify the design. One possible way to do this is to assume that the variables are independent. Another alternative is to reduce the number of states, m, by designing a decision tree classifier.

A decision tree consists of a root node, intermediate nodes and $m^*(m^* \ll m)$ terminal nodes. At the root node, the best feature performs the decision. At the intermediate nodes, different features may participate at the same level. The final classification of the discrete vector $\mathbf{X}$ is performed according to the class number attributed to each particular terminal node. There are several methods to construct decision trees (see *e.g.* [3, 22, 39, 36]). An advantage of such a classification rule is its applicability for classification of objects described by mixed variables and a comparatively easier interpretation of the classification.

3 Sensitivity to Design Sample Size

In the finite design sample case, the parameters of the classifiers are estimated with low accuracy. Therefore, the resulting plug-in classifiers differ from optimal ones, resulting in an increase in the classification error. An increase in the classification error due to the finiteness of the design sample size $\Delta_N^\alpha = EP_N^\alpha - P_\infty^\alpha$ depends, first of all, on the type of the classification rule α, on the number of features p, and, further, on the value of the asymptotic probability of misclassification. Significant research efforts have been made to find the relationship between classification error, learning sample size, dimensionality and complexity of the classification algorithm (see, for example, the reviews in [1, 7, 12, 16, 19, 28, 33, 32, 37, 40]).

While designing the parametric classifiers, each parameter estimate introduces it own contribution to the increase in the classification error. Below, we present an asymptotic formula for the increase in the expected PMC of parametric classifiers [33] under the assumption of Gaussian class conditional densities.

$$\Delta_N{}^\alpha = EP_N^\alpha - P_\infty^\alpha = \frac{1}{N}\frac{\psi(\frac{\delta}{2})}{\delta}\sum_{i \in C}\theta_i, \tag{8}$$

where $\psi(t) = \frac{1}{2\pi}\exp(\frac{-t^2}{2})$. The number of terms in set C in Equation (8) depends on the classifier type. In Equation (2), we used only sample means of the two classes. Therefore, for the Euclidean distance classifier, only the θ_2 term appears in Equation (8). For the linear Fisher discriminant function (Equation (3)), we used the estimates of the priors, means and common covariance matrix. Therefore, here we should use the terms θ_1, θ_2 and θ_5. Analogously, for the quadratic discriminant function (Equation (4)), we should use the terms θ_1, θ_2 and θ_6.

The classification error of the parametric classifiers is proportional to $\frac{1}{N}$ and depends on the dimensionality of the feature vector p; for the linear classifiers, the relationship is linear and for the quadratic classifier the relationship is quadratic (only for large p when $p \gg \delta^2$). Analytical and simulation studies show that for the nonparametric classifiers (Parzen and multinomial), the decrease of $\Delta_N = EP_N - P_\infty$ with an increase in the design sample size, N, is slower ($O(\frac{1}{\sqrt{N}})$ or $O(\frac{1}{N^{1/3}})$). For large values of the smoothing parameter λ (when the Parzen window classifier becomes similar to the Euclidean classifier), the decrease of Δ_N is of $O(\frac{1}{N})$. Our theoretical and simulation studies have shown that when we use the same smoothing parameter λ for all features, then the increase in classification error Δ_N of the Parzen window classifier depends not on the actual dimensionality p, but on the intrinsic dimensionality, p^* of the patterns [29]. The analysis also shows that the design sample size required to achieve a learning accuracy determined *a priori*, depends on the dimensionality exponentially:

$$N = \alpha\beta^{p^*}, \tag{9}$$

where scalars α and β depend on asymptotic and expected probabilities of misclassification, and on the value of the smoothing parameter. The required design sample size for multinomial classifier depends linearly on the number of states m (when the distribution of the probabilities p_{ij} is "quasiuniform").

Estimates of the design sample sizes have been obtained for the case of Gaussian distributions with identical covariance matrices. In reality, additional factors effect the increase in the classification error, such as unequal covariance matrices and unequal design sample sizes from both populations. Therefore, the above estimates only provide some guidelines. Moreover, these relations between sample size and dimensionality are determined for a fixed value of asymptotic PMC. While solving real pattern recognition problems, P_∞ decreases with the addition of new variables, but then the problem of determining optimal number of features arises (see Section 5).

