Class-Confidence Critic Combining

Matti Aksela, Ramūnas Girdziušas, Jorma Laaksonen, Erkki Oja
Helsinki University of Technology, Laboratory of Computer and Information Science
P.O.Box 5400, Fin-02015 HUT, Finland
{matti.aksela,ramunas.girdziusas,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

This paper discusses a combination of two techniques for improving the recognition accuracy of on-line handwritten character recognition: committee classification and adaptation to the user. A novel adaptive committee structure, namely the Class-Confidence Critic Combination (CCCC) scheme, is presented and evaluated. It is shown to be able to improve significantly on its member classifiers. Also the effect of having either more or less diverse sets of member classifiers is considered.

1 Introduction

In on-line handwriting recognition the classifier or classifiers must be capable of processing natural handwriting at high accuracies for the application to be comfortable for the user. Including the vast amount of intrinsic variation in handwriting in the initial character models is often impossible, or at least very impractical. Thus adaptation is a feasible or even an unavoidable way of improving performance on any user-dependent handwriting recognition system.

Combining several different classifiers in a committee form is another way to reach for the best attainable recognition performance. Combining the results of several classifiers can improve performance because in the outputs of the individual classifiers the errors are not necessarily overlapping. Committee methods generally require more than one member classifier to recognize the input. In on-line handwritten character recognition, this is not computationally too complex for even the smallest platforms due to the continuous increase in available computational power. The basic operation of a committee classifier is to take the results of a set of member classifiers and attempt to combine them in a way that improves performance. The two most important features of the member classifiers that affect the committee's performance are the individual error rates of the member classifiers and the correlatedness of the errors. The more different the mistakes made by the classifiers, the more beneficial combining them can be. Numerous committee structures have recently gained attention, for example boosting [1] and critic-driven combining [4]. The Behavior-Knowledge Space (BKS) method [3] is based on a K-dimensional discrete knowledge space that is used to determine the class labels based on previously stored decision information.

Though the adaptation is usually performed by adapting a single classifier to the training data, it is also possible to construct a committee that is adaptive as a whole. The members of an adaptive committee can be adaptive or non-adaptive themselves. One adaptive classifier combination strategy is to combine the member classifiers linearly using weighting coefficients dynamically acquired from a combination coefficient predictor [8]. We present here an adaptive committee classification scheme based on critics evaluating the trustworthiness of the members. The technique is named Class-Confidence Critic Combining (CCCC).

2 Class-Confidence Critic Combining

Generally, a critic-based approach is one in which a separate expert makes a decision on whether the classifier it is examining is correct or not. Critic-driven approaches to classifier combining have been investigated previously, e.g. in a situation where the critic makes its decision based

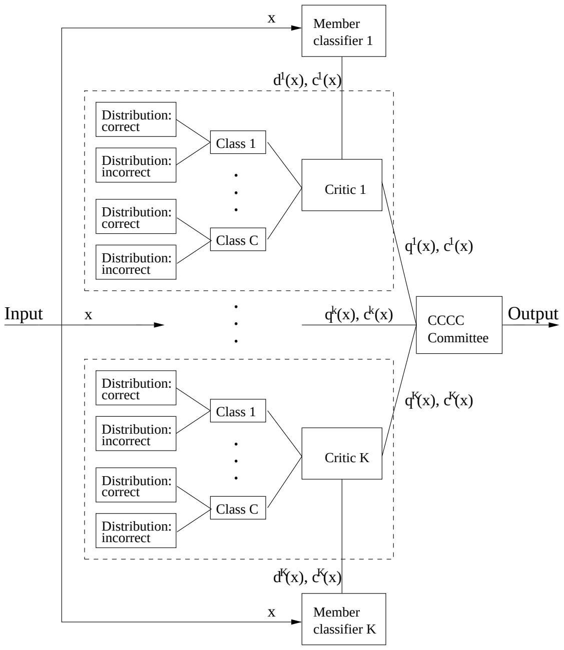

Figure 1. Structure of the CCCC committee

on the same input data as the classifier [4]. In our Class-Confidence Critic Combining (CCCC) approach the main idea is to try to produce as good as possible an estimate on the classifier's correctness based on its prior behavior for the same character class.

