ESTIMATION OF CLASSIFIER'S PERFORMANCE WITH ERROR COUNTING

Jan Heikkilä
Institute of Photogrammetry and Remote Sensing
Helsinki University of Technology
002150 Espoo 15, Finland
e-mail : jani@fifille.hut.fi
Commission III

ABSTRACT

Because of the complexity of error estimation in statistical pattern recognition, error counting methodology has been extensively used for evaluating the performance of a classifier. The paper compares different error estimation procedures, both theoretically and experimentally by simulation.

The classical error counting methods, like resubstitution, hold-out, leave-one-out and bootstrapping, are compared with the risk averaging methods and with methods using the relationships of error and reject tradeoffs. These second generation estimators have two clear advantages. First, they are designed to have a smaller variance. Secondly, and more importantly, the test samples can be unclassified. Thus, the risk-averaging methods are economical and in addition the mislabelling problems in the test set is avoided. Finally they are well suited in non-parametric methods and thus provide a possible solution for a general classification system. The simulation studies show that this is possible to achieve.

1. INTRODUCTION

The problem of estimating the performance of a classifier is a traditional one in statistical pattern recognition. The performance is usually measured with some estimate of the error rate, the percentage of errors the system will produce. Even in the Bayesian case (known statistical models), the expected error rate is extremely hard to evaluate, and is known to have an analytical solution only in some special cases (see e.g. Devivjer & Kittler 1982, chapter 2). In practice the probability models must be inferred from available training data. If the true density functions are replaced by the corresponding estimates, an estimate for the Bayes error is achieved. This is usually called *the apparent error rate*. Even the interpretation of this measure is ambiguous. It is also known to be optimistically biased (Devivjer & Kittler 1982, chapter 10). The computation of the apparent error rate is extremely difficult involving numerical integration in complex regions, and often in high dimensional spaces. These difficulties have been pushing the practitioners and the theoreticians to look for more simple methods for estimating the error rate of a classifier.

If some finite number of test samples with known labelling is available, the so called *error counting* procedure is a natural choice for error estimation. That is, the test set is classified and the number of misclassifications is counted. The properties of this type of procedures have been extensively discussed in the literature and a multitude of methods have been developed. The limited number of labelled samples cause usually problems. Computation intensive methods have been developed to cleverly use the small number of samples so that both the design and the test sets can utilize all available pattern vectors (see e.g. Fukunaga 1990).

There is another alternative for replacing the computation of the apparent error rate. These methods are either based on the idea of *risk averaging* or they use the relationship between the error and reject rates, the so called *error-reject trade-off* (Devivjer & Kittler 1982). These estimators have two advantages over the traditional error counting procedures. First, their variance is designed to be smaller (large sample sizes). Second, they permit to use unclassified samples as test samples. In remote sensing applications this means that one can use all the pixels in the satellite image as a test set. In addition to the great economical benefit, the problem of outliers in the test set is avoided. A large test set is particularly essential to accurately evaluate a classifier with low error rate (Raudys & Jain 1991).

The difficulty, which arises in designing a pattern recognition system comes from the variety of choices available. A large number of classification rules, both parametric and non-parametric, have been developed. A multitude of optimization criteria for finding a minimum set of features with maximal discrimination capabilities have been established. At least fifteen different methods are available for empirical error estimation. The total number of classification methods is thus in the order or hundreds. One goal of this research project is to test by simulation, if a general rule can be found, which can be applied to most cases, when no a prior information about the structure of the probability density functions is given. A general rule implies the use of a nonparametric classifier. The optimality of nonparametric classifiers must be found in practice by empirical error estimation (Fukunaga 1990, chapter 7). This error estimation criteria should have as small bias (hopefully none) and as small variance as possible, and should be robust against outliers.

The paper is divided into five chapters. In chapter two we will discuss about the effect of finite sample sizes to the classifier design. This chapter is a review of the mostly theoretical results, which have been achieved until now in statistical pattern recognition. Chapter three discusses in detail about the methods of empirical error estimation. Altogether ten methods are discussed. The chapter can be considered as a review of empirical error estimation, and utilizes both theoretical and empirical results from the literature. In chapter four, the simulation results are described. The simulations are based on an extensive work. Totally, more than 80000 different cases have been studied (yet the test is limited). The most important trends of these simulations are listed in chapter four. Finally, chapter five will draw some conclusions.

2. ERROR ESTIMATION AND CLASSIFIER DESIGN

In this chapter we review the effect of finite sample sizes to the empirical error estimators and to the classifier design. In the analysis below, like in the simulations in chapter 4, we will restrict to two class cases.

2.1 Effect of finite sample sizes to empirical error estimation

The expected performance of a classifier degrades because of two sources: the finite number of samples used to design the classifier and the finite number of samples used to test the classifier. A theoretical analysis about the effects of both of these can be found from (Fukunaga 1990).

The effect of the finite number of test samples in the error counting approach can be directly derived from the binomial distribution

$$\begin{aligned} E_T\{\hat{\epsilon}\} &= \epsilon \\ Var_T\{\hat{\epsilon}\} &= \left(\frac{P_1}{N_1}+\frac{P_2}{N_2}\right)\cdot\epsilon_1(1-\epsilon_1) \text{ ,} \end{aligned} \tag{1}$$

where $E_T\{\hat{\varepsilon}\}$ is the expected value and $Var_T\{\hat{\varepsilon}\}$ the variance of the error estimate, ε is the true error rate, ε_1 is the true error rate of class 1, P_1 and P_2 are the prior probabilities and N_1 and N_2 are the sample sizes for both classes. The finiteness of the test set does not affect to the bias of the estimate, but produces a variance, which is the higher the smaller is the expected error rate.