An important quantity is the variance of the conditional probability of misclassification, $V(P_N)$. From Efron's analysis [7], it follows that for several parametric linear classifiers (Fisher's discriminant, logistic regression, and the Euclidean distance classifier), the increase in classification error ($P_N - P_\infty$) is distributed as a scaled chi-squared random variable $\chi^2 \cdot c/N$ with p degrees of freedom, where the constant c depends on the asymptotic probability of misclassification and on the type of classification rule. Thus the ratio of the standard error of P_N to the mean increase in PMC $\Delta_N = EP_N - P_\infty$ is $\sqrt{V(P_N)}/P_\infty = \sqrt{(2/p)}$, which tends to zero as dimensionality increases.

4 Performance Estimation

A number of methods for estimating the classification error have been proposed in the literature reviews [9, 11, 14, 21, 24, 31, 34, 38, 25, 26]. These methods can be studied by using the following two factors:

- The way in which multivariate observations are used to design the classifier and to test its performance;
- The pattern error function that determines the contribution of each observation of the test set to the estimate of the probability of misclassification.

There are four main approaches to use the given observations as the design set and as the test set.

1. The **Resubstitution Method $\mathcal{R}$**: all observations are used to design the classifier and used again to estimate its performance.
2. The **Hold-Out Method $\mathcal{H}$**: Suppose the total number of available observations is n^*. One portion of the set of observations (the *design set* containing N observations) is used to design the classifier, and the remaining $(n^* - N)$ portion (the *test set*) is used to estimate the error rate.
3. The **Leave-k-Out Method $\mathcal{L}$**: In this method, $\begin{pmatrix} n^* \\ k \end{pmatrix}$ classifiers are designed. Each classifier is designed by choosing k of the n^* observations as a design set, and its error rate is estimated using the remaining observations. This process is repeated for all distinct choices of k patterns from the set of observations. A popular choice for the value of k is $k = 1$.
4. The **Bootstrap Method $\mathcal{B}$**: A *bootstrap design sample* of size N is formed from the N observations by sampling with replacement. The classification rule is designed using this bootstrap sample and is tested twice:

    - N observations of the bootstrap design sample are used to obtain a bootstrap resubstitution estimate $P_{\mathcal{R}}^{\mathcal{B}}$; and
    - the original design set is used to obtain the bootstrap estimate of conditional error $P_N^{\mathcal{B}}$.

    This procedure is repeated r times (typically, r lies between 10 and 200). An arithmetic mean $\bar{\Delta}_{N\mathcal{R}}^{\mathcal{B}}$ of the differences

    $$\Delta_{N\mathcal{R}}^{\mathcal{B}^i} = P_N^{\mathcal{B}^i} - P_{\mathcal{R}}^{\mathcal{B}^i}, i = 1, ..., r \tag{10}$$

    is used to reduce the optimistic bias of the resubstitution estimate:

    $$\hat{P}_{\mathcal{B}} = \hat{P}_{\mathcal{R}} + \bar{\Delta}_{N\mathcal{R}}^{\mathcal{B}} \tag{11}$$

There are many modifications of the bootstrap method: the randomized bootstrap, the 0.632 estimator, the MC estimator, the complex bootstrap (see *e.g.* [14, 15]).

Each of the above error estimation procedures can be used with different pattern error functions $h(\hat{g}(\mathbf{X}))$, where $\hat{g}(\mathbf{X})$ is the sample-based discriminant function:

$$\hat{P} = \frac{1}{n_t} \sum_{j=1}^{n_t} h(\hat{g}(\mathbf{X}_j)), \qquad (12)$$

and $\mathbf{X}_1, ..., \mathbf{X}_{n_t}$ are test sample observations.

1. *Error Counting* (EC):

$$h^{EC}(\hat{g}(\mathbf{X})) = \begin{cases} 1 & \text{if } \hat{g}(\mathbf{X}) < 0 \text{ and } \mathbf{X} \in \pi_1 \\ 1 & \text{if } \hat{g}(\mathbf{X}) > 0 \text{ and } \mathbf{X} \in \pi_2 \\ 0 & \text{otherwise} \end{cases} \qquad (13)$$

Here, correctly recognized observations do not affect the estimate of PMC.

