Rejection methods for an adaptive committee classifier

Matti Aksela[1,2], Jorma Laaksonen[1], Erkki Oja[1]
Helsinki University of Technology
Neural Networks Research Centre
P.O.Box 5400, Fin-02015 HUT, Finland
{matti.aksela,jorma.laaksonen,erkki.oja}@hut.fi

Jari Kangas
Nokia Research Center
P.O.Box 100, Fin-33721 Tampere, Finland
jari.a.kangas@nokia.com

Abstract

Adaptation is an effective method for improving classification accuracy and a committee structure can in general improve on its members' performance. Therefore an adaptive committee structure is a tempting approach. Rejection may be used in handwriting recognition to improve performance through either directing the problematic character to a special classifier that handles such hard cases or discarding it. The experiments in this paper compare several fundamentally different approaches to implementing rejection in an adaptive committee classifier. A Dynamically Expanding Context (DEC) - based committee is used for evaluating these approaches. The results show that if the rejected classes are handled with a 50% error rate, the performance is improved. A scheme in which there is an adjustable threshold for distance-based rejection is an effective method for implementing rejection in this setting.

1 Introduction

In classification it is a common approach to use a set of reference samples to match the input sample against. If there is significant variation, having a large enough number of reference samples may quickly become impractical or even impossible. In such cases classifier adaptation is an effective method for improving performance.

Another approach to improve recognition performance is to combine different classifiers in a committee. This is feasible because in the outputs of several classifiers the errors are not necessarily overlapping. Although the most common way of adaptation is to adapt a single recognizer, it is also possible to construct a committee that as a whole is adaptive.

[1]Acknowledgement: This research was partly financed by the project New Information Processing Principles, Finnish Centre of Excellence Programme 2000-2005, Academy of Finland.

[2]Acknowledgement: This research was partly financed by a grant from the Nokia Foundation.

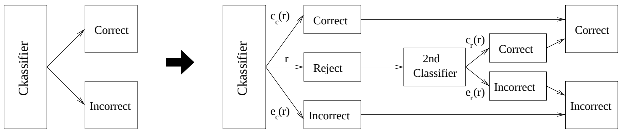

Figure 1. A schematic diagram of rejection to a secondary classifier.

There are many applications where recognition errors are to be avoided at all costs. In such situations the ability for a classifier, or a committee of classifiers, to reject an input of which it is uncertain becomes a very desirable property. Rejection can also be used to direct problematic samples to a separate classifier, a reject handler.

As we have previously experimented with adaptive committee structures, the purpose of this paper is to explore possible rejection implementations. Some basic reasoning for choosing an appropriate level of rejection is considered and the actual methods used are explained and evaluated.

2 Rejection

In general, the objective of rejection is to detect problematic patterns by some means, and refrain from classifying them entirely or redirect them to a reject handler classifier. An example approach is comparing the k nearest neighbors from several nearest neighbor classifiers and performing rejection if they differ [8]. Rejection can also be used as the primary method of classification through iteratively rejecting classes until only one class remains as the result [2].

When the objective of a system is to produce as many correct recognitions as possible, the main function of rejection is to redirect samples with high uncertainty to a separate, perhaps specialized, classification stage. The addition of such a stage is illustrated in Figure 1. Let r be the somehow tunable rejection rate of the initial classifier, or a committee as is here the case, $e_c(r)$ the first stage error rate at the rejection rate r and $e_r(r)$ the recognition error rate for the specialized reject handler. The correct classification rates are $c_c(r)$ and $c_r(r)$ analogously. We then have $c_c(r)+e_c(r)+r=1$ and $c_r(r)+e_r(r)=1$. Thus the total error rate becomes $e_{\text{tot}}=e_c(r)+re_r(r)$.

