An Experimental Study of the Optimal Class-Selective Rejection Rule

Thien M. HA
University of Berne
Institut für Informatik und Angewandte Mathematik
Neubrückstr. 10, CH-3012 Berne, Switzerland
Phone: +41 / 31 / 631 86 99
Fax.: +41 / 31 / 631 39 65
E-Mail: haminh@iam.unibe.ch

March 11, 1996

Abstract

This report reviews various class-selective rejection rules for pattern recognition. A rejection rule is called class-selective if it does not reject an ambiguous pattern from all classes but only from those classes that are most unlikely to issue the pattern. Both optimal and suboptimal rules, e.g. top-n ranking, are considered. Experimental comparisons performed on the recognition of isolated numerals from the NIST databases show that the optimal class-selective rejection rule is actually better than two other heuristic rules.

CR Categories and Subject Descriptors: I.5.0 [Pattern Recognition]: General; I.5.1 [Pattern Recognition]: Models; I.5.2 [Pattern Recognition]: Design Methodology; I.5.m [Pattern Recognition]: Decision.
Key Words: classification, decision rule, Bayes rule, Chow's rule, neural networks.

Contents

1 Introduction

In statistical pattern recognition, the probability that a given sample or pattern x belongs to the i^{th} class, in a N-class problem, is provided by the *posterior* probability $P(i/x)$ through the Bayes formula:

$$P_i(x) \equiv P(i/x) = \frac{p(x/i) \cdot \pi_i}{p(x)}; i = 1, .., N \tag{1}$$

where $p(x/i)$ is the i^{th} class conditional probability density function (p.d.f.), π_i is the *a priori* probability of observing the i^{th} class, $\sum_{i=1}^{N} \pi_i = 1$, and[1]

$$p(x) = \sum_{j=1}^{N} p(x/j) \cdot \pi_j \tag{2}$$

is the absolute probability density function [4, 5]. It follows immediately that the *posterior* probabilities sum up to 1, i.e.,

$$\sum_{i=1}^{N} P_i(x) = 1 \tag{3}$$

The connection between classification and decision is illustrated in Fig. 1, for a three-class problem. Here, the (soft) classifier is a device that computes the *posterior* probabilities, for a given x, which is then dichotomised into classes by the decision process. Assuming that $\{P_i(x); i = 1, .., N\}$ are known, the *decision rule* is designed so as to optimise some criterion, e.g. minimising the error rate. The resulting optimum decision rule constitutes a theoretical limit that any particular system attempts to reach but can never exceed. The well-known Bayes rule is a typical example, where it is implicitly assumed that the *posterior* probabilities are exactly known. Note that some authors take into account the classification method in the design of the decision rule [7, 3]. However, the former approach – optimise the decision rule assuming that the *posterior* probabilities are known – still provides the theoretical limit that the latter attempts to reach.

In most practical applications, $\{P_i(x); i = 1, .., N\}$ are unknown but can be estimated from a set of labelled patterns, called training or learning set. Many estimation methods exist, e.g. Parzen estimate, nearest neighbour, potential functions, and neural networks [4, 5, 8]. When the estimated functions are used instead of the true (unknown) functions, the optimality of the decision rule is no longer guaranteed. Therefore it is important to test a decision rule not only under ideal conditions but also under realistic ones, in which estimated *posterior* probabilities – obtained from training patterns – are used.

In this report, we test a newly introduced decision rule, namely the optimum class-selective rejection rule [6]. The considered problem is that of handwritten

[1] Without loss of generality, it will be assume that $p(x)$ is nonzero over the entire pattern space X, otherwise the region over which $p(x)$ is zero is first deleted.