In CCCC the correctness evaluation results in a confidence value which is based on the earlier performance of the classifier for the same character class. There are two distance distributions for each class stored in each critic. One corresponds to correct classification results and the other one to incorrect results. This is illustrated in Figure 1. Each time a new character x is processed, it is classified by all member classifiers and the confidences of the classification are calculated in each critic. Two opposing confidences are estimated in each critic, one for the correctness and the other for the incorrectness of the classification. The confidences in the label $c^k(x)$ from classifier k are based on the distribution of a distance-indicating value $d^k(x)$. For calculating the $d^k(x)$ value we need to know the distances from the input x to the nearest prototypes, $d_1^k(x)$ and $d_2^k(x)$, calculated for the nearest and second-nearest class, respectively. In the case that the classifier is not based on distances, we may use another analogous measure that decreases as similarity increases. The value $d^k(x)$ can be obtained by taking the ratio between the distance to the first result $d_1^k(x)$ and the sum of the first and second result distances,

$$d^k(x) = \frac{d_1^k(x)}{d_1^k(x) + d_2^k(x)} \in [0, \frac{1}{2}]. \tag{1}$$

Or, we may use directly the distance to the nearest prototype calculated by the classifier,

$$d^k(x) = d_1^k(x) \in [0, \infty). \tag{2}$$

The committee then uses one of the decision mechanisms specified in Section 2.3 to produce the final output from the input label information and critic confidence values $q^k(x)$ calculated from the confidences for $d^k(x)$.

The adaptation of the critics, in practice the modification of the distributions, is performed assuming that it is known whether the recognition result was correct or incorrect. The $d^k(x)$ values received from the member classifiers are incorporated into the corresponding critic's appropriate distribution, depending on the suggested class and the correctness of the result. In practice this is done by appending the new $d^k(x)$ value to the list of values for that distribution and recalculating the parameters needed for presenting the distribution.

2.1 Distribution types

In order to obtain the confidences for the decisions based on previous results, the received $d^k(x)$ values must be somehow modeled. The approach of gathering previous values into distributions from which the value for the confidence can be obtained has been chosen for this task. The notation used is that each distribution i, where the shorthand distribution index i runs over both correct and incorrect distributions for each class c in each member classifier k, contains N^i previously collected values $z_j^i, j = 1, \ldots, N^i$. The notation for the confidence obtained from the distribution i stands as $p^i(d^k(x))$. For shortening the notation further, we shall use $d^k(x) = z$.

Gaussian normal distribution: The Gaussian normal distribution is used by calculating the mean and variance from the already obtained samples and then calculating the values of a Gaussian normal distribution,

$$p^i_{\text{gaussian}}(z) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(z-\mu_i)^2}{2\sigma_i^2}}, \tag{3}$$

where μ_i is the mean and σ_i^2 the variance for the distribution i. Initial values are used for the mean when no samples exist and for the variance when less than two samples have been received for the particular distribution.

Non-parametric distribution: The non-parametric model is based on calculating the number of points in the distribution that are further from the mean of the distribution $\mu_{i,k}$ than the value for the input z, i.e. $n_f(z,i) = \sum_{j=1}^{N^i} v(z,i,j)$, where $v(z,i,j) = 1$ if $|z - \mu_i| < |z_j^i - \mu_i|$ and zero otherwise. The confidence is then based on the ratio between $n_f(z,i)$ and the total number of points in the distribution N^i so that

$$p^i_{\text{nonparam}}(z) = \frac{n_f(z,i)}{N^i}. \quad (4)$$

Nearest neighbor approach: The nearest neighbor (NN) approach is not really a distribution, but the nearest neighbor rule is used in the sense of calculating the distance $d^i_{\text{min}}(z) = \min_{j=1}^{N^i} |z - z^i_j|$ from the input value z to the nearest value already in the distribution i.This is then used with the largest attainable distance d^i_{max} to create a measure of confidence,

$$p^i_{\text{NN}}(z) = 1 - \frac{d^i_{\text{min}}(z)}{d^i_{\text{max}}}. \quad (5)$$

If equation (1) is used, $d^i_{\text{max}} = 0.5$. When equation (2) is used, d^i_{max} is taken to be the largest value observed.