The effect of a finite design set is much more difficult to analyze and the derivation goes far beyond the scope of this paper. The interested reader can find a detailed derivation from (Fukunaga 1990, p. 201-214). It is shown that the bias produced by a finite design set is always positive and the variance of second order approximation of a Bayesian classifier (assuming correct probability model is used) is zero. If the classifier is not Bayesian or higher order terms are used in the analysis, the variance is not anymore zero, and is dependent on the underlying density structures being proportional to $1/N_D^2$.

When considering the effect of independent test and design sets, the following may thus be concluded: The bias comes mainly from the finite design set, and the variance from the finite test set.

2.2 Effect of finite sample sizes in Classifier design

There is a large variety of classification rules. We consider here only those, which we have used in our simulations. The primary concern is the bias produced by the finite design set, because the variance of the error estimate comes primarily from the test set. Also the robustness against outliers is considered.

2.2.1 Parametric Classifiers If the density functions can be expressed in parametric form, corresponding classifiers are called parametric. Most often the density functions are described with the help of first and second order moments. Depending on the assumptions made, the decision boundaries are either of linear (linear classifier, equal covariance matrices) or of quadratic form (quadratic classifier, different covariance matrices).

In the simulations carried out, we have used classifiers based on the assumption of multivariate normal distribution. The classifiers are known to be asymptotically Bayesian, if normality assumption is valid. In this case, the effect of the finite design set can be analyzed theoretically (Fukunaga 1990, chapter 5). The drift from the validity of the normality assumption (modelling error) is harder to analyze. The effect of this drift was analyzed by simulation during this project, but this part is not reported here.

If the covariance matrices in both classes are equal to the identity matrix, an explicit formula for the bias caused by the finite design set can be derived. This is of interest to have some kind of feeling about the dependencies. For linear classifier the bias is (Fukunaga 1990, p. 211)

$$E_D^L\{\nabla\epsilon\} \simeq \frac{\nu_L}{N_D} \text{ ,} \tag{2}$$

where

$$\nu_L = \frac{e^{\frac{-\mu^T\mu}{8}}}{2\sqrt{2\pi\,\mu^T\mu}}\cdot\left[\left(1+\frac{\mu^T\mu}{4}\right)d-1\right] . \tag{3}$$

Correspondingly, the effect of the final design set to the quadratic classifier in this case is

$$E_D^Q\{\nabla\epsilon\} \simeq \frac{v_Q}{N_D}\ , \tag{4}$$

where

$$v_Q = \frac{e^{\frac{-\mu^T\mu}{8}}}{4\sqrt{2\pi\,\mu^T\mu}} \cdot \left[d^2 + \left(1 + \frac{\mu^T\mu}{2}\right)d + \frac{(\mu^T\mu)^2}{16} - \frac{\mu^T\mu}{2} - 1\right]. \tag{5}$$

In (2)-(5) N_D is the size of the design set, d dimensionality of the feature space and μ is difference between the mean vectors (μ_1-μ_2). If d>>1, from (2)-(5) it can be seen that the bias of a linear classifier is proportional to d/N_D and the bias of a quadratic classifier is proportional to d^2/N_D. In the case of linear classifier this means that a number of datavectors, which is a fixed multiple of the dimensionality of the feature space (N_D=c*d), is enough to compensate the effect of finite sample size. As can be expected this is not valid for the quadratic case. The size of the constant c is dependent on the separability between the classes (~the expected error rate). The higher is the expected error, the greater value should the constant have.

2.2.2 Nonparametric Parzen classifiers The nonparametric classifiers are appealing, because they do not do any prior assumptions about the density structures of the underlying problem. However, they are more computation intensive and are shown to be heavily biased in high dimensional spaces (see below). They also have some parameters to tune for achieving optimality. The tuning of these parameters is extremely important and can be done by experimental error estimation. These topics are discussed in this chapter. The detailed derivations can be found from (Fukunaga 1990, chapters 6 and 7).

A nonparametric density function can be estimated by the so called *Parzen method.* The corresponding classifier is called a Parzen classifier. The Parzen estimate of a density function can be expressed as

$$\hat{p}(x) = \frac{\sum_{i=1}^{N_D} k\cdot K(Q,x_i,x)}{N_D}\ . \tag{6}$$

where K is a kernel function, k determines the size of the kernel, Q is a metric used to compute distances and x_i:s are the sample vectors. The corresponding classifier compares the a posteriori probabilities to a decision threshold, t. An analysis of the effect of all these variables is needed.

A second order approximation of the expected value and variance of the Parzen density estimates can be shown to be (Fukunaga 1990, p.258-259)

$$E\{\hat{p}(x)\} = \frac{p(x) + p(x)\,\alpha(x)k^2}{2} \tag{7}$$

and

$$Var\{\hat{p}(x)\} = \frac{1}{N_D}\cdot g\left(p(x),\alpha(x),Q,p^2(x),k^2\right)\ , \tag{8}$$

where

$$\alpha(x) = tr\left\{\frac{\nabla^2 p(x)}{p(x)}Q\right\}\ , \tag{9}$$

and the precise form of (8) can be found from (Fukunaga 1990, p. 259). (7)-(9) show that only the variance is dependent on the sample size being proportional to $1/N_D$. Thus it can be reduced by increasing the sample size. On the contrary the bias is not at all dependent on the sample size and must be minimized by a proper selection of the parameters k and Q. From (9) we can see that both the bias and the variance are dependent on the curvature of the density structures.

Of course the bias and variance of the density estimates affect to the bias of the classification error. After some hard mathematics this can be expressed as

$$E\{\nabla\epsilon\} \simeq a_1k^2 + a_2k^4 + a_3k^{-\frac{d}{N_D}} - b\nabla t\ , \tag{10}$$

where a_1, a_2, a_3 and b are complicated functions depending on $\alpha_1(x)$-$\alpha_2(x)$, Q and p(x), and Δt is the bias of the decision threshold.