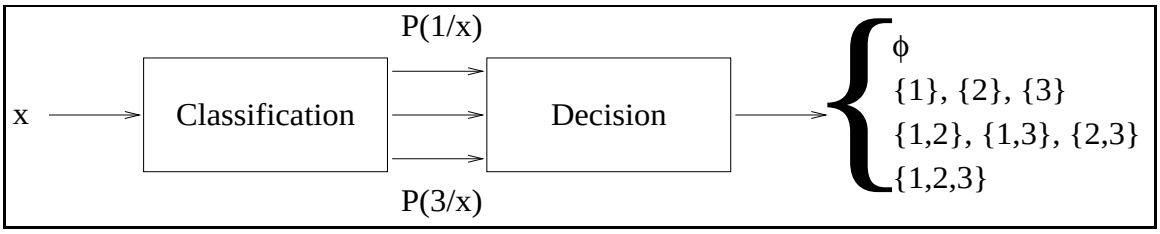

Figure 1: Relation between classification and decision. All possible outcomes of the decision process are shown on the right side.

numeral recognition (ten classes) and the estimation method to approximate the *posterior* probabilities is based on neural networks. In particular, we compare the results of the optimum decision rule to those obtained by two heuristic rules, namely, constant risk and top-n ranking. The next section reviews all three class-selective rejection rules and Section 3 compares their performance.

2 Class-Selective Rejection Rules

We briefly recall the concept of class-selective rejection by relating it to other well-known decision rules, namely, Bayes rule and Chow's rule. Then three class-selective rejection rules are presented, the first of which is optimum (in the sense defined below) whereas the remaining two (constant risk and top-n ranking) are heuristic.

Different existing decision rules differ from one another mainly in the choice of the decision outcomes. In the Bayes rule, the possible outcomes of the decision process are limited to the singletons, i.e., subsets that are formed by exactly one class each. They are $\{1\}$, $\{2\}$, and $\{3\}$ for a three-class problem. Fig. 2a illustrates the partition of the pattern space X into three regions, each of which corresponds to a single class, when the Bayes rule is used. The Bayes rule assigns to pattern x the class that has the highest *posterior* probability. It is known that this rule is optimal in the sense that no other rules can yield a lower error probability, or error rate.

The Bayes rule has also been modified by Chow to cope with a reject option [1, 2]. The idea is that when a pattern lies on or near a separation plane between two classes, the assignment to one or the other class is merely a guess. In such a case, it may be better to withhold making the assignment (decision) and to reject the input pattern. Thus, the outcomes of Chow's rule are also singletons, like in the Bayes rule, but augmented by the empty set $\emptyset$, which represents the reject option; see Figs. 1 and 2b. With a reject option, the optimality espouses another meaning, that of a tradeoff between the error rate and the reject rate (reject probability). More specifically, Chow's rule minimises the error rate for a given reject rate, or vice versa. The rule simply consists in rejecting the pattern if its highest *posterior* probability is lower than some threshold $(1-t), t \in [0, 1-\frac{1}{N}]$; otherwise, the decision is identical to Bayes' one, i.e. choosing the best class.

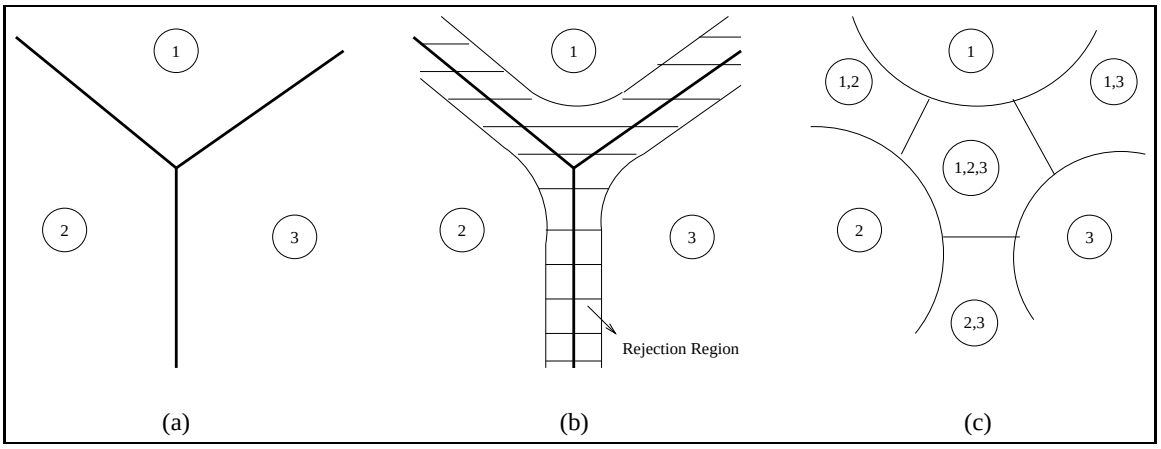

Figure 2: Three decision types.