Triangular kernel distribution estimate: This distribution estimate uses a triangular kernel function, defined by the peak bandwidth b, which is given as a parameter. The estimate can be calculated by applying a kernel over all data points z^i_j in the distribution i and normalizing by the number of points N^i. Because b is independent of the distribution and critic, there is no need to take it into account in the normalization;

$$p^i_{\text{trikernel}}(z) = \frac{1}{N^i} \sum_{j=1}^{N^i} \max\{0, \frac{1}{b}(b - |z - z^i_j|)\}. \quad (6)$$

Gaussian kernel distribution estimate: The distribution is estimated through the use of a Gaussian function as the kernel. The kernel bandwidth b is used as the variance for the Gaussian. The evaluation of the distributions' values at specific points is performed as for the triangular kernel,

$$p^i_{\text{gausskernel}}(z) = \frac{1}{N^i} \sum_{j=1}^{N^i} e^{-\frac{(z-z^i_j)^2}{2b}}. \quad (7)$$

2.2 Combining confidence values

The overall confidence $q^k(x)$ given by critic k to the classification result $c^k(x)$ of classifier k is obtained from the correct and incorrect classification result distribution confidences $p^{\text{correct}}(d^k(x))$ and $p^{\text{incorrect}}(d^k(x))$, respectively, either by subtracting them from one another, where

$$q^k(x) = p^{\text{correct}}(d^k(x)) - p^{\text{incorrect}}(d^k(x)), \quad (8)$$

or by using just the confidence from the correct distribution as the overall confidence,

$$q^k(x) = p^{\text{correct}}(d^k(x)). \quad (9)$$

It should be noted that (8) may produce also negative confidences indicating that the result from that member classifier is expected to be incorrect.

2.3 Decision mechanisms

As the committee now has label information from the member classifiers and the corresponding confidence values from the critics to work with, a scheme is needed for combining them into a final result. The decision schemes take the labels $c^k(x)$ for the input samples x from classifiers k and the corresponding critics' confidences $q^k(x)$ to form the decision.

Maximum confidence selection: The decision is made by selecting the result whose critic has the highest confidence,

$$c(x) = c^j(x), \quad j = \arg\max_{k=1}^{K} q^k(x). \quad (10)$$

Confidence-weighted majority voting: Weighted majority voting is performed with the confidences as the weights. With the use of the confidences, the majority voting scheme is modified to assigning

$$c(x) = \arg\max_{c=1}^{C} \sum_{k=1}^{K} q^k(x)\Delta_{ck}, \quad (11)$$

where C is the total number of classes and K the number of recognizers. $\Delta_{ck} = 1$ when the result from the classifier k is the class c and zero otherwise.

Modified Current-Best-Learning decision: The Current-Best-Learning (CBL) algorithm [5] is originally a framework for learning general logical descriptions. This is accomplished through maintaining a single hypothesis and adjusting it as new examples arrive. Operations known as generalization and specialization are used to adjust the current hypothesis so that the resulting hypothesis is consistent with all the examples.

The algorithm used here has grown quite far from that initial idea, but as the resemblance is still evident, it is here called the Modified Current-Best-Learning (MCBL) approach. If one interprets CBL as a method of combining classifiers, the system can be viewed as a two-dimensional grid, with each column representing a member classifier and each row corresponding to a particular class. The values stored in the grid are estimates for the confidence of a member classifier's decision if it classifies an input in that particular class. Specialization and generalization then give rise to changing the confidence values.

