Combination of Multiple Classifiers Using Local Accuracy Estimates

Kevin Woods[2], W. Philip Kegelmeyer Jr.[3], and Kevin Bowyer[2]

2. Computer Science & Engineering, Univ. of South Florida, Tampa, FL 33620
woods@csee.usf.edu, kwb@csee.usf.edu

3. Sandia National Laboratories, MS 9214, PO Box 969, Livermore, CA, 94551
wpkege@california.sandia.gov

This work was sponsored by the Department of the Army research grants #DAMD17-94-J-4328 and #DAMD17-94-J-4015, and by the United States Department of Energy at Sandia National Laboratories, Livermore, through contract #DE-AC04-76DO00789.

Combination of Multiple Classifiers Using Local Accuracy Estimates

Kevin Woods, W. Philip Kegelmeyer Jr., and Kevin Bowyer

ABSTRACT

This paper presents a method for combining classifiers that uses estimates of each individual classifier's local accuracy in small regions of feature space surrounding an unknown test sample. An empirical evaluation using five real data sets confirms the validity of our approach compared to some other Combination of Multiple Classifiers algorithms. We also suggest a methodology for determining the best mix of individual classifiers.

1 Introduction

There are two basic approaches a Combination of Multiple Classifiers (CMC) algorithm may take: classifier fusion, and dynamic classifier selection. In classifier fusion algorithms, individual classifiers are applied in parallel, and their outputs are combined in some manner to achieve a "group consensus". Dynamic Classifier Selection attempts to predict which single classifier is most likely to be correct for a given sample. Only the output of the selected classifier is considered in the final decision.

Previous classifier fusion algorithms include the majority vote [1, 2], the Borda count [3], unanimous consensus [3, 2], thresholded voting [2], polling methods which utilize heuristic decision rules [4, 5], the "averaged Bayes classifier" [2], logistic regression to assign weights to the ranks produced by each classifier [3], Dempster-Shafer theory to derive weights for each classifier's vote [2, 6], and methods of multistage classification [7].

For dynamic classifier selection, a method of partitioning the input samples is required. For example, partitions can be defined by the set of individual classifier decisions [8], according to which classifiers agree with each other [3], or even by features of the input samples. Then, the "best" classifier for each partition is determined using training or validation data. For classification, an unknown sample is assigned to a partition, and the output of the best classifier for that partition is used to make the final decision.

The objective of this work is to present a general method of improving accuracy in CMC systems. We begin with descriptions of our proposed algorithm and three other CMC algorithms which were implemented for comparison. We then present experimental procedures and results for five different sets of data from various real applications.

2 Algorithms for Comparison

We have selected two previously published algorithms [8, 9] for direct comparison to our proposed algorithm. We also implemented a modified version of one of these algorithms.

2.1 The Proposed Approach: DCS-LA

We term our approach to CMC as Dynamic Classifier Selection by Local Accuracy, or DCS-LA. The basic idea is to estimate each classifier's accuracy in local regions of feature space surrounding an unknown test sample, and then use the decision of the most locally accurate classifier. In our implementation "local regions" are defined in terms of the K-nearest neighbors in the training data. We examine two methods for estimating local accuracy. One is simply the percentage of training samples in the region that are correctly classified. We shall refer to this as the overall local accuracy. Another possibility is to estimate local accuracy with respect to some output class. Consider a classifier that assigns a test sample to class C_i. We can determine the percentage of the local training samples assigned to class C_i by this classifier that have been correctly labeled. We shall refer to this as the local class accuracy.

2.2 The Behavior-Knowledge Space Approach

The Behavior-Knowledge Space (BKS) algorithm has recently been proposed in connection with an application for recognizing handwritten numerals. Behavior-Knowledge Space is an N-dimensional space where each dimension corresponds to the decision of one classifier. Each classifier can assign a sample to one of M possible classes. Each unit of a BKS represents a particular intersection of individual classifier decisions. Thus, the BKS represents all possible combinations of the individual classifier decisions. Each BKS unit accumulates the number of training samples from each class. For an unknown test sample, the decisions of the individual classifiers index a unit of BKS, and the unknown sample is assigned to the class with the most training samples in that BKS unit[1].

2.3 The Classifier Rank Approach

Sabourin et al. [9] present an algorithm which has some similarities to our DCS-LA approach. One variation of their algorithm selects the classifier that correctly classifies the most consecutive neighboring training samples (relative to the unknown test sample). The selected classifier is said to have the highest "rank". Although they do not associate their algorithm with the concept of local accuracy, their notion of classifier rank certainly has this flavor. We will refer to this algorithm as the Classifier Rank method.

2.4 A Modified Classifier Rank Approach

In terms of our work, the Classifier Rank algorithm presented in [9] uses what we would describe as an overall local accuracy estimate. An obvious alternative would be to use local class accuracy. Given a test sample assigned to class C_i by a classifier, local accuracy for the classifier is estimated as the number of consecutive nearest neighbors assigned class C_i which have been correctly labeled. We refer to this algorithm as Modified Classifier Rank.