Now, let us consider another family of rejection rules, which differ from Chow's rejection rule in that the outcomes of the decision process are extended to the power set of the set of classes, while excluding the empty set $\emptyset$. These rejection rules are called *class-selective* because they do not reject the pattern from all classes but only from those classes that are most unlikely to issue the pattern. (Recall that Chow's rule rejects a pattern if its highest *posterior* probability is lower than a given threshold, disregarding the probability distribution of the remaining classes.) For instance, for a pattern lying on the separation plane between classes 1 and 2, while being very far away from the center of the third class, the rule rejects only the third class and declares that the pattern belongs to the group composed of the first and the second classes. In other words, the pattern space is partitioned into regions each of which corresponds to a subset of classes. Since there are 2^N subsets in a set of N elements, the resulting partition may comprise up to $2^N - 1$ regions, excluding the empty set, in a N-class problem. In Fig. 2c, there are $2^3 - 1 = 7$ regions corresponding to the subsets $\{1\}$, $\{2\}$, $\{3\}$, $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, and $\{1,2,3\}$.

For class-selective rejection rules, an error occurs if the true class of a pattern belongs to the rejected classes, or equivalently, the true class does not belong to the selected classes. For example, if a pattern from class 2 is assigned to subset $\{1,3\}$, then the decision is wrong. It is clear that the more number of classes the decision rule chooses, the lower the error rate will be. In the limit, assigning every pattern to the set of all classes nullifies the error rate. However, such a rule would be useless because it corresponds to a trivial partition of the pattern space, i.e., to assigning the whole pattern space to the set of all classes.

In order to define the optimality of the class-selective rejection rule while avoiding the trivial partition, an additional constraint – the average number of classes $\bar{n}$ – was introduced [6].

$$\bar{n} = \int_X n(x)p(x)dx \tag{4}$$

where $n(x)$ is the number of classes assigned to pattern x. The choice of $\bar{n} =$

$E_X[n(x)]$ is natural, and more importantly, it can be directly estimated from experiments by the sample mean $\frac{1}{N_s}\sum_{i=1}^{N_s} n_i$, where n_i is the number of classes assigned to pattern x_i, and N_s is the total number of patterns involved in the experiment.

The optimality of the class-selective rejection rule is then defined as the rule that minimises the error rate for a given average number of classes. The error rate is given by

$$e = \int_X risk(x)p(x)dx \tag{5}$$

where $risk(x)$ is the (conditional) probability of making a wrong decision, for a given x.

$$risk(x) = 1 - \sum_{i \in Selected\ Subset} P_i(x) = \sum_{i \in Rejected\ Subset} P_i(x) \tag{6}$$

In the following, we present three class-selective rejection rules that will be used for comparison in Section 3. All three rules share one common feature: they select the best classes in terms of *posterior* probability. They differ from each other in the way the number of best classes are selected for a given pattern.

2.1 Optimum Rule

The optimum class-selective rejection rule assigns to pattern x all classes whose *posterior* probability is greater than a pre-specified threshold t. If there exist no such classes, the rule simply selects the (a) single best class [6]. The domain of the pre-specified threshold is

$$0 \leq t \leq \frac{1}{2} \tag{7}$$

When $t = \frac{1}{2}$, the rule is equivalent to Bayes rule, i.e., select the (a) single best class. When $t = 0$, we obtain the trivial partition of the pattern space. In between, the rule dynamically selects an appropriate number (between 1 and N) of best classes.

It should be clear the time complexity of this rule is $\mathcal{O}(N)$.

2.2 Constant Risk Rule

This is a heuristic rule and consists in selecting the smallest number of best classes such that $risk(x)$ is lower than a pre-specified threshold t_1. The domain of the pre-specified threshold is

$$0 < t_1 < 1 - \frac{1}{N} \tag{8}$$

and $risk(x)$ is computed as follows.