When forming the class-wise MCBL confidence values, one uses the confidences obtained from the critics, $q^k(x)$. By combining them into class-wise confidence values $f^k(c^k(x))$, where k is the index of the classifier and $c^k(x)$ the class suggested by that classifier for the input x, a table consisting of each classifier's classification result and its confidence can be formed. To modify the hypothesis, the values $f^k(c^k(x))$ are adjusted when the committee as a whole is incorrect. When any individual classifier k of the committee members is correct, the $q^k(x)$ value is added to the confidence of the class for that classifier. On the other hand, when a classifier produces an incorrect result, its confidence for that class is multiplied with the value $q^k(x)$. The modifications can thus be formulated as

$$\forall k \in \{1,\ldots,K\}:$$
$$f^k(c^k(x)) := \begin{cases} f^k(c^k(x)) + q^k(x), & \text{if } c^k(x) \text{ correct} \\ f^k(c^k(x)) \cdot q^k(x)\ , & \text{otherwise.} \end{cases} \tag{12}$$

When the committee produces a correct result, the current hypothesis has been effective and no changes are made. Due to the on-line nature of the adaptation, no backtracking is performed and each sample is processed only once. The confidence values can be initialized as the inverse of the ordering of the classifiers according to their decreasing recognition performance, ie. $f^k(\omega_j) = \frac{1}{k}$ for all classifiers k and class labels ω_j.

Prior to the final decision, the obtained confidences were still modified by joining the critic's current confidence value into the obtained MCBL confidence value by using the transformation of equation (12). As the correctness is not known at this point, the selection is made based on whether the critic believes the member to be correct ($q^k(x) > 0$) or not. This last step should be beneficial when the critics directly produce sufficiently accurate confidence estimates.

This modification scheme was used as it was the one found to produce the best results from a number of schemes experimented with. For the final decision from the MCBL confidence values, both the original scheme selecting the result based on the maximum value as in equation (10) and a scheme using the weighted majority voting approach of equation (11) were experimented with.

3 Reference committee classifiers

To evaluate the results of the CCCC committee, some runs with reference committee classifiers have also been carried out. They include the standard plurality voting, adjusting plurality voting, and adjusting best approaches.

Plurality voting committee: The first reference committee simply uses the plurality voting rule to decide the final output. In the case of a tie the approach of iteratively dropping the classifier with the lowest correctness ranking and revoting was used.

Adjusting plurality voting committee: A simple approach to adaptive committee decisions is to use a weighted variation of the original plurality voting rule. Adaptation was implemented by introducing weights based on an evaluation of correctness for each voting classifier, where $w_k = \frac{1+N_c^k}{1+\sum_{j=1}^{K} N_c^j}$ is the weight for the output and N_c^k is the current count of correct recognitions for the classifier k, and K is the total number of classifiers. The addition of one in both the nominator and denominator is made to avoid both zero weights and divisions by zero. The final plurality voting decision is obtained as in equation (11), with the weights w_k replacing the confidences $q^k(x)$.

Adjusting best committee: In the adjusting best committee the main idea is to select the best classifier for each individual writer by evaluating each classifier's performance during operation and using the result from the classifier that has performed the best up to that time. The performance evaluation is conducted by simply keeping track of correct results obtained from each classifier. At any given time the committee's decision is thus the result from the classifier with the highest correct answer count at that point, $c(x) = c^j(x)$, where $j = \arg\max_{k=1}^{K} N_c^k$, with N_c^k being the current count of correct recognitions for classifier k and $c^k(x)$ the class suggested by that classifier. In the case of a draw, the result from the higher-ranked classifier is used.

4 Member classifiers

The adaptive committee experiments were performed using a subset of six classifiers from the total of eight different classifiers created. Four of the member classifiers were based on stroke-by-stroke distances between the given character and prototypes. Dynamic Time Warping (DTW) was used to compute one of two distances, point-to-point (PP) or point-to-line (PL) [7]. The PP distance uses the squared Euclidean distance between two data points as the cost function. 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 were scaled so that the length of the longer side of their bounding box was normalized and the aspect ratio kept unchanged. Also the centers of the character, defined either as the input sample's mass center (MC) or as the center of the sample's bounding box (BBC), were moved to the origin. These classifiers are the first four in Table 1.