2. *Smooth Modification of EC* (SM) [11]:

$$h^{SM}(\hat{g}(\mathbf{X})) = \begin{cases} 1 & \text{if } \hat{g}(\mathbf{X}) \geq a \text{ and } \mathbf{X} \in \pi_2 \\ 1 & \text{if } \hat{g}(\mathbf{X}) \leq -a \text{ and } \mathbf{X} \in \pi_1 \\ 1 - \frac{\hat{g}(\mathbf{X})+a}{b} & \text{if } -a < \hat{g}(\mathbf{X}) < b - a \text{ and } \mathbf{X} \in \pi_1 \\ 1 + \frac{\hat{g}(\mathbf{X})-a}{b} & \text{if } a - b < \hat{g}(\mathbf{X}) < a \text{ and } \mathbf{X} \in \pi_2 \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

Here, part of the correctly classified observations contribute to the estimation of miclassification probability.

3. *Posterior probability estimate* (PP) [10, 23]:

$$h^{PP}(\hat{g}(\mathbf{X})) = \frac{1}{2}[1 - |\tanh(\hat{g}(\mathbf{X})/2)|], \qquad (15)$$

where the $\hat{g}(\mathbf{X})$ defined in Equation (15) is given by

$$\hat{g}(\mathbf{X}) = \frac{q_1 \hat{f}_1(\mathbf{X})}{q_2 \hat{f}_2(\mathbf{X})} \qquad (16)$$

and $\hat{f}_i(\mathbf{X})$ is a sample estimate of the probability density function $f_i(\mathbf{X})$. An advantage of this estimate is that the test sample observations can be unlabeled. Information about the design sample propagates into the error estimate as well.

4. *Quasiparametric estimate* (QP) [21]: Here it is assumed that the values of the discriminant function $\hat{g}(\mathbf{X})$ have a Gaussian distribution. PMC is found analytically from sample means and variances of the values $\hat{g}(\mathbf{X}_j^{(i)}), i = 1, 2$ and $j = 1, ..., N_i$.

Thus, in principle, we can have 16 error estimation methods.

The resubstitution method results in optimistically biased estimates of the asymptotic PMC P_∞. Therefore, it can be used only when the sample size is sufficiently large. It was shown analytically [8, 31] (for Euclidean and Fisher classifiers) and experimentally that the bias of the resubstitution estimate ($\Delta_{\mathcal{R}} = P_\infty - \hat{P}_{\mathcal{R}}$) is approximately equal to the bias of the expected PMC ($\Delta_N = EP_N - P_\infty$). The hold-out error counting estimate results in an unbiased estimate of the expected PMC. The disadvantage of this method is that not all observations of the design sample take part in the learning process and only a part of observations are used to evaluate the classification error.

The leave-one-out error counting prodcedure produces a practically unbiased estimate of the expected PMC if the sample observations are statistically independent. In case of dependent observations this method approaches the resubstitution method and results in an optimistically biased estimate of the expected PMC, EP_N. The leave-one-out estimate $\hat{P}_{\mathcal{L}}$ can be used together with resubstitution estimate $\hat{P}_{\mathcal{R}}$ in order to get an estimate of the asymptotic PMC:

$$\hat{P}_\infty = \frac{\hat{P}_{\mathcal{L}} + \hat{P}_{\mathcal{R}}}{2}. \qquad (17)$$

Analytical and experimental investigations of the error-counting methods show that the variance can be expressed by a simple formula [31, 35, 34].

$$V\hat{P}_\eta = \frac{E\hat{P}_\eta(1 - E\hat{P}_\eta)}{n_t}, \qquad (18)$$

where $E\hat{P}_\eta$ and $V\hat{P}_\eta$ are the mean and variance of the error estimate $\hat{P}_\eta$, and η indicates the method: $\mathcal{R}, \mathcal{H}, \mathcal{L}$, or $\mathcal{B}$; n_t is the number of test samples (for $\mathcal{R}$, $\mathcal{L}$, and $\mathcal{B}$ methods, $n_t = N$). Since the resubstitution method is optimistically biased it has the smallest mean and consequently the smallest variance.