Let us introduce the sequence $\{Q_i(x)\}$, which is simply a reordered sequence of $\{P_i(x)\}$ in decreasing order of *posterior* probability

$$\{P_i(x); i = 1, .., N\} \rightarrow \{Q_i(x); i = 1, .., N\} / Q_i(x) \geq Q_{i+1}(x); i = 1, .., N-1 \tag{9}$$

Thus, $Q_1(x)$ is the maximum *posterior* probability of pattern x. The risk defined by Eq. (6) can then be expressed in a more explicit form as follows

$$risk_n(x) = 1 - \sum_{i=1}^{n} Q_i(x) \tag{10}$$

where n is the number of classes that is selected by the rejection rule, for a given pattern x. Obviously, $risk_n(x)$ decreases from $(1 - Q_1(x))$ down to 0 as n increases from 1 to N. For instance, in Fig. 3, the constant risk rule selects the two best classes because $n = 2$ is the smallest value that makes $risk_n(x) < t_1$, where t_1 is an arbitrary prespecified threshold.

Like the optimum rule, the constant risk rule is also dynamic in that the number of selected classes changes with pattern x. Unlike the optimum rule, the constant risk rule needs sorting, so the time complexity is $\mathcal{O}(N \cdot log(N))$ on the average and $\mathcal{O}(N^2)$ in the worst case [9].

2.3 Top-n Ranking Rule

The top-n ranking rule works as follows. For $n = 1$, select the (a) single best class, i.e., using the Bayes rule. For $n = 2$, select the best and the second best classes, and so on.

Unlike the two previous rules, the top-n ranking rule is static in the sense that the number of selected classes does not change with pattern x. The number of best classes, n, is chosen *a priori* and does not depend on the *posterior* probabilities of pattern x. Due to its simplicity, top-n ranking is by far the most used rule in practice. Generally, top-n ranking needs sorting, so the time complexity is also $\mathcal{O}(N \cdot log(N))$ on the average and $\mathcal{O}(N^2)$ in the worst case, like the constant risk rule.

3 Experiments

In this section we compare the performances of the above three class-selective rejection rules for the problem of handwritten numeral recognition. The comparison is based on the error-(average number of classes) tradeoff curves.

3.1 Database

Two databases, namely, SD3 and SD7, were provided by the American National Institute of Standards and Technology (NIST) in 1992 as parts of a conference to assess the state-of-the-art in isolated handwritten character recognition [10]. Twenty-nine groups from Europe and North America participated to compare the performance of their OCR systems. In total, 47 systems, both commercial and research, were presented. The databases contain isolated numerals (digits) as well as upper- and