Two Support Vector Machine (SVM) -based classifiers were created so that the on-line characters were first mapped into bitmaps. The bounding box was first identified for every character and scaled into a normalized box. The character bitmap image was constructed by thickening the lines and creating high resolution 400×400 binary images. After applying a down-sampling procedure, the resulting gray-level character bitmaps of size 20×20 were created. The bitmaps were then stacked column-wise into 400-dimensional vectors and their projections onto 64 principal components were used as features. The SVM classifier was applied to classify the obtained features by constructing binary classifiers, each one separating one class from the rest. The decomposition principle implemented in [6]

Table 1. Member classifier rates

Classifier	Distance measure	Errors
1	DTW-PP-MC	10.9%
2	DTW-PL-MC	11.5%
3	DTW-PP-BBC	12.2%
4	DTW-PL-BBC	13.6%
5	SVM-Gaussian	21.8%
6	SVM-Polynomial	22.6%
7	DTW-NPP-MC	12.3%
8	DTW-NPP-BBC	13.4%

was used to train the SVMs in the experiments [2]. The SVM classifiers can be found on lines 5 and 6 in Table 1.

We did experiments to evaluate the benefit of having diverse classifiers, i.e., using the two SVM-based classifiers in addition to the DTW-based classifiers. In the experiments the two SVM-based classifiers were replaced with two additional DTW-classifiers, so the committee consisted of six different DTW-classifiers. These last two also use the same preprocessing as explained above, but a distance measure called the normalized point-to-point (NPP) distance [7]. This measure is very similar to the point-to-point distance but with the addition of normalizing the calculated cost by the number of matchings performed. These classifiers are on lines 7 and 8 in Table 1.

5 Experiments

The data used in the experiments were isolated on-line characters collected on a Silicon Graphics workstation using a Wacom Artpad II tablet. The data was stored in UNIPEN format. The preprocessing is covered in detail in [7]. The databases are summarized in Table 2. The databases consisted of characters by entirely different writers. Only lower case letters and digits were used in the experiments. Database 1 consists of characters written without any visual feedback. The *a priori* probabilities of the classes were somewhat similar to that of the Finnish language. Databases 2 and 3 were collected with a program that showed the pen trace on the screen and recognized the characters on-line. The distribution of the character classes was approximately even.

Database 1 was used for forming the initial user-independent member classifiers. The prototype set for the DTW-based classifiers consisted of 7 prototypes per class, and the SVM extracted a total of approximately 6000 support vectors. Database 2 was used for evaluating some general numeric parameters for the CCCC committee and determining the performance rankings of the classifiers. Database 3 was used as a test set.

Table 2. Summary of the databases used

Database	Writers	Characters	(a-z,0-9)
DB1	22	$\sim$ 10 400	8461
DB2	8	$\sim$ 8 100	4643
DB3	8	$\sim$ 8 100	4626

Table 3. Effects of CCCC components

Distribution/Decision	Average error%	Best error%
Triangular kernel distribution	11.2	8.4
Gaussian kernel distribution	14.7	9.3
Non-parametric distribution	18.1	8.3
Nearest neighbor "distribution"	18.4	8.0
Gaussian distribution	19.1	8.4
MCBL decision	15.2	8.5
MCBL-vote decision	15.3	8.0
Maximum confidence decision	16.6	9.7
Weighted voting decision	17.7	9.3

6 Results

The results for the CCCC configurations have been obtained by using the first six member classifiers from Table 1. The committees were implemented and run in batch mode: on-line operation was simulated by taking data in its original order and disallowing reiteration. The error rates have been calculated over all characters for all writers. All adaptive committee classifiers were reset in between writers for writer-dependent operation.

The most effective combination for the CCCC scheme seems to be to use the nearest-neighbor confidence model along with the MCBL-voting-decision mechanism, as can be seen from the best results column in Table 3. This combination provides an error rate of 8.0%.