The variance of the SM, PP, and QP estimates can be less than the variance of the error-counting estimate. However, the SM, PP, and QP estimates are often biased. The bias of the SM estimate depends directly on the degree of smoothing of the pattern error function (parameters a and b in Equation (14)) and can exceed the absolute value of the classification error. It was observed experimentally [31] that the QP estimate is pessimistically biased in the low-dimensional case, when the distribution function of the discriminant function $\hat{g}(\mathbf{X})$ is non-Gaussian. The PP estimate is based on an information contained in the test sample and on additional information used to obtain estimates $\hat{f}_i(\mathbf{X})$ of the probability densities $f_i(\mathbf{X})$ of the pattern classes. In the parametric case, this information can be useful and can reduce the variance of the PP estimate. When the design sample size is small or when the additional information is incorrect (*e.g.*, we assume normality for the class-conditional densities when in fact they are significantly non-Gaussian), then the estimated class-conditional probability densities and consequently the pattern error function (Equation (15)) are determined with large errors. This leads to significant bias of the PP estimates. The bias is especially large in the nonparametric case, where very vague prior information is typically used to obtain estimates of $\hat{f}_1(\mathbf{X})$ and $\hat{f}_2(\mathbf{X})$ in Equation 16.

In feature selection (see Section 5), the bias of the estimates of the classification error is not critical if it is approximately equal for all subsets of variables. However, in estimating the performance of a complete pattern recognition system, the use of the biased estimates is dangerous.

5 Feature Selection

The purpose of feature selection and extraction is to identify those features which are important in discriminating among pattern classes. The need to retain only a small number of *useful* and *good* features in designing a classifier has been well documented in the literature [6].

5.1 Optimal number of features

In Section 3, we examined the relationship between sample size and dimensionality for fixed asymptotic PMC, P_∞. While solving real pattern recognition problems, the addition of new features usually decreases P_∞. Usually, the 'best' features are added first, and less-useful features are added later. Therefore, the rate of decrease in P_∞ slows as the number of features increases. Adding new features requires that new parameters be estimated. An inexact estimation of parameters increases classification error. If this increase is larger then the decrease in classification error produced by the addition of the new feature, then the net effect is that addition of the new feature increases the error rate. Therefore, in the finite-sample-size case we have a 'peaking' phenomenon: classification error initially drops with addition of new features, then attains a minimum, and then begins to increase. The number of features at which the expected PMC, EP_N, is minimal is called the ***optimal*** number of features, and is denoted p_{opt}. It depends on the design sample size, the type of classification rule, the class-conditional distributions of the pattern vector $\mathbf{X}$, and most importantly, on the effectiveness of features and their ordering [19, 30]. In practice, it is important to know if the optimal number of features, p_{opt}, is lower than the initial number of features, p. Typically, p_{opt} is smaller for smaller design sample sizes, for more complex classification algorithms, and for better orderings of features. When all features are equally effective, or when the features are unordered and added in a random way, p_{opt} can be large (for the linear discriminant function, $p_{opt} = \frac{N}{2} - 1$ for N training patterns [18]; for the quadratic discriminant function, p_{opt} is significantly lower than $\frac{N}{2} - 1$, but still increases with the number of training patterns N).

5.2 Accuracy of feature selection

Usually, features are not ranked according to their effectiveness in discrimination *a priori*. We use the sample information to compare the effectiveness of features and rank them. The sample estimates of the effectiveness are not exact. Therefore, only the best features can be ranked properly. The effectiveness of the worst features differs negligibly and the accuracy of sample estimates is not sufficient for exact ranking of features or the feature sets. The inaccuracy of the estimates of the feature effectiveness causes a bias in the estimates of the best subsets containing $i = 1, ..., p-1$ features. Therefore, the estimates of p_{opt} become biased also. The problem of estimating p_{opt} in the case of empirical ordering of features is unsolved.

The ordering of features is an important step in the design of a pattern recognition algorithm. It is well known that the ordering of features and the ordering of feature subsets are two different subjects. In general, the best subset of t features and the set of t individually-best features are not identical ($t < p$). The only procedure which guarantees that the best subset is found, is a complete inspection of all subsets, which is computationally expensive. Therefore, many suboptimal procedures for feature subset selection have appeared in the literature [20]. None of these techniques guarantee that the best feature subset will be found, but they typically require much less computation than exhaustive search. Most procedures use either sequential addition of features, sequential deletion of features, or a combination of both approaches. Other techniques include random search (inspection of randomly-selected subsets), directed random search, and branch-and-bound techniques.

Each feature selection procedure can be carried out by using every one of a number of feature effectiveness criteria or the classification error estimation methods mentioned in the previous section.