The purpose is to reject samples to be recognized incorrectly, but in practice usually also some samples that would have been correct are rejected. If $e_c(r)$ and $e_r(r)$ were continuous differentiable, the optimal level of rejection would be found at

$$\begin{aligned}\frac{d}{dr}e_{\text{tot}} &= \frac{d}{dr}[e_c(r)+re_r(r)] \\ &= \frac{d}{dr}e_c(r)+e_r(r)+r\frac{d}{dr}e_r(r)=0\end{aligned} \quad (1)$$

In practice with finite data, differentiation of the total error is usually impossible and the optimal rate of rejection can be found by finding the value $r* = r$ that minimizes the final error rate,

$$r^* = \arg\min_r [e_c(r)+re_r(r)] \quad (2)$$

As the correct percentage loses some intuitive value with its decrease as rejection increases, a measure of reliability, defined as in [11], is used for the primary classifier.

$$rel(r) = \frac{c_c(r)}{1-r} = \frac{c_c(r)}{c_c(r)+e_c(r)} \quad (3)$$

3 Adaptive committee

The basic operation of a committee classifier is to take the results of the member classifiers and attempt to combine them in a way that improves performance. The member classifiers have a significant impact on the final performance of the committee. It can generally be said that the less correlated the errors of the member classifiers are, the more effective the committee can be in improving the recognition rate.

Numerous committee structures have been studied over the years. They include majority voting [7], Bayesian approaches and k nearest neighbor combination methods [8], boosting [3], class ranking methods [10] and multistage combination [9], to name a few.

An adaptive committee can be thought of as consisting of two parts. First, every committee must have a base decision rule, which is used when no adaptation has been performed. Then, some rule or rules for the adaptation must be included. The type of the rules can vary from very simple weighting schemes to the creation of complex lists of rules to determine the committee's behavior.

One adaptive committee used in our work in on-line handwritten character recognition is based on the Dynamically Expanding Context (DEC) algorithm [5]. The algorithm was originally developed to create transformation rules that would correct typical coarticulation effects in phonemic speech recognition. The notation for a DEC rule stands as $l(A)r \rightarrow B$, where A is a segment of the source string S, B is the corresponding segment in the transformed string T, and $l(\cdot)r$ is the context in string S where A occurs. So in other words A is replaced by B under the condition $l(\cdot)r$. The main idea behind the approach is to determine just a sufficient amount of context for each individual segment A so that all conflicts in the set of training samples will be resolved. The method always first tries to find a production of the lowest contextual level sufficient to separate contradictory cases.

The DEC principle has been slightly modified to suit the setting of isolated handwritten character recognition [6]. In the DEC committee, the classifiers are first initialized and then tested separately and ranked in the order of decreasing performance. The primary outputs and the second-ranking results of every member classifier are used as a one-sided context for the creation of the DEC rules. Each time a character is input to the system, the existing rules are first searched through. If no applicable rule is found, the default decision is applied. The classification result is compared to the correct class. If the recognition was incorrect, a new rule is created. Every new rule that is created employs more contextual knowledge, if possible, than the rule causing the conflict. Eventually the entire context available will be used and more precise rules can no longer be written. For this situation a method for tracking the correctness of the rules can be used and the highest level rule most likely to be correct can be applied.

4 Rejection methods used

With our committee classifier structure, the rejection is implemented at the committee level. The rejection is independent of the committee result, and if rejection is performed, the committee does not process that character. The information available for performing recognition or rejection for an input sample x are the first and second ranking class labels from each of a total of N member classifiers and the distances to the nearest prototype of both result classes, $d_1^i(x)$ and $d_2^i(x)$ respectively for every member classifier i. The two suggested classes are always different.

The rejection decision can thus be based on either the class labels the member classifiers suggest or some measure obtainable from comparing the distances to the nearest prototypes of the two classes, or both. Also, for some rejection methods, external knowledge is introduced in the form of *a priori* difficult classes to recognize, or classes having previously caused a notable number of errors.

Voting Rejection (VR) is based on examining the variation within the results from the member classifiers. A parameter T_{vote} is given to determine how many different results may appear in the member outputs before rejection is performed. Thus if the number of different results is T_{vote} or less, no rejection is performed, and with the decrease in T_{vote}, rejection becomes more likely as only that number of differing results are allowed.

Distance Rejection (DR) is performed by comparing the first and second result distances obtained from all the member classifiers. If the averaged ratio

$$r_d(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{d_1^i(x)}{d_1^i(x) + d_2^i(x)} \tag{4}$$

is greater than a given threshold T, rejection is performed.