Although the constants a_1, a_2 and a_3 are complicated functions, they are only dependent on the probability structures and the metric, and totally independent on the kernel size and on the sample size. When k is small, the term a_3 is dominating. This term reflects the variance term, Var{p(x)}, of the density estimate. When k grows, terms a_1 and a_2 start to dominate. These terms reflect the effect of the bias term of the density estimate, E{p(x)}, to the classification error. Both terms should be adjusted properly.

When k is small the variance term of the density estimate dominates. Thus the bias can be reduced by increasing the sample size. However, on bigger values of k, the bias term of the density estimate dominates. This effect cannot be anymore reduced by increasing the sample size. The difficulty is overemphasized in higher dimensional spaces. When the intrinsic dimensionality of the feature space is high it becomes hopeless to increase the sample size (because the amount of samples needed grows so fast, Fukunaga 1990, p. 264), and the only choice to get a reasonable estimate is to increase k. However, we can see from (10) that a further reduction to the classification error can be achieved by choosing a proper threshold t, and form (9) that the metric Q and K also affect to the final bias.

The effect of the decision threshold can be optimized, if the density functions of the underlying problem are normal (Fukunaga, p. 329). For more general cases an optimal t is impossible to achieve analytically. However, the following experimental procedure based on empirical error estimation can be carried out. For each value of k, find the threshold, which produces the smallest experimental upper bound for the error. The value of t, which corresponds to the smallest upper bound can be chosen together with the corresponding k to the final classification.

Another way to compensate the effect of the bias terms is to consider the metric Q and the kernel shape K. From (9) and (10) we can see that the terms, which are independent of N_D can be compensated, if $\alpha_1(x)=\alpha_2(x)$. Hence, for each x, one should find a solution to

$$tr\left\{\frac{\nabla^2 p_1(x)}{p_1(x)}Q_1\right\} = tr\left\{\frac{\nabla^2 p_2(x)}{p_2(x)}Q_2\right\} . \qquad (11)$$

This is extremely hard to obtain. Only in the case of normal distribution a solution can be achieved, and if additionally the covariance structures are similar, a choice of $Q_i=\Sigma$, is reasonable. In the general case, if Q is of second order, an approximate solution can be accomplished by choosing (Fukunaga 1990, p. 337)

$$Q_i = \Sigma_i + \gamma_i(x-\mu_i)(x-\mu_i)^T , \qquad (12)$$

where γ_i should be chosen to be a little larger than

$$\frac{-1}{(x-\mu_i)(x-\mu_i)^T} . \qquad (13)$$

The effect of the metric is demonstrated in figure 1 (adopted from Fukunaga 1990), which shows how dramatic can the effect be on bigger kernel sizes.

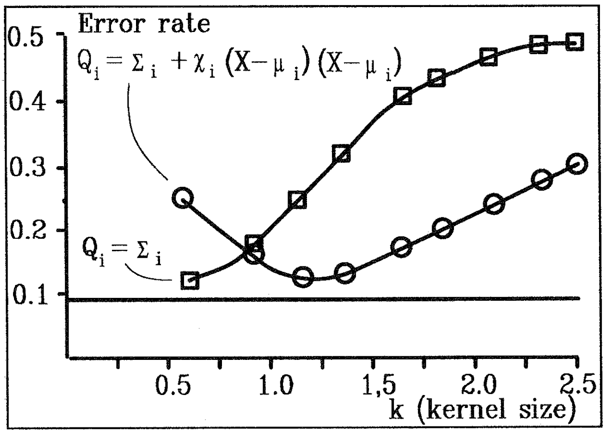

Figure 1. The effect of the metric to the estimated error rate, data Gaussian with unequal covariance matrices.

The kernel shape K is again a parameter to be choosed. It was experimentally shown in (Fukunaga 1990) that the more uniform type the kernel is, the more it will affect the lower bound for the error, but have only little effect on the upper bound. If Gaussian kernel is used, it may happen that too much emphasis is put to the centre of the kernel. A uniform kernel, like the hyperball or hypercube, is another choice. It puts equal emphasis throughout the neighbourhood. It is quite evident that this produces many a posterior probabilities in the border regions to be equal.

A computation intensive estimate of the Bayes error can be achieved with the Parzen method from (10)

$$E\{\hat{\epsilon}\} \simeq \epsilon + a_1 r^2 + a_2 r^4 + a_3 r^{\frac{-d}{N_D}} . \qquad (14)$$

After varying k, we can finally perform a least squares fit using model (14) and from that have an estimate, $\hat{\varepsilon}$, of the Bayes error ε. Because it is reasonable to believe that all the constants in (14) are positive, this can be used as a constraint in the estimation procedure. The fit should be repeated for each value of t, because for (14) to be valid, a Bayesian decision should be made in each case. The value of t, which gives the lowest estimate of error should finally be chosen.

2.2.3 Voting kNN classifiers In the well-known voting kNN procedure distances to the k nearest design samples are computed and the pattern vector is addressed to the class, which gets the majority of the votes. If ties occur, a reject option can be used like in the following analysis. This simple, though computing intensive method has become appealing because of the following result, which is achieved via the asymptotic analysis (as $N_D \rightarrow \infty$, $P(\hat{\alpha}_i|X_{kNN}) \rightarrow P(\hat{\alpha}_i|X)$, see Devivjer & Kittler 1982)

$$\frac{\epsilon}{2} \leq \epsilon^a_{2NN} \leq \ldots \leq \epsilon \leq \ldots \leq \epsilon^a_{NN} \leq 2\epsilon , \qquad (15)$$

where ε^a_{kNN} stands for the asymptotic error based on k nearest neighbours and ε stands for the Bayes error. According to this asymptotic result very tight bounds for the Bayes error can be achieved by applying a kNN classifier, if k is large enough.