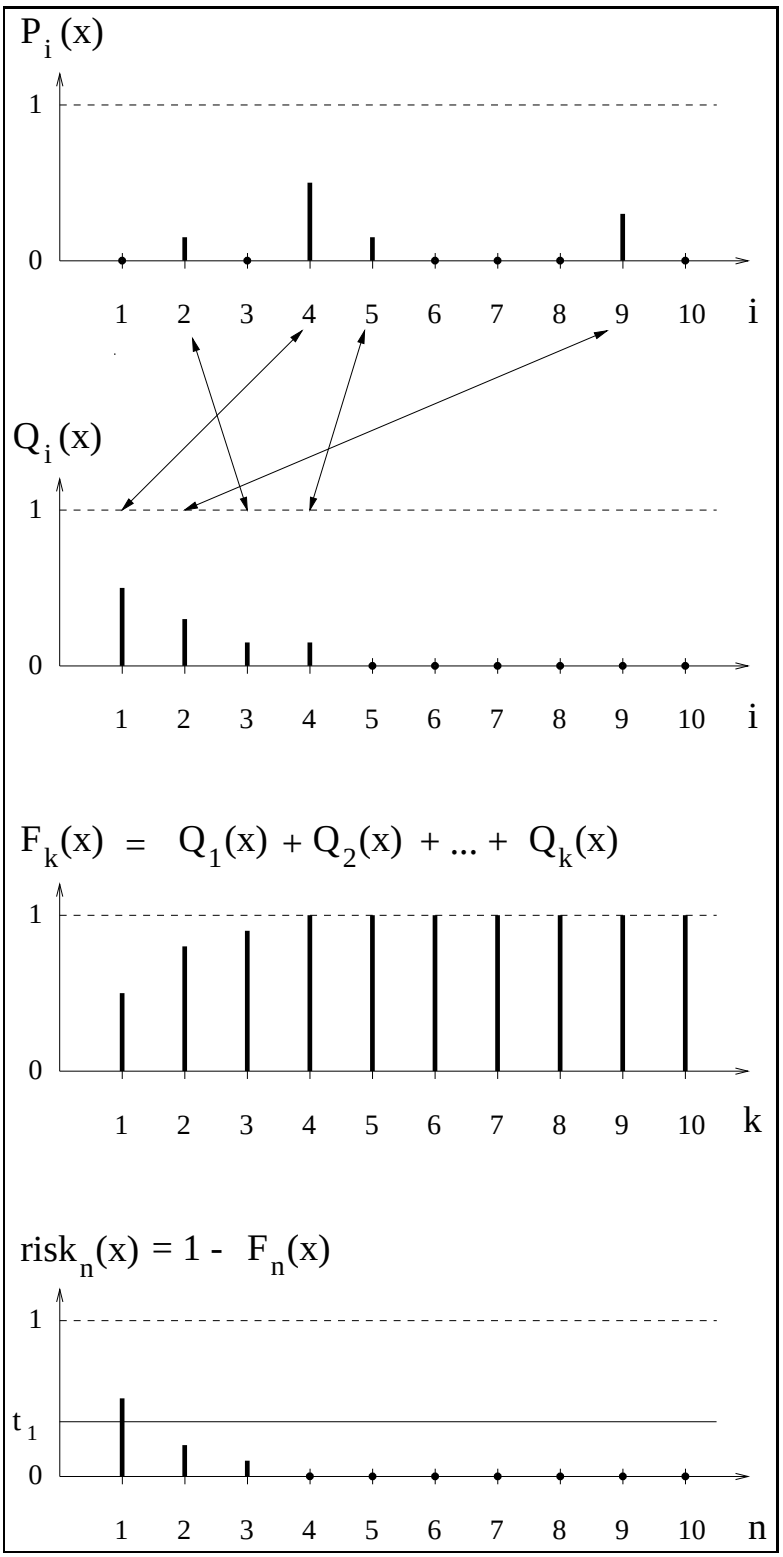

Figure 3: Illustration of $risk_n(x)$.

lower-case letters. In this report, we describe only experiments involving isolated numerals from SD3 and SD7, which contain 223124 and 58646 numerals, respectively. At the conference, most systems used SD3 for training and all used SD7 for testing; a few of them were trained using proprietary databases. The best system among those that used SD3 for training achieved an error rate of 3.16% at zero rejection level.

3.2 A Numeral Recognition System

The recognition system used in our study is a combination of two subsystems via a simple class-wise *posterior* probabilities summation scheme. Let $\hat{P}_i^{(1)}(x)$ and $\hat{P}_i^{(2)}(x)$ be the estimated *posterior* probabilities of class i by subsystems 1 and 2, respectively. The estimated *posterior* probability of class i for the combined system is obtained by

$$\hat{P}_i(x) = \frac{1}{2}[\hat{P}_i^{(1)}(x) + \hat{P}_i^{(2)}(x)]; i = 1, .., N. \tag{11}$$

Subsystem 1 estimates the *posterior* probabilities by first extracting a projection-based feature vector from the input pattern and then feeding it to a fully connected feed-forward multi-layer perceptron with architecture $49 : 60 : 10$ (60 hidden nodes). Subsystem 2 estimates the *posterior* probabilities by first extracting a contour-based feature vector from the input pattern and then feeding it to a fully connected feed-forward multi-layer perceptron with architecture $104 : 60 : 10$ (60 hidden nodes). See Appendix for details about the two feature extraction methods. Both neural networks are trained with the back-propagation algorithm [8].

Both neural networks were trained on the first 40000 numerals from SD3 and the next 10000 numerals were used to control the stopping of the training process. The test was performed on SD7 and achieved an error rate of 3.20% at zero rejection level, which is comparable to the best system presented at the NIST conference.

3.3 Comparison of Class-Selective Rejection Rules

For the optimum rule, the threshold t is varied within its appropriate range according to Eq. (7). For each value of the threshold, the optimum rule is applied to all test patterns yielding an error rate and an average number of classes. The produced pairs allow us to plot the error-(average number of classes) tradeoff curve. The procedure for the constant risk rule is similar (the range of threshold t_1 is given by Eq. (8)). For the top-n ranking rule, n is varied (discretly of course) from 1 to N.