Learning Distance Rejection (LDR) works like the basic DR approach, but the parameter T is altered based on the received results. For this purpose a step value T_{step}, the amount how much T can be changed at a time, is given. If rejection was performed even though the result would have been correct, the value of T is increased as in $T(t+1) = T(t) + T_{step}$, as unnecessary rejections can be expected to become less frequent with a higher threshold value. On the other hand, if an incorrect result was not rejected, T is decreased by $T(t+1) = T(t) - T_{step}$, as in such a situation it should be expected that easier rejection could have helped in preventing the error. Thus rejected but correct answers would result in a less strict threshold (less probable rejection) for the following samples, and not rejected incorrect ones would tighten the threshold (more probable rejection).

Knowledge-based rejection (KR) refers to rejection based on a known set of easily confusable characters, given to the classifier as additional information. In our implementation these confusion sets are defined as the groups $\{o, O, 0\}$, $\{c, C\}$, $\{s, S\}$,$\{x, X\}$, $\{z, Z\}$ and $\{v, V\}$. Three alternative approaches are applied, in the order of increasing total rejection:

1. Only the first results of the member classifiers are considered. If two different members of one confusion group are found, rejection is performed.

2. Both the first and second results from the member classifiers are examined. If two different members of one confusion group are found, rejection is performed.

3. If the first result obtained from any one member classifier belongs to the confusion set, the result is rejected unless all the member classifiers agree on the result.

Learning Knowledge-based Rejection (LKR) keeps track of classifications that have occurred to each character class and the errors that have been made. If the ratio of errors and classifications is greater than a given threshold T, rejection is performed. The value of T is adjusted during classification as with LDR. In this case the adjustment is done only towards more rejection: when rejection did not occur and the result was incorrect, the threshold is lowered.

It should be noted that for the LDR and LKR methods the rejection rate is not directly steerable, because the adjustment process attempts to find a suitable threshold.

Class Rejection (CR) is the most simple of the applied rejection methods and constitutes of simply disregarding a predetermined set of classes known to be difficult. These difficult classes were determined in previous experiments as those where most errors resulted into. The rejected classes were obtained by using progressively more and more characters from the string "$nmurhs0ol9adkbcefgyv1i$".

Table 1. Summary of the databases.

Database	Subjects	Characters	(a-z,0-9)
DB1	22	~ 10 400	8461
DB2	8	~ 8 100	4643

5 Experiments

The committee was run in batch mode simulating on-line operation by using all samples once in their original order. The DEC rule base was reset for each writer for writer-dependent adaptation. The rejection methods were tested by varying the values of their parameters.

5.1 Description of the data sets

The data used in the experiments were isolated on-line characters collected on a Silicon Graphics workstation using a Wacom Artpad II tablet and stored in UNIPEN format [4]. The preprocessing is covered in detail in [12]. The databases are summarized in Table 1. Database 1 consists of characters which were written without any visual feedback. The pressure level thresholding the measured data into pen up and pen down movements was individually set for each writer. The *a priori* probabilities of the classes were based on the Finnish language. Database 2 was collected with a program that showed the pen trace on the screen and recognized the characters on-line. The minimum writing pressure for detecting pen down movements was the same for all writers. The distribution of the character classes was approximately even.

The databases consisted of different writers. Database 1 was used for constructing the initial user-independent prototype set which consisted of 7 prototypes for each class and database 2 was used as a test set. Only lower case letters and digits, a total of approximately 580 characters per writer from database 2, were used as test data in the experiments. As only lower-case letters were used, the effect of KR is limited to confusions between the classes *o* and 0.

Table 2. Recognition error rates of the four committee member classifiers.