However, in practice we have to cope with finite sized design sets and (15) will be heavily affected by the corresponding bias. The simplified presumption $P(\hat{\alpha}_i|X_{kNN})=P(\hat{\alpha}_i|X)$ does not hold and the corresponding risks will be biased. A detailed study of this bias in the NN and 2NN cases can be found from (Fukunaga & Hummels 1987a). They showed that a second order approximation of the bias of the voting NN procedure is

$$E\{\Delta\hat{\epsilon}_{NN}\} = \beta_1 \cdot E_x\left\{|Q|^{\frac{-1}{d}} tr\left(QB_1(x)\right)\right\} , \qquad (16)$$

where

$$\beta_1 \simeq \frac{\Gamma^{\frac{2}{d}}\left(\frac{d+2}{2}\right)\cdot\Gamma\left(1+\frac{2}{d}\right)}{d\pi} \cdot N_D^{\frac{-2}{d}} \tag{17}$$

and B_1 is a function depending on the probability structures and the metric used. Correspondingly, the second order approximation of the 2NN bias is

$$E\{\Delta\hat{\epsilon}_{2NN}\} = \beta_2 \cdot E_x\left\{\left[|Q|^{\frac{-1}{d}} tr(QB_2(x))\right]^2\right\}, \tag{18}$$

where

$$\beta_2 \simeq \frac{\left[\Gamma^{\frac{2}{d}}\left(\frac{d+2}{2}\right)\right]^2 \cdot\Gamma\left(1+\frac{4}{d}\right)\cdot\left(1+\frac{4}{d}\right)}{[d\pi]^2\cdot\left(1+\frac{2}{d}\right)} \cdot N_D^{\frac{-4}{d}} \tag{19}$$

and B_2 is again dependent on the probability structures of the underlying problem.

The effect of the sample size to the bias is dependent only on the constants β_1 and β_2. The bias seems to be very difficult to compensate by increasing the sample size. Luckily the bias of the 2NN case is much smaller than in the NN case. This can be seen from figure 2. This agrees with the fact that the NN method seems to give reasonable result in practice only in the lower dimensional cases.

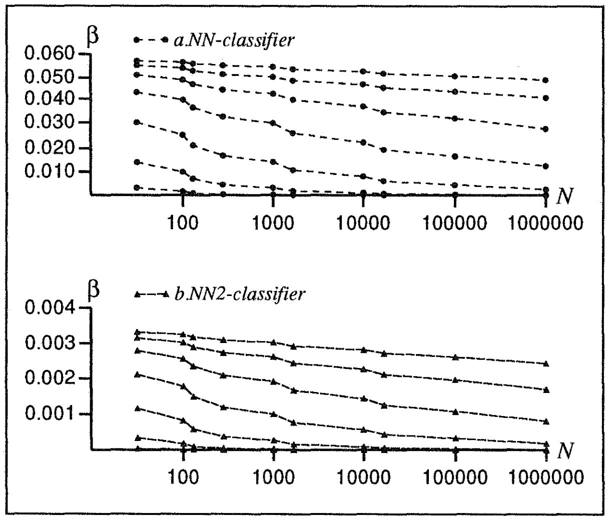

Figure 2. The bias terms of the NN and 2NN classifiers. The feature space dimension varies from 2 to 128 from bottom to top, respectively.

For larger values of k, the kNN procedure reminds closely the Parzen procedure. Correspondingly it gives similar results in all aspects of the error estimation problem (Fukunaga & Hummels, 1987b). For large values of k, the bias can be expressed by

$$E(\Delta\hat{e}) \simeq \frac{b_1}{k} + b_2\cdot\left(\frac{k}{N_D}\right)^{\frac{2}{d}} + b_3\cdot\left(\frac{k}{N_D}\right)^{\frac{4}{d}} - a\cdot\Delta t\ . \tag{20}$$

Equation (20) has a remarkable similarity with equation (10). The a_3 term is missing, because (20) is only valid for large values of k and the a_3 term vanishes when k is increasing. The constant term b_1/k is the only clear exception from (10). It shows that the kNN procedure converges to the Bayes error only for large values of k, even in the asymptotic case. A serious disadvantage of the *voting* kNN procedure is the fact that there is no possibility to adjust the decision threshold for reducing the bias.

2.2.4 Robustness of the classifiers The unbiasness of the estimated parameters of parametric density functions is shown in every statistical textbook. The effect of outlying observations has been a primary concern during the last decade (see e.g. Huber 1981). For parametric classifiers the traditional estimation techniques have been shown to be extremely sensitive to outliers, the *breakdown point* being asymptotically zero (a single outlier can cause the system to break down). However, the concept of the breakdown point is a little misleading, because it analyzes only the worst case. The robustness is naturally a function of the size and of the amount of outliers. A theoretical analysis of these effects is hard to carry out. To find out the dependence of the estimated error rate against these parameters, a simulation study is a feasible choice. We will shortly return to this subject in chapter 4.

In case of the nonparametric methods the outliers are not so serious, because only some local neighbourhood is used for density estimation. If none of the outliers does not belong to this neighbourhood, no damage will be created. The situation is worse in the case of labelling errors in the design set, because in that case remarkable violations will occur also in the non-parametric case. We will simulate both these cases in chapter 4.

Because we are more interested in non-parametric classifiers due to their generality, we will not concentrate in this context to the parametric robust estimation techniques.

3. EMPIRICAL ERROR ESTIMATION

In this context we classify the empirical error estimation methods into three categories: error counting methods, methods based on risk averaging and methods based on the error reject tradeoff. In this chapter we will shortly review the methods, which we have used in the simulations of chapter 4.

3.1 Error counting methods

Error counting methods have been traditionally used in pattern recognition for estimating the error rate. Many variates exists, from which we have used four in this context. The error counting estimate is given by

$$\hat{\epsilon}_{ec} = \frac{\sum_{x \in M} 1}{N_D} , \qquad (21)$$

where M is the set of misclassified samples. The methods differ from each other from the way they use the available sample set.