There are approximately two dozen parametric criteria of feature effectiveness known in the pattern recognition literature [2, 42]. Examples are the Mahalanobis, Bhattacharyya, Patrick-Fisher, and Matusita distances, divergence, mutual Shannon information, and entropy. Analytical expressions of these criteria are usually simpler then the Bayes error expressions. Some of them use additional information about the populations to be classified. For example, the Mahalanobis distance criterion is based on assumptions of multivariate normality for all classes, with a common covariance matrix. The variances of criteria containing such assumptions are usually lower than variances of the nonparametric estimators described above; however, biased estimates are produced if the parametric assumptions are not valid. Therefore, the usefulness of the criteria with parametric assumptions can be justified only when the bias is approximately equal for all subsets of features. It is unclear whether one specific strategy provides consistently better performance than most others. One experimental comparison [27] concluded that forward selection and random search outperformed other procedures in one application.

The analysis of feature selection strategies is complicated by the fact that all feature effectiveness criteria are subject to error, caused by both sample size effects and the simplifying assumptions. Inaccurate criteria of effectiveness can lead to incorrect rankings of features and feature subsets. Therefore, one objective of feature selection is to find feature subsets which produce an expected PMC close to the ideal value (the value produced by the truly 'optimal' feature subset).

6 Sample Size Determination

There are two occasions when the designer of a pattern recognition system has to determine the size of the sample:

1. To find a sample size sufficient to achieve a desired level of learning accuracy, and
2. To find the size of the test sample sufficient to estimate the classification error.

It was mentioned in Section 3 that the minimum design sample size depends on the method used to find coefficients of the classification rule, the number of features, the asymptotic PMC, and the desired learning accuracy. Estimates of the sufficient sample size depend slightly on the asymptotic PMC. Therefore, in order to use them in practice we have to estimate an interval in which P_∞ will lie. For example, we might guess that by using the Fisher's linear discriminant, $P_\infty \in [0.01, 0.1]$. Then, the requirement for the efficiency of the design sample requires $32 \leq N \leq 72$ for a dimensionality of eight. If we assume $P_\infty \in [0.1, 0.5]$, then $N \leq 32$.

The estimates of the minimum design sample size were obtained for some idealized (spherically Gaussian or *quasiuniform*) class-conditional densities $f_i(\mathbf{X})$ (see Section 3). For other distributions, the required design sample sizes can be different. Therefore, in order to estimate the sufficiency of the design sample size, we recommend additionally to use the following nonparametric estimate of the increase in the classification error $\Delta_N = EP_N - P_\infty$:

$$\hat{\Delta}_N = \frac{\hat{P}_{\mathcal{L}} - \hat{P}_{\mathcal{R}}}{2}, \tag{19}$$

where $\hat{P}_{\mathcal{L}}$ and $\hat{P}_{\mathcal{R}}$ are the leave-one-out and resubstitution estimates of the classification error, respectively. The estimate $\hat{\Delta}_N$ is based on the fact that the dependencies $EP_N = \phi_1(N)$ and $E\hat{P}_{\mathcal{R}} = \phi_2(N)$ are nearly symmetrical (see Section 4) and that $E\hat{P}_{\mathcal{L}} \approx EP_N$.

If the difference $\hat{\Delta}_N$ is small in comparison with the empirical estimate of the asymptotic PMC (Equation (17)), then we can conclude that the design sample size is sufficient. Here, we have to pay attention to the variances of the estimates $\hat{\Delta}_N$ and $\hat{P}_\infty$. Extensive simulation studies have suggested that the estimates $\hat{P}_{\mathcal{L}}$ and $\hat{P}_{\mathcal{R}}$ are practically statistically independent. Therefore, the estimates of the variances and mean square errors (MSE) of $\hat{\Delta}_N$ and $\hat{P}_\infty$ can be found from Equation (18):

$$MSE(\hat{\Delta}_N) = \frac{1}{2}\sqrt{\frac{\hat{P}_{\mathcal{L}}(1-\hat{P}_{\mathcal{L}})}{n_t} + \frac{\hat{P}_{\mathcal{R}}(1-\hat{P}_{\mathcal{R}})}{n_t}} = MSE(\hat{P}_\infty). \tag{20}$$

For example, when solving a pattern recognition problem with $N_1 = N_2 = 100$, (*i.e.* $N = 200$), with $\hat{P}_{\mathcal{L}} = 0.09$ and $\hat{P}_{\mathcal{R}} = 0.05$, then $\hat{P}_\infty = 0.07$, $\hat{\Delta}_N = 0.02$, $MSE(\hat{P}_\infty) = MSE(\hat{\Delta}_N) = 0.013$, *i.e.* the increase in the classification error is 30% of the asymptotic PMC. Therefore, we can conclude that the design sample size is sufficient. Note that in nonparametric estimation of the classification error, dimensionality does not play a role.