Classifier	Distance measure	Bounding box	Mass center	Errors
1	PL		•	14.9%
2	NPP		•	15.1%
3	NPP	•		18.2%
4	PL	•		19.6%

5.2 Member classifiers

For these experiments, four individual classifiers were used as the committee members. All the classifiers were based on stroke-by-stroke distances between the given character and the prototypes. Dynamic Time Warping (DTW) was used to compute one of two distances, normalized point-to-point (NPP) or point-to-line (PL) [12]. The NPP distance simply uses the squared Euclidean distance between two data points as the cost function and the total sum is divided by the number of matchings performed. In the PL distance the points of a stroke are matched to lines interpolated between the successive points of the opposite stroke. All character samples are scaled so that the length of the longer side of their bounding box is constant and the aspect ratio is kept unchanged. Also the centers of the character, defined by either the 'Mass center' as the input sample's mass center or by 'Bounding box' as the center of the sample's bounding box, are moved to the origin. The member classifiers were not adaptive. The configurations and error rates of the member classifiers are shown in Table 2.

In general, a committee can be expected to perform the better, the less the errors made by its members are correlated. Unfortunately uncorrelatedness is not the case here [1]. As the DTW-based classifier is the only one of our classifiers currently capable of acceptable recognition performance, the member classifiers are rather similar.

5.3 DEC configuration

Several options were explored in the search for the best achievable recognition result using the DEC committee. In-depth experiments with the DEC committee classifier and this data set have been conducted [1]. It was found that the DEC committee should use both the first- and the second-ranking results in the manner that all first-ranked results are used prior to any second-ranked results from any classifier to obtain the best performance. The default rule was to use the result of the best individual classifier.

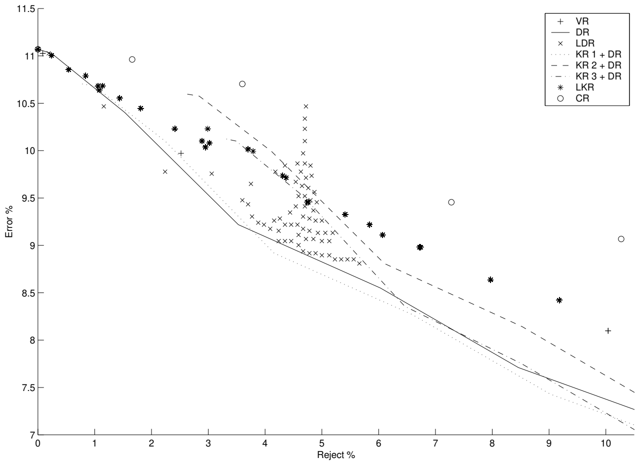

Figure 2. Rejection experiment results.

6 Results

The rejection strategies described in Section 4 were tested by varying the available parameters. The results obtained are shown in Figure 2. The '$+$'-marks correspond to voting rejection, the solid line to distance rejection and the '$\times$'-marks to learning distance rejection. The visible peak is a result of the adjustment of the rejection threshold directing it to a suitable value, which it may oscillate around. Depending on the initial parameter values different characters are rejected in the process, as can be seen from the varying error rates. The dotted line is the first knowledge-based method, the dashed the second, and the dash-dot the third, all in combination with the distance based rejection scheme. The '$*$'-marks represent learning knowledge rejection and the circles class rejection. While all the other methods were tested independently, the knowledge rejection options were applied in conjunction with distance rejection. The result of the knowledge reject option alone can be seen at the beginning of the respective lines.

As the intention is to give the rejected samples to a classifier specifically designed to handle the rejections, it is impossible to determine which rejection rate vs. error rate combination is generally the most effective, since this depends on the reject classifier error rate. As the rejection handling stage has not yet been implemented, exact numbers are not available. But one may assume that an accuracy of 50% may be obtainable for the rejects. Based on this assumption the optimal values with regard to (2) have been gathered into Table 3. The column 'total error' includes the errors from the rejected characters assuming that the reject handler error rate is constant, $e_r = 0.5$.

It can be seen that rejection can decrease the overall error rate when assuming the reject classifier to function at

Table 3. Best results for each rejection method assuming rejection handler error rate $e_r = 0.5$.