3.1.1 Resubstitution In the resubstitution method the whole sample set is used for designing and again for testing the classifier. This method is known to be optimistically biased (e.g. Devivjer & Kittler, chapter 10), thus giving a lower bound for the Bayes error (or for the asymptotic error). However the bias reduces then the sample size increases. Thus, if enough samples is in use, resubstitution method can be used.

3.1.2 Hold-out The holdout procedure is a special case of cross-validation, where the available sample set is divided into exactly two sets. The other set is used for designing the classifier and the other set as a test set. The method is known to give an upper bound for the Bayes error. Its variance is a little bigger than that of resubstitution. What makes it rather complicated from the practical point of view, is the demand for independent test and design sets. The division is far from a non trivial problem. The neglectance of some part of the available data from designing the classifier is not very pleasing, too. An arithmetic average of the resubstitution and holdout methods should be used as the final estimate of the error rate.

3.1.3 Leave-one-out The other extreme of cross-validation is given then each of the samples in turn is used for testing the classifier and the N_D-1 remaining samples are used as a design set. The available samples are thus more effectively utilized. Also the design and test sets are statistically independent. In case of independent sample vectors, the method is known to be practically unbiased (Efron 1983). For long time the method has been recommended to be used in the context of small sample sizes. Unfortunately, especially for small sample sets, the variance of the leave-one-out method is high. Because the variance component dominates in small sample sets (Efron 1983), a low variance estimate is recommendable in such cases. Another disadvantage of leave-one-out method is the high computational cost for some types of classification algorithms, because N_D different classifiers must be designed. Fortunately for many cases, recursive methods can be used to get the leave-one-out estimate practically with the same computation time as the resubstitution estimate (e.g. Fukunaga, 1990, chapters 5 and 7).

3.1.4 Bootsrapping A bootstrap design sample of exactly size N_D is generated by sampling with replacement. Two different estimates can be computed. In the *bootstrap resubstitution* estimate, the samples of the design set are used also for testing. The *bootstrap hold-out* estimate is achieved by using those samples for testing, which do not belong to the generated design set. The procedure is repeated 100-200 times (100 in the simulations of chapter 4) and the final estimate is the arithmetic mean of the individual estimates. Many modifications have been developed from this basic algorithm. The one, which in various empirical studies (e.g. Jain et al. 1987) has shown good performance, is called the *0.632 bootstrap* estimate. In this heuristic method a weighted average of the two bootstrap estimates is computed given weight 0.632 to the holdout estimate. Theoretically (Fukunaga 1990, chapter 5) the bias of the original bootstrap method should coincide with the bias of the leave-one-out method. However, the variance should be smaller, and close to the resubstitution estimate.

Some hints of the weakness of the bootstrap method have been reported. In (Chernick et al. 1985) the 0.632 bootstrap estimate showed weaker results when used with linear classifiers in high error rate situations. In (Jain et al. 1987) it was reported that the 0.632 estimator should not be used together with the NN classifier.

3.2 Methods based on risk averaging

The risk averaging methods are designed especially to have a low variance. A big advantage comes from the fact that an unlabelled test set can be used for testing the classifier. That is why the test set can be large (e.g. the whole satellite scene) and thus the method should be suitable for low-error rate cases. Because the variance of the error comes primarily from the test set, this already produces a low variance. In risk averaging the risk for each decision is estimated and the final error estimate is the average of all of the estimates. Thus

$$\hat{\epsilon}_{ra} = \frac{\sum_{i=1}^{N_T} \hat{\epsilon}_i}{N_T} , \qquad (22)$$

where $\hat{\epsilon}_i$ is the estimated risk for decision i, which equals to 1-max[$P(\omega_j|x_i)$]. Estimate (22) is sometimes called the grouped estimate. The performance of this estimate has not been fully studied in comparison with the error counting. Note that the method is especially suitable for nonparametric classifiers, where the risks are always computed. No modifications for the algorithms are needed.

It is shown in (Devivjer & Kittler 1982, chapter 10) that the estimate is unbiased and the variance of the method is (in case of equal prior probabilities)

$$Var\{\hat{\epsilon}_{ra}\} \leq \frac{\epsilon(1-\epsilon)}{N_T} - \frac{\epsilon}{2N_T} , \qquad (23)$$

which is less than half of the variance of error counting (compare to (1)).

3.2.1 Direct ("resubstitution") estimate A direct estimate will be formed by directly evaluating formula (22) for each test sample. Because of the design phase, the bias and variance of the estimate $\hat{\varepsilon}$ must be taken into account. This will effect the final estimate to be optimistically biased and the direct estimate thus gives a lower bound for the error.

3.2.2 Reference set ("holdout") estimate For getting an upper bound (and thus an estimate as the average) we must have another labelled set, which is now called the reference set. When this estimate is computed, the classification is taken from the design set, but the risk is computed with the help of the reference set. It is shown in (Devivjer & Kittler 1982) that this estimate is asymptotically unbiased and has a variance, which is in the order of (23).

3.3 Methods based on error-reject tradeoff

If a reject option is included to the classification system, the error rate of the classifier reduces, because the reject option discards those pattern vectors, whose classification has a high risk. The dependence between the reject and error rates is (see e.g. Devivjer & Kittler 1982)

$$\epsilon(\lambda_r) = -\int_0^{\lambda_r} \lambda \cdot dR(\lambda) \,, \qquad (24)$$

where R is the reject rate and λ_r is the rejection threshold (if risk is greater than λ_r reject decision is made). Thus observing the rejection rate and varying λ_r an estimate of ε can be computed.

These methods have all the advantages of the methods of the previous chapter, especially the one that an unlabelled test set can be used.