Fig. 4 shows the tradeoffs between error rate and average number of classes for all three class-selective rejection rules. Of course, $\hat{P}_i(x)$ were used instead of the (unknown) $P_i(x)$. Fig. 5 shows the same curves but on a logarithmic scale for the error rate.

It can be seen that the top-n ranking rule is largely sub-optimum compared to the optimum rule. For instance, for $\bar{n} = 2$, the former yields an error rate of more

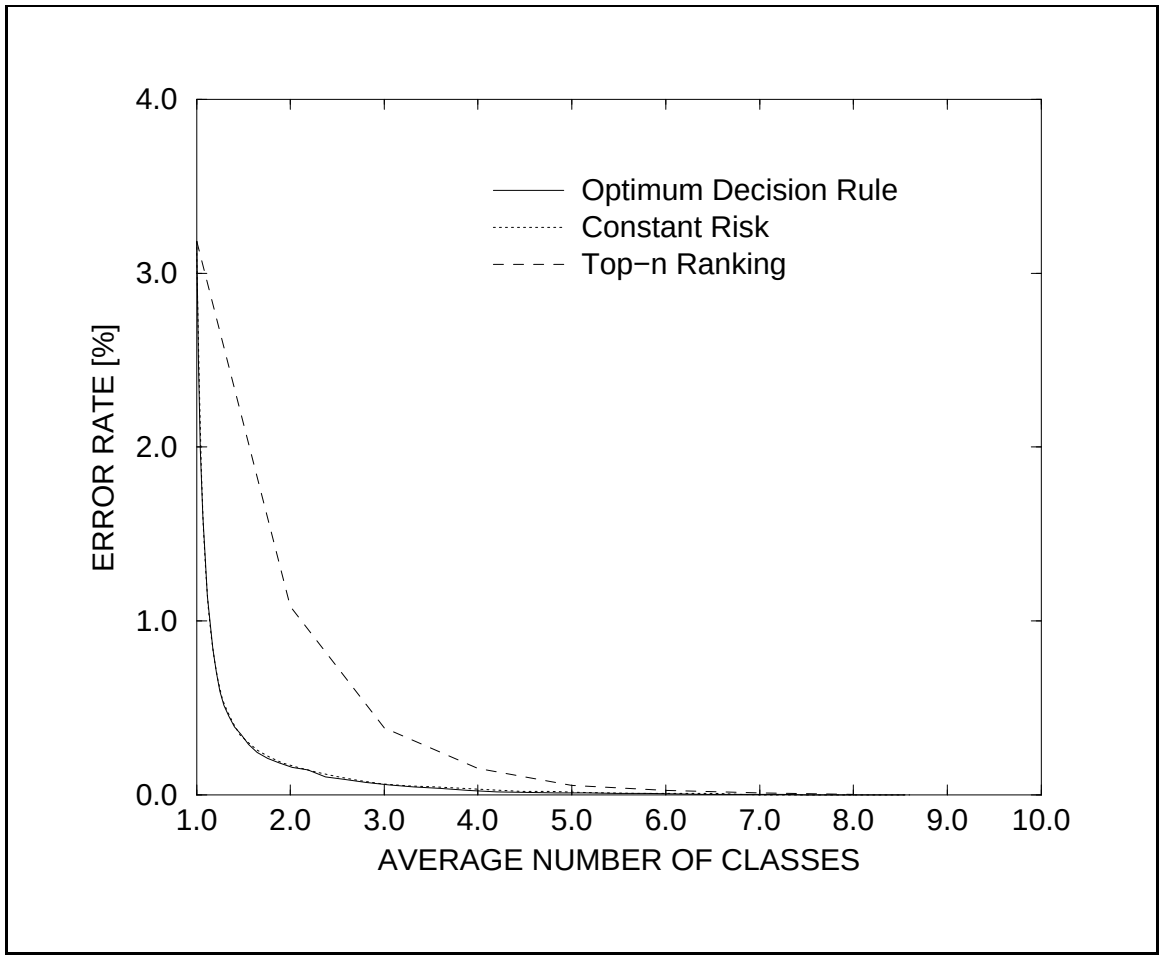

Figure 4: Tradeoff between error rate and average number of classes.

than 1% whereas the latter reduces it to less than 0.2%. The optimum rule thus reduces the error rate by a factor five, or 80%.