The size of the test sample n_t used to determine the performance of the classifier can be determined from the variance (Equation (18)). If we require that the error counting estimate of the PMC does not deviate from the true value P by more than k%, then

$$2\sqrt{\frac{P(1-P)}{n_t}} = \frac{Pk}{100}. \tag{21}$$

Therefore, the estimate of the sufficient test sample size n_t is

$$n_t = \frac{4(1-P)}{P(k/100)^2}. \tag{22}$$

Here, we again have to guess at a possible value or interval of the true classification error. For example, if we assume $0.02 < P < 0.1$, then Equation (22) produces an interval of $900 < n_t < 4900$ for $k = 20\%$.

While using the hold-out error estimation method we have to divide an existing set of observations into two parts: the design sample and the test sample. If the design sample is small, the classification error will be large. If the test sample is too small, then the variance of the error estimator will be large. In order to find an optimal balance between the sizes of the design and test samples, we have to introduce a loss function. One possible loss function is

$$\mathrm{LOSS}(N_1, N_2) = C_1(EP_N(N_1, N_2) - P_\infty) + C_2\mathrm{MSE}\{\hat{P}(n^* - N_1 - N_2)\}, \tag{23}$$

where n^* is the total number of observations, $N_1 + N_2$ of which are used to design the classifier and the remainder used as the test sample, C_1 and C_2 are the costs associated with an increase in classification error (due to design sample size) and an increase in MSE of the error estimate (due to test sample size), respectively.

From the definition of the loss function above, it follows that an optimal division of the samples into testing and design sets depends on the type of classifier, the dimensionality, and on the asymptotic PMC. The theoretical results mentioned in Sections 3 and 4 can help to find a solution, but no complete procedure has yet been devised. From Equation (18) we have the MSE of the error counting estimate:

$$MSE(\hat{P}(n^* - N_1 - N_2)) = \frac{\hat{P}(1-\hat{P})}{n^* - N_1 - N_2}. \tag{24}$$

For parametric classifiers, and assuming $N_1 = N_2 = N$, Equation (8) yields

$$EP_N - P_\infty = \alpha(P_\infty, p)\frac{1}{N}, \tag{25}$$

where the coefficient $\alpha(P_\infty, p)$ depends on the classifier, dimensionality, and P_∞. For practical use of this methodology some prior guesses about the value of P_∞ should be available. Let $n^* = 300$, $N_1 = N_2 = \frac{N}{2}$, $C_1 = C_2 = 1$, the dimensionality $p = 8$, and assume the linear classifier is employed with $P_\infty = 0.1$. From a table in [32], we obtain $EP_N/P_\infty = 1.18$ when $N_1 = N_2 = 40$ and find $\alpha(0.1, 8) = (EP_N - P_\infty)(N_1 + N_2) = 0.018 \cdot 80 = 1.44$. Then the loss function (Equation (23)) is

$$LOSS(N) = \frac{\alpha}{N} + \frac{\sqrt{P(1-P)}}{\sqrt{300 - N}} = \frac{1.44}{N} + \frac{0.3}{\sqrt{300 - N}}, \tag{26}$$

which attains a minimum near $N \approx 140$ and therefore $N_1 = N_2 = 70$, $n_t = 160$. The optimal balance for other values of P_∞ is found in a similar way.

7 Discussion

One needs a large number of training samples if a complex classification rule with many features is being utilized. In many pattern recognition problems, the number of potential features is very large and not much is known about the characteristics of the pattern classes under consideration, so it is difficult to determine a priori the complexity of the classification rule needed. Therefore, even when the designer believes that a large number of training samples has been selected, they may not be enough for designing and evaluating the classification problem at hand.

A small sample size can cause many problems in the design of a pattern recognition system. For example, the classification error depends on the particular training sample set used, so it is a random variable. Its expectation, the expected error, approaches the asymptotic error as the number of training samples increases. The value of the asymptotic error depends on the true class-conditional densities (nature) and on the type of the classification rule used (designer). The rate at which the expected error converges to the asymptotic error depends on the complexity of the classification rule and the number of features. Therefore, in case of finite training sample size, addition of new features or the use of more complex decision rules is not always useful. This, then, leads to the well-known problem of determining the optimal number of features and the best classification rule [19, 30].