3.3.1 A method utilizing ordered sets We have used a special version of error-reject tradeoffs, which is specially designed for the voting knn procedures. It is shown in (Devivjer & Kittler 1982, chapter 3) by asymptotic analysis that the following bounds can be formed for the error rate

$$\frac{1}{2}\sum_{i=1}^{\kappa} \frac{A_{2i,s_i}}{2i-1} \leq \epsilon \leq \frac{1}{2}\sum_{i=1}^{\kappa} \frac{A_{2i,s_i}}{2i-1} + \frac{1}{2}A_{k-1,s_\kappa} \,, \qquad (25)$$

where A_{k,s_i} means the acceptance rate of kNN classifier with exactly s=i votes. (25) inherently measures the asymptotic probability of ties, which occur in 2iNN classifications and takes a series expansion of all of them.

Unfortunately the bias produced by the finite design sets (see (20)) is heavily deteriorating the asymptotically tight bounds of (25). However, it is hoped that the average of the upper and lower bounds will produce good results.

4. SIMULATION RESULTS

An extensive simulation work has been carried out. The primary goals of these simulations was: a) to compare risk averaging methods with error counting methods, b) to compare nonparametric and parametric classifiers, c) to test the robustness of the different error estimation methods and d) to get at least an idea about a classification system, which is general and gives as reliable error estimates as possible in all circumstances. Although the simulation study is extensive, it is still limited in many respects. This is unavoidable, because of the many parameters to be tested.

The generated datasets are listed on table 1.

Data ID	Optimal Classifier	Bayes error rate
II	Linear	0.251
I4	Quadratic	0.324
NN	Nonparametric	0.128

Table 1. The three types of data used in the simulations.

The datasets II and I4 are generated from normal distribution and the dataset NN from a mixture normal distribution. Unless otherwise stated the given Bayes error rates are the ones listed in table 1. The Bayes error rate of 0.128 for dataset NN represent the case when the generated dataset differs maximally from normal distribution. All simulations carried out are two class cases.

Risk averaging methods vs. error counting

As an example, the estimated error rates and standard deviations of the different error estimators for datasets II and NN in case of a two dimensional feature space are presented in Figures 3-6.

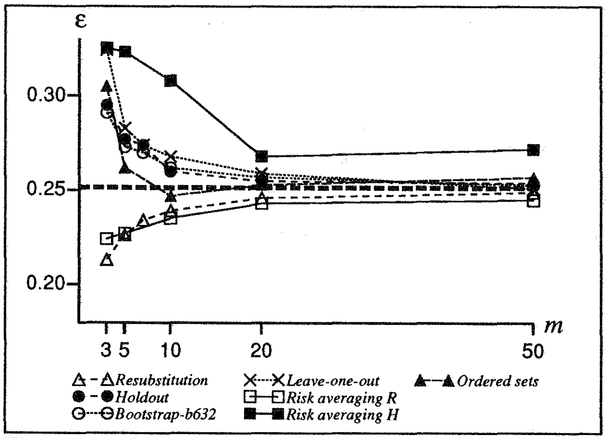

Figure 3. The estimated error rates of different estimators. Data II, Bayes error equals the dashed line, m*d=N_D.

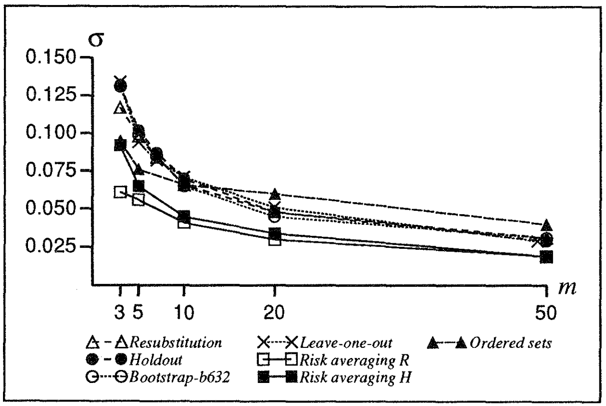

Figure 4. Standard deviations of the different error estimators, data II, m=d*N_D.

The results mostly agree with what was expected. The variances of the risk averaging methods are smaller than the variances of the error counting methods, especially in small sample size situations. This result is clearly in favour of the risk averaging in small sample situations, where the variance term mostly dominates. The upper bounds in both cases have a bigger variance than the lower bounds as expected. The use of the error reject tradeoff is comparable with risk averaging, but the difference between the upper and lower bounds (not shown in figures) is extremely big in small sample size situations. E.g. in dataset NN for the two smallest sample sizes the upper and lower bounds are 0.17 vs. 0.35 and 0.17 vs. 0.28, respectively. The convergence to the asymptotic case is slower than could be expected, which might come from the constant bias term of (20). However the mean of these bounds quite well predicts the error rate and the variance is nearly comparable to that of risk averaging.

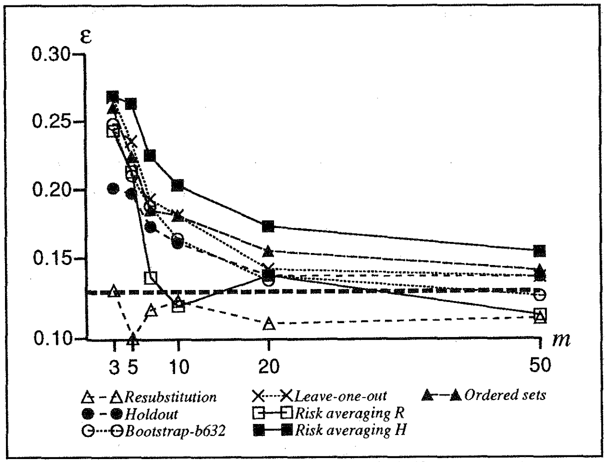

Figure 5. Estimated error rates of different estimators, dataset NN, dashed line = Bayes error, N_D=m*d.