The constant risk rule performs nearly as well as the optimum rule in this problem. For large values of $\bar{n}$, the curves are no longer stable because the number of errors becomes very small, typically on the order of $0.01\% \cdot 58646 \approx 12$ numerals, for $\bar{n} = 6$. It is clear that no reliable statistics can be done on such a small sample size.

4 Conclusion

In pattern recognition, decision rules are usually optimised by assuming that the *posterior* probabilities are exactly known. In most practical applications, these probabilities are unknown and must be estimated from a set of labelled patterns. When the estimated probabilities are used instead of their true values, the optimality of the decision rule is no longer guaranteed. To justify the use of a decision rule in

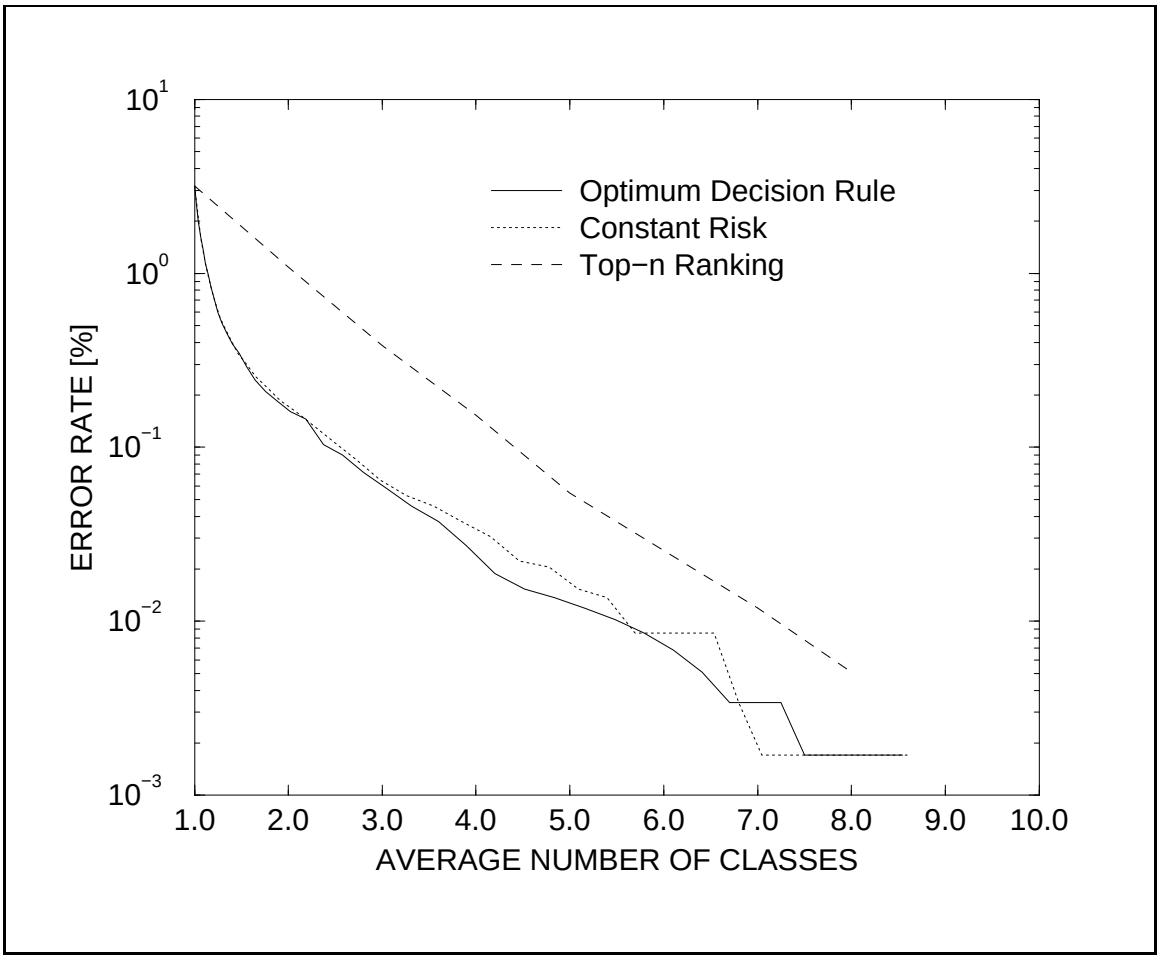

Figure 5: Tradeoff between error rate and average number of classes.