Rej	e_c%	rej%	rel%	parameters	e_{tot}%
None	11.07	0.00	88.96	-	11.07
VR	11.03	0.09	88.96	$T_{vote} = 2$	11.07
DR	9.22	3.53	90.44	$T = 0.47$	10.98
LDR	9.76	2.26	90.02	$T = 0.48$, $T_{step} = 0.01$	10.89
KR 1	8.92	4.16	90.70	$T = 0.47$	11.00
KR 2	8.81	6.10	90.62	$T = 0.47$	11.86
KR 3	8.36	6.46	91.07	$T = 0.47$	11.58
LKR	10.86	0.54	89.09	$T = 1$, $T_{step} = 0.66$	11.13
CR	10.96	1.66	88.85	reject 'n','m'	11.79

the 50% recognition rate. When examining the total error rate, taking into account the reject processing classifier's errors, it can be seen that the best individual result is obtained through the learning distance-based rejection scheme, followed by the distance-based rejection alone and the combination of the first knowledge-based option and distance-based rejection.

On the other hand, when examining the reliability values, which actually correspond to the first stage's correct recognition percentage over the characters not rejected, the best value would seem to be obtained from the third knowledge-based rejection mode, again combined with distance rejection. It is also the one with the highest reject rate. The change in order most likely stems from the definition of reliability (3): more rejection and fewer errors naturally increase this value. When the situation is focused on as many correct recognitions as possible and the cost of errors vs. rejections is not high, it would seem that the reliability really is not as interesting a measure, but the total error rate is what should be looked at.

7 Conclusions

In the case of a system where minimizing errors is the objective, the tradeoff is how much rejection is acceptable, as increasing rejection effectively reduces the error rate. But in a system where the primary objective is to produce as many correct responses as possible, it should be desirable to use a separate mechanism for classifying the rejects.

The rejection mechanisms, with the exception of voting rejection, could also be used on the member classifier level. But when using a committee such as our DEC-based one, having one classifier entirely restrain from giving a result would lead to rules with empty "context places", which would undermine the basic concept of the DEC committee. Thus in this case rejection is better performed on the committee level.

It has been shown that for the committee reaching 11% error rates the use of a reject classifier with a 50% error rate can improve the results if an effective rejection scheme is used. Of the tested methods, a scheme in which there is an adjustable threshold for distance-based rejection was the most effective one.

References

[1] M. Aksela, J. Laaksonen, E. Oja, and J. Kangas. Application of adaptive committee classifiers in on-line character recognition. In *Proceedings of International Conference on Advances in Pattern Recognition*, pages 270–279, 2001.

[2] S. Baker and S. K. Nayar. Pattern rejection. In *Proceedings of the conference on Computer Vision and Pattern Recognition*, pages 544–549, 1996.

[3] H. Drucker, R. Schapire, and P. Simard. Boosting performance in neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):705–719, 1993.

[4] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and recognizer benchmark. In *Proceedings of International Conference on Pattern Recognition*, pages 29–33, 1994.

[5] T. Kohonen. Dynamically expanding context. *Journal of Intelligent Systems*, 1(1):79–95, 1987.

[6] J. Laaksonen, M. Aksela, E. Oja, and J. Kangas. Dynamically Expanding Context as committee adaptation method in on-line recognition of handwritten latin characters. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 796–799, 1999.

[7] L. Lam and C. Y. Suen. A theoretical analysis of the application of majority voting to pattern recognition. In *Proceedings of 12th International Conference on Pattern Recognition*, volume II, pages 418–420, Jerusalem, Oct. 1994. IAPR.

[8] A. M. Michael Sabourin and D. Thomas. Classifier combination for hand-printed digit recognition. In *Proceedings of the Internationa Conference on Document Analysis and Recognition*, pages 163–166, 1993.

[9] J. Paik, S. bae Cho, K. Lee, and Y. Lee. Multiple recognizers system using two-stage combination. In *Proceedings of International Conference on Pattern Recognition*, pages 581–585. IEEE, 1996.

[10] J. J. H. Tin Kam Ho and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

[11] Y. B. Vladimir Radevski. Reliability control in committee classifier environment. In *Proceedings of International Joint Conference on Neural Networks*, volume 2, pages 561–565, 2000.

[12] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas. Experiments with adaptation strategies for a prototype-based recognition system of isolated handwritten characters. *International Journal of Document Analysis and Recognition*, 3(2):150–159, 2001.