One comment concerning the nonparametric results. The tuning of the kernel size of the nonparametric classifier was done in too rough a quantization in case of small sample sizes. That is the probable reason why some of the curves (e.g. resubstitution curve) do not behave smoothly. The bias term still dominates in some cases.

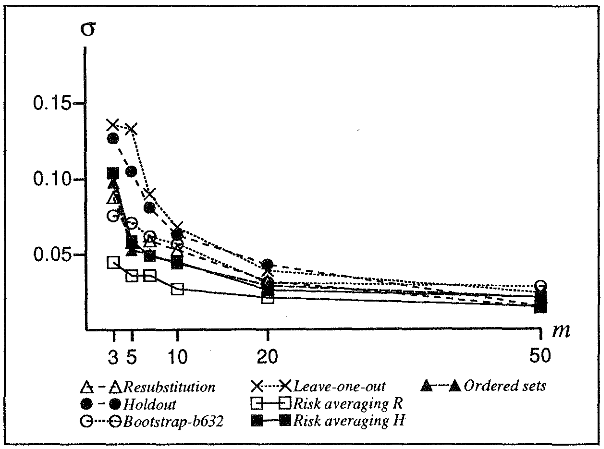

Figure 6. Standard deviations of different estimators, N_D=m*d.

In table 2 the different types of error estimation methods are compared with each other as a function of the Bayes error. All results are average values from the lower and upper bounds of the estimated Bayes error (e.g. error counting = ½[leave-one-out+resubstitution]). The following can be observed from the table. a) Risk averaging methods have dominantly smaller variance than the traditional methods. The difference is the bigger the smaller is the Bayes error and the smaller is the sample size. b) As has been illustrated in many simulations, bootstrapping method does not perform well in low error rate situations. c) The risk averaging methods and the method using error reject tradeoff are pessimistically biased in low error rate situations. These methods (also bootstrapping) works better under such circumstances, if the lower bound only is used (e.g. the lower bound for risk-averaging in the 0.01 error rate case with 5*d samples equals to the correct value 0.01, but the upper bound claims 0.02). The effect could be corrected by using a leave-one-out type procedure for the upper bound. d) The traditional

Method	Bayes error	m=5		m=10	
	ε	$\hat{\varepsilon}$	$\hat{\sigma}$	$\hat{\varepsilon}$	$\hat{\sigma}$
Error counting	25.1	25.6	10.6	25.4	6.6
Bootstrapping		27.3	9.9	26.2	6.5
Risk averag.		23.8	6.0	23.1	4.3
Error reject		26.1	7.8	24.7	6.6
Error counting	10.0	9.7	6.3	10.4	4.8
Bootstrapping		11.8	6.7	10.6	5.2
Risk averag.		9.9	3.0	9.8	2.2
Error reject		14.5	5.9	11.8	4.2
Error counting	5.0	4.9	5.0	5.0	3.3
Bootstrapping		6.4	4.8	5.5	2.9
Risk aver.		5.5	2.2	4.9	1.5
Error reject		7.5	3.3	5.8	2.6
Error counting	1.0	0.9	2.3	1.0	1.6
Bootstrapping		1.6	2.3	1.0	1.4
Risk aver.		1.5	1.1	1.2	0.6
Error reject		1.7	1.2	1.4	0.8

Table 2. Comparison of error counting methods as a function of the separability between classes, all numbers in percentages, m stands for sample size (m*d=N_D), d=8, linear classifier.

methods work quite nicely, if the sample size is high enough. However, the variance term is always about 2 times bigger than in risk averaging methods.

Parametric vs. nonparametric methods

Table 3 shows that the nonparametric method, if the parameter tuning is properly performed (see chapter 2), has potential also in the cases, where the optimal classifier is simpler. The opposite is of course not true. A linear classifier will never do proper work in datacase NN. Especially in higher dimensional spaces a successful application of a nonparametric classifier presumes that experimental parameter tuning is performed.

case	m=3	m=5	m=10
	$\hat{\varepsilon}$ $\hat{\sigma}$	$\hat{\varepsilon}$ $\hat{\sigma}$	$\hat{\varepsilon}$ $\hat{\sigma}$
Linear	.25 .13	.25 .10	.25 .07
Nonparam.	.27 .13	.26 .10	.25 .08
Quadratic	.33 .12	.32 .09	.32 .08
Nonparam.	.33 .11	.31 .08	.32 .08

Table 3. Comparison of a nonparametric classifier to the optimal ones in cases, where the asymptotically optimal classifiers is either linear (I1) or quadratic (I4), d=8, N_D=m*d.

In table 4 the robustness of the different type of classifiers are compared. The dataset is contaminated with outlying design samples, and the bias produced by the contaminated data is shown. As predicted the nonparametric methods are much more robust against the outliers. A closer look to the results (not shown in the table) reveals that the upper bound (leave-one-out estimate) of the nonparametric method grows when then percentage of outliers comes high, but the lower bound grows only moderately (the lower bound of the given example is 0.28 (ε=0.25) when 50% of outliers are present). In the parametric case both bounds grow equally fast.

p Δ	$\Delta\hat{\varepsilon}$ param.	$\Delta\hat{\varepsilon}$ nonparam.
5 30	0.107	0.003
25 30	0.194	0.036
50 30	0.206	0.084
25 5	0.060	0.038

Table 4. Robustness of classification methods against outliers, dataset II, d=4, m=20, p=percentage of outliers, Δ=size of outliers (multiple of feature standard deviation).

However, the type of outliers affect to the robustness. If the errors are labelling errors, also the nonparametric methods are more affected. An example is given in table 5. Again the lower bounds (resubstitution estimate in this case) are only moderately biased, but the upper bounds are strongly distorted.