It is very difficult to directly apply known theoretical results about finite training sample effects to classifier design. First of all, true class-conditional densities are not known to the designer. Further, theoretical results are obtained only under some idealized conditions, such as Gaussian densities, very large number of training samples, etc. Therefore, theoretical results can only guide the designers in the proper direction. The final design should be based on empirical results obtained by comparing competing pattern recognition algorithms. With the availability of powerful desk top workstations, it is fairly easy to evaluate competing classifiers. However, an important thing to remember in this comparison is that the error estimates are biased in finite sample case. For this reason, apparent error is not a reliable measure for an empirical evaluation of classifiers.

In conclusion, the small sample effects make the problem of designing a classification system very difficult, and these effects should not be ignored in practice.

References

[1] Aivazian, S.A., Buchstaber, V.M., Yenyukov, I.S. and Meshalkin, L.D., "Applied Statistics: Classification and Reduction of Dimensionality," Reference edition, *Finansy i statistika*, Moscow, 1989 (in Russian).

[2] Ben-Bassat, M., "Use of Distance Measures, Information Measures and Error Bounds in Feature Evaluation," *Handbook of Statistics*, Vol.2 (P.R. Krishnaiah and L.N. Kanal, eds.), North Holland, pp.773-791, 1982.

[3] Breiman, L., Friedman, J., Olsen, R.A. and Stone, C.J., *Classification and Regression Trees*, Belmont: Wadsworth, 1984.

[4] Chandrasekaran, B. and Jain, A.K., "On balancing decision functions," *J. Cybernetic Info. Sci.*, **2**, pp. 12-15, 1979.

[5] Devroye, L. and Wangner, T.J., "Nearest Neighbor Methods in Discrimination," *Handbook of Statistics*, **2** (P.R. Krishnaiah and L.N. Kanal, eds.), North Holland, pp. 193-198, 1982.

[6] Duda, R.O. and Hart, P.E., *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

[7] Efron, B., "The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis," *J. Am. Stat. Ass.* **70**, pp. 892-898, 1975.

[8] Foley, D.M., "Considerations of Sample and Feature Size," *IEEE Trans. on Info. Theory* **IT-18**, pp. 618-626, 1972.

[9] Fukunaga, K., "Statistical Pattern Recognition," *Handbook of Pattern Recognition and Image Processing* (T.Y.Young and K.S.Fu, eds.), New York:Academic, pp. 3-32, 1986.

[10] Fukunaga, K. and Hostetler, L.D., "Optimization of K-Nearest-Neighbor Density Estimates," *IEEE Trans. Inf. Theory*, **IT-19**, pp. 320-326, 1973.

[11] Glick, N., "Additive Estimators for Probabilities of Correct Classification," *Pattern Recognition* **10**, 3, pp. 211-222, 1978.

[12] Goldstein, M. and Dillon, W.R., *Discrete Discriminant Analysis*, New York: Wiley, 1978.

[13] Grabauskas (Institute of Mathematics and Cybernetics, Academy of Sciences, Lithuania), personal communication, 1983.

[14] Hand, D.J., "Recent Advances in Error Rate Estimation," *Pattern Recognition Letters* **5**, pp. 335-346, 1986.

[15] Jain, A.K., Dubes, R.C. and Chen, C.C., "Bootstrap Techniques for Error Estimation," *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-9**, 9, pp. 628-63, 1987.

[16] Jain, A.K. and Chandrasekaran, B., "Dimensionality and Sample Size Considerations in Pattern Recognition Practice," *Handbook of Statistics* **2** (P.R. Krishnaiah and L.N. Kanal, eds.), North Holland, pp. 835-855, 1982.

[17] Jain, A.K. and Ramaswami, M.D., "Classifier Design with Parzen Windows," *Pattern Recognition and Artificial Intelligence*, (E. S. Gelsema and L. N. Kanal, eds.), Elsevier, pp. 211-228, 1988.

[18] Jain, A.K. and Waller, W.G., "On the Optimal Number of Features in the Classification of Multivariate Gaussian Data," *Pattern Recognition* **10**, pp. 365-374, 1978.

[19] Kanal, L. and Chandrasekaran, B., "On Dimensionality and Sample Size in Statistical Pattern Classification," *Pattern Recognition* **3**, pp. 238-255, 1971.

[20] Kittler, J., "Feature Selection and Extraction," *Handbook of Pattern Recognition and Image Processing* (T.Y.Young and K.S.Fu, eds.), New York:Academic, pp. 60-83, 1986.