Also several less fundamental options were experimented with. They included the possibility of using the second-ranking result if the first-ranked result from the classifier had low confidence, learning only on the committees errors, repeatedly inserting samples into the distributions to enhance learning effects, adjusting the confidences with run-time recognition rates and not accepting results with negative confidences. But due to space concerns the reporting has been omitted here, as their significance to the method was much lower.

The effects of the individual components were evaluated by averaging over all the runs with a particular option in use. The best error percentages correspond to the best run using the component. The averages presented in the tables have been calculated over all combinations of the options, resulting in notably low average rates due to some absurd combinations that result in very high error rates. Table 3 shows that in general the kernel-function-based distribution estimates do perform better, with the triangular kernel function performing on the average the best. The difference between using the non-parametric distribution and the nearest-neighbor approach is quite small. The use of one Gaussian seems to be insufficient. But looking at the lowest error rates, the picture is quite different with the nearest-neighbor approach performing the best, followed by the non-parametric, triangular kernel and simple Gaussian distributions, and the Gaussian kernel being clearly the worst.

Table 3 also shows that the MCBL decision mechanism applied to the confidences obtained from the critics produces clearly the best results. The difference between using the single maximum or voting variation is very small, but the MCBL variation of selecting the single largest confidence is on the average slightly better than its counterpart based on weighted voting and the MCBL-vote approach producing the best individual result. The weighted voting approach seems to be inferior to just choosing the result with the best confidence. But the best single result from the two decision mechanisms not based on MCBL is received through the voting-based approach.

The results of the committees are compared in the middle column of Table 4. Also the result from the best individual member classifier and the average of members are shown. The CCCC committee outperforms all the other methods used. The voting approaches perform better than the adjusting best approach, the only one unable to outperform all its members. The adjusting plurality voting is able to perform slightly better than the basic voting scheme.

An additional experiment was run to evaluate the benefits of having the SVM-based classifiers, which produce worse results but also different errors than those based on DTW, used in the committees. To evaluate their benefit, experiments were run also using only DTW-based committee classifiers. For these comparison experiments the two SVM-based member classifiers 5 and 6 in Table 1 were replaced with the DTW-based member classifiers 7 and 8. The results are in the last column of Table 4. As can be seen, especially the more advanced CCCC combination method benefits notably from having the more diverse set of member classifiers, even though the average error rate over the member classifiers is clearly higher. The plurality voting approaches also benefit from the diversity, as can be seen in the slight increases in error percentages when moving to all-DTW member classifiers. On the other hand the adjusting best scheme is more dependent on having well-performing member classifiers than on anything else. As such it benefits from having more similar member classifiers with lower individual error rates.

Table 4. Comparison of adaptive committees

Combination method	Error % DTW& SVM	Error % all DTW
CCCC	8.0	9.3
Adjusting Plurality Voting	10.1	10.3
Plurality Voting	10.2	10.4
Adjusting Best	11.4	11.3
Best member classifier	10.9	10.9
Member classifier average	15.4	12.3

7 Conclusions

The experiments regarding adaptive CCCC committee have shown notable improvements in performance over any of the individual members. The CCCC approach using a nearest-neighbor distribution and the MCBL-vote decision rule was the most effective combination of the ones tested. It is also clear from the results that combining more diverse member classifiers is beneficial, even if some of the members by themselves perform worse. The most important factor is that the member classifiers should not make the same mistakes, as the situations where the member classifiers all suggest a single incorrect result is the most difficult one to correct.

References

[1] 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.

[2] R. Girdziušas. Discriminative on-line recognition of isolated handwritten character. Master's thesis, Helsinki University of Technology, 2001.

[3] Y. Huang and C. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):90–94, 1995.

[4] D. Miller and L. Yan. Critic-driven ensemble classification. *IEEE Transactions on Signal Processing*, 47(10):2833–2844, 1999.

[5] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[6] A. Schwaighofer. SVM toolbox for Matlab (version v0.4). http://svm.first.gmd.de/, May 2001.

[7] 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.

[8] B. Xiao and C. W. nd R.W. Dai. Adaptive combination of classifiers and its application to handwritten chinese character recognition. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 327–330, 2000.