Robustness of the error estimators

In table 6 the error estimators are compared with respect to the robustness. The numbers are again averages of the upper and lower bounds. For the risk averaging and error reject also the lower bounds are listed, because they are very robust against outliers.

p	$\Delta\hat{\varepsilon}$ param.	$\Delta\hat{\varepsilon}$ nonparam.
10	0.053	0.034
29	0.152	0.111
50	0.219	0.157

Table 5. Robustness of classification methods against outliers, dataset II, d=4, m=20, p=percentage of labelling errors.

The risk averaging method is extremely insensitive to labelling errors in the design set of a parametric classifier. As can be seen the bias is between 1 and 2 percent (true bayes error 25%). On the contrary, if the errors are just outliers lying far away from the true distribution, the risk-averaging tends to be strongly optimistically biased. This is because the scatter will spread out, but the unlabelled test set does not obey that distribution. Both the upper and lower bounds will be similarly biased.

In the nonparametric case, opposite to that of the parametric case, risk averaging is robust against outliers, but pessimistically biased in case of labelling errors. The same applies for the method utilizing the error-reject tradeoff. This is because of the upper bound. The lower bound is extremely robust in the nonparametric case. This phenomenon can be taken into advantage. If the difference between the upper and lower is big compared to the difference between the traditional error counting and the lower bound, the design set contains labelling errors, and the lower bound can be used for prediction.

Case	$\Delta\hat{\varepsilon}_{EC}$	$\Delta\hat{\varepsilon}_{RA}$	$\Delta\hat{\varepsilon}_{Er}$
Parametric			
p=10	0.05	0.01	
p=29	0.15	0.02	
p=50	0.22	0.01	
p=5 Δ=30	0.11	-0.11	
p=25 Δ=30	0.20	-0.13	
p=50 Δ=30	0.21	-0.18	
Nonparam.			
p=10	0.04	0.03 -0.04	0.06 0.01
p=29	0.12	0.10 -0.01	0.15 0.08
p=50	0.16	0.12 0.00	0.18 0.10
p=5 Δ=30	0.01	0.01 -0.03	0.02 -0.02
p=25 Δ=30	0.04	0.03 0.00	0.03 -0.01
p=50 Δ=30	0.09	0.04 0.00	0.04 0.00

Table 6. Robustness of different error estimators, p=% of outliers, Δ=size of outliers, labelling errors if none, EC=error counting, RA=risk averaging, Er=Error reject.

General classification system

According to the above results it seems possible to use the nonparametric classification method combined with the risk averaging methods in favour of a general classification system.

CONCLUSIONS

Empirical error estimation has been studied by simulation. The comparison was made between traditional error counting, risk averaging and a method utilizing the error-reject tradeoff. The error counting methods included resubstitution, holdout, leave-one-out and the bootstrapping method.

There are three reasons, why the risk averaging methods are recommended. First, it was confirmed that the variance of the risk averaging methods is superior to that of error counting. Secondly, because an unlabelled test set can be used, the method is economical and can always utilize a lot of test samples. Thirdly, and most importantly, the method is extremely robust against outliers, especially in the context of nonparametric classification. The use of the error reject tradeoff is also appealing because of the same reasons, but more research is needed to test it. On the other hand the MacLaurin expansion in the derivation of the elegant voting kNN modification is unfortunately not easily expandable to the multiclass case. This is in favour of the risk averaging. The bias of the direct risk averaging method causes it to act as a lower bound. This must be compensated somehow. The upper bound used in this project is based on the use of a holdout type estimate via a reference set. This method is not feasible from the practical point of view, because the method ignores half of the learning samples from the design. In this respect the usage of error reject tradeoff is more viable. Unfortunately its sample based upper and lower bounds are not at all tight and do not converge to the asymptotic case unless the kernel size, $k \rightarrow \infty$. A leave-one-out type of an estimator for the upper bound of the risk averaging could be economically established in the context of nonparametric estimation, because the design set uses only a local neighbourhood. It is hoped that in this case the upper bound behaves more nicely.

The simulations confirmed that a nonparametric classifier, if it is properly tuned, can perform as well as a parametric one, even in the case the prior information favours a simple linear classifier.

The primary goal of this simulation study was to test, if a general classification system (performing well in most cases) can be found so that a designer does not have to choose from so many different possibilities. The recommended system consists of: a) *A nonparametric classifier*, preferably a Parzen classifier, because it is easier to tune. It is of utmost importance to optimize all the parameters of a Parzen classifier. This concerns both the kernel shape and size, and the decision threshold. This optimization can be done via empirical error estimation. The error estimation should be done via risk averaging methods, which have a low variance and are robust against outlying observations, especially when nonparametric methods are used.

The extension of these results to multiclass cases is a demand for future research.

REFERENCES

+ Chernick, M., Murthy, V., Nealy, C., 1985: Application of the Bootstrap and Other Resampling Techniques: Evaluation of Classifier Performance. Pattern Recognition Letters, vol. 3, pp. 167-178.
+ Devivjer, P., Kittler, J., 1982: Pattern Recognition : A Statistical Approach. Prentice Hall.
+ Efron, B., 1983: Estimating the Error Rate of a Prediction Rule. Journal of American Statistical Association, vol 78, pp. 316-333.
+ Fukunaga, K., 1990: Introduction to Statistical Pattern Recognition, Academic Press.
+ Fukunaga, K., Hummels, D., 1987a: Bias of Nearest Neighbour Error Estimates, IEEE PAMI, vol. 9, no. 1, pp. 103-112.
+ Fukunaga, K., Hummels, D., 1987b: Bayes Error Estimation Using Parzen and k-NN Procedures. IEEE PAMI, vol. 9, no. 5, pp. 634-643.
+ Huber, P., 1981: Robust Statistics, Wiley, New York.
+ Jain, A., Dubes, R., Chen, C., 1987: Bootstrap Techniques for Error Estimation. IEEE PAMI, vol. 9, pp. 628-633.
+ Raudys S., Jain, A., 1991: Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practioners. IEEE PAMI, vol 13, no. 3, pp. 252-264.