[21] Lachenbruch, P.A. and Mickey, R.M., "Estimation of Error Rates in Discriminant Analysis," *Technometrics* **10**, 1, pp. 1-11, 1968.

[22] Lbov, G.S., "Logical Functions in the Problems of Empirical Prediction," *Handbook of Statistics* **2** (P.R. Krishnaiah and L.N. Kanal, eds.), North Holland, pp. 479-491, 1982.

[23] Lissack, T. and Fu, K.S., "Error Estimation in Pattern Recognition via L-distance Between Posterior Density Functions," *IEEE Trans. Inf. Theory* **IT-22**, pp. 34-45, 1976.

[24] McLachlan, G.J., "The Bias of the Apparent Error Rate in Discriminant Analysis," *Biometrika* **63**, pp. 239-244, 1976.

[25] McLachlan, G.J., "Assessing the Performance of an Allocation Rule," *Comp. and Maths. with Appls.* **12A**, pp. 261-272, 1976.

[26] McLachlan, G.J., "Error Rate Estimation in Discriminant Analysis: Recent Advances," in *Advances in Multivariate Statistical Analysis*, A.K. Gupta (ed.), pp. 233-252, Reidel, 1987.

[27] Miroshnichenko, L., (Dniepropetrovskij gornyj Institut, USSR), Personal communication, 1988.

[28] O'Neill, T.Y., "The General Distribution of the Error Rate of a Classification Procedure With Application to Logistic Regression Discrimination," *J. Am. Stat. Ass.* **75**, pp. 154-160, 1980.

[29] Pettis, K.W., Bailey, T.A., Jain, A.K., and Dubes, R.C., "An Intrinsic Dimensionality Estimator from Near-Neighbor Information," *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-1**, 1, 1979, pp. 25-37.

[30] Raudys, Š., "On the Problems of Sample Size in Pattern Recognition," *Proc. 2nd All-Union Conf. on Statistical Methods in Control Theory*, Moscow: Nauka, pp. 64-67, 1970 (in Russian).

[31] Raudys, Š., "Comparison of the Estimates of the Probability of Misclassification," *Proc. 4th Int. Conf. Pattern Recognition*, Kyoto, November 1978.

[32] Raudys, Š. and Pikelis, V., "On Dimensionality, Sample Size, Classification Error, and Complexity of Classification Algorithm in Pattern Recognition," *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-2**, 3, pp. 242-252, 1980.

[33] Raudys, Š., "The Influence of Sample Size on Classification Performance," *Statistical Problems of Control*, Issue 66, Vilnius: Inst. Math. and Cyb. Press, pp. 9-42, 1984 (in Russian).

[34] Raudys, Š. and Vaitukaitis, V., "Methods to estimate the probability of misclassification," *Statistical Problems of Control*, Issue 66, Vilnius: Inst. Math. and Cyb. Press, 1984.

[35] Raudys, Š., "On the Accuracy of a Bootstrap Estimate of the Classification Error," *Proc. 9th. Int. Conf. Pattern Recognition*, Rome, November 1988.

[36] Sethi, I.K. and Sarvarayudu, "Hierarchical Classifier Design using Mutual Information," *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-4**, pp. 441-445, 1982.

[37] Siotani, M., "Large sample approximations and asymptotic expansions of classification statistics," *Handbook of Statistics* **2** (P.R. Krishnaiah and L.N. Kanal, eds.), North Holland, pp. 61-100, 1982.

[38] Toussaint, G.T., "Bibliography on estimation of misclassification," *IEEE Trans. Inf. Theory* **20** , pp. 472-479, 1974.

[39] Vanichsetakul, N., "Tree structured classification via Recursive Discriminant analysis," *Ph.D. Thesis*, Univ. of Wisconsin, 1986.

[40] Vapnik, V.N., *Recovery of Dependencies From Empirical Data*, Springer Verlag, 1982.

[41] Wolverton C.T. and Wagner, T.J., "Asymptotically Optimal Discriminant Functions for Pattern Classification," *IEEE Trans. Inf. Theory* **IT-15**,2, pp. 258-265, 1969.

[42] Žvirėnaitė, D., "Criteria for Selecting the Informative features in Pattern Recognition," *Statistical Problems of Control*, Issue 74, Vilnius: Inst. Math. and Cyb. Press, pp. 76-103, 1986 (in Russian).