an application, it is therefore necessary to test the decision rule by using the estimated *posterior* probabilities. We have performed such a test on the optimum class-selective rejection rule for the problem of handwritten numeral recognition. Experimental comparisons using the NIST databases suggested that the optimum rule is actually better than two other heuristic rules. In particular, the commonly used top-n ranking rule was shown to be largely suboptimal in this problem.

Acknowledgements: This work was partly supported by the Union Bank of Switzerland Information Technology Laboratory (UBILAB) and the Swiss National Science Foundation. The author would like to thank Guido Kaufmann, Bernard Achermann, and Dieter Niggeler for numerous instructive discussions as well as proofreading. Special thanks go to Prof. H. Bunke for his constant encouragement.

A Feature Extraction

A.1 Projection-Based Features

The input image is normalised to a 32×32 binary image. Black pixels are projected in 4 main directions resulting in 4 histograms. Similarly, we count the number of black-to-white transition pixels in 4 main directions resulting in another 4 histograms. In addition, 8 contour profiles are computed from 8 main directions (a contour profile value is defined as the number of white pixels separating the border and the 1^{st} black pixel seen from a given direction). Thus, we obtain 16 one-dimensional functions from each of which 3 features are extracted by sub-sampling with overlapping cosinusoidal windows. The last component of the feature vector is the aspect ratio of the input image. Thus, a 49-dimensional feature vector $(16 \times 3 + 1)$ is obtained.

A.2 Contour-Based Features

The input image is normalised to a 72×54 binary image. Both inner and outer contours of the numeral are extracted. At each contour point the direction is estimated and quantised into 8 uniform quantisation intervals. The normalised space is then subdivided into $4 \times 3 = 12$ overlapping regions. In each region and for each of the 8 discrete directions, the total number of contour pixels is counted. The counting is weighted according to its position with respect to the center of the corresponding region using a separable 2-dimensional cosinusoidal window. In addition, the global 8-directional contour histogram is computed. This process yields a 104-dimensional feature vector $(8(12 + 1))$.

References

[1] C.K. Chow, "An Optimum Character Recognition System Using Decision Functions," *Institute of Radio Engineers (IRE) Transactions on Electronic Computers,* Vol. EC-6, No. 4, pp. 247-254, December 1957.

[2] C.K. Chow, "On Optimum Recognition Error and Reject Tradeoff," *IEEE Transactions on Information Theory,* Vol. IT-16, No. 1, pp. 41-46, January 1970.

[3] P.A. Devijver, "Error and Reject Tradeoff for Nearest Neighbor Decision Rules," in G. Tacconi (Ed.) *Aspects of Signal Processing,*, Part 2, D. Reidel Publishing Company, Dordrecht-Holland, pp. 525-538, 1977.

[4] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.

[5] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second edition, Academic Press, 1990.

[6] Thien M. Ha, "An Optimum Decision Rule for Pattern Recognition," Technical Report IAM-95-009, Institute of Computer Science and Applied Mathematics, University of Berne, Switzerland, November 1995.

[7] M.E. Hellman, "The Nearest Neighbor Classification Rule with a Reject Option," *IEEE Transactions on Systems, Science, and Cybernetics,* Vol. SSC-6, No. 3, pp. 179-185, July 1970.

[8] C.G.Y. Lau (Editor), *Neural Networks: Theoretical Foundations and Analysis,* IEEE Press, 1992.

[9] R. Sedgewick, *Algorithms,* Addison-Wesley, 1988.

[10] R.A. Wilkinson, J. Geist, S. Janet, P.J. Grother, C.J.C. Burges, R. Creecy, B. Hammond, J.J. Hull, N.W. Larsen, T.P. Vogl, and C.L. Wilson, *The First Census Optical Character Recognition Systems Conference,* The U.S. Bureau of Census and the National Institute of Standards and Technology, Technical Report #NISTIR 4912, Gaithersburg, MD, Aug. 1992.