3 Empirical Comparison on ELENA Data Sets

From the ELENA project[2], we selected four data sets representing real applications: iris_CR, phoneme_CR, satimage_CR, and texture_CR. The CR notation indicates that each database was preprocessed by a normalization routine in which each feature is centered and reduced to unit variance. These four databases are summarized in the first four rows of Table 1.

We randomly partition each data set into two equal halves, keeping the class distributions similar to that of the full data set. Initially, one set is used as training data for the individual

[1] In the event that a tie exists in a BKS unit, we select the output of the most globally accurate classifier.

[2] The ELENA project is a resource of databases and technical reports designed for testing and benchmarking machine-learning classification algorithms. All the databases, their preprocessing, and a technical report describing them in detail are available via anonymous ftp at: *ftp.dice.ucl.ac.be* in the directory *pub/neural-nets/ELENA/databases*.

Table 1: Summary of the data sets used in CMC experiments.

Data Set	Number of Classes	# of Features Available	# of Inputs After Feature Selection	Number of Samples
iris_CR	3	4	4	150
phoneme_CR	2	5	5	5,404
satimage_CR	6	36	5 to 9	6,435
texture_CR	11	40	8 to 11	5,500
mammography	2	63	5 to 15	47,923

classifiers and the CMC algorithms. This includes any feature selection and classifier-specific parameter optimization. The classification accuracy is then evaluated using the other set. Next, the roles of the two sets are reversed. Accuracy is reported as the average of the two results.

3.1 Individual Classifiers

For this round of experiments, up to five individual classifiers are used in the various CMC algorithms, two parametric and three non-parametric. They are: Linear Bayesian, Quadratic Bayesian, K-Nearest Neighbor (K-NN) with the Euclidean distance metric [10], a fully connected backpropagation artificial neural network (ANN) with sigmoid activation functions [11], and the C4.5 decision tree implementation [12].

For a CMC approach to be of practical use, it should improve on the best individual classifier, given that the individual classifiers have been reasonably optimized with regards to parameter settings and available feature data. In our work, an earnest effort is made to optimize each individual classifier with respect to selecting "good" values for the parameters which govern its performance. For brevity, we will omit the details. For the K-NN classifier, a value of K must be determined. For ANNs, the numbers of hidden layers and hidden nodes in a layer must be selected. The parameters for the C4.5 decision tree algorithm are selected based on our previous experience with this classifier. The Bayesian classifiers do not require any sort of parameter selection or optimization.

If each individual classifier is not given the opportunity to select from all features, then the comparison of CMC algorithms to individual classifiers is biased. Table 1 lists the number of features actually used for each data set after applying a feature selection algorithm. The number and specific features actually used depends on the individual classifier. Since the iris and the phoneme data already have a small dimensionality, all features are used by all classifiers in experiments with these two data sets.

3.2 DCS-LA Implementation and Application

The DCS-LA algorithm uses the training data, which may be different for each classifier, and the class assignments made by each classifier. Given an unknown sample, it is first labeled by all the individual classifiers. If all classifiers agree, there is no need to estimate local accuracy. When the individual classifiers disagree, local accuracy is estimated for each classifier, and the decision of the classifier with the highest local accuracy estimate is selected.

Occasionally, two (or more) classifiers with conflicting decisions will have the highest local accuracy estimates. Tie-breaking is handled by choosing the class that is selected most often among the tied classifiers. If a tie still exists, the classifier(s) with the next highest local accuracy will break the tie in the same manner as before. Determining the appropriate size for a local region is part of designing the DCS-LA approach. We ran experiments for various region sizes ranging from $K = 1$ to $K = 51$ using the Euclidean distance metric (since the feature values have been normalized).

3.3 Results

Results for the individual classifiers and the CMC algorithms for the ELENA data sets are summarized in Table 2. We also show the results for an “Oracle” which chooses the correct class if *any* of the individual classifiers did so. This is a theoretical upper bound for all CMC algorithms discussed in this work. Of course, the best individual classifier is a lower bound for any meaningful CMC algorithm.

For the iris data, the Oracle is no better than the Linear Bayes. The upper and lower

Table 2: Classification accuracy for individual classifiers and several CMC algorithms on four real data sets. The best individual and CMC results for each data set are bold.

Method of Classification	Data Set			
	iris_CR	phoneme_CR	satimage_CR	texture_CR
K-Nearest Neighbor	92.00%	**87.76%**	**87.79%**	97.78%
Neural Network	95.33%	79.21%	83.98%	94.85%
C4.5 Decision Tree	92.67%	83.92%	83.50%	88.09%
Quadratic Bayes	95.33%	75.41%	85.78%	**99.04%**
Linear Bayes	**97.33%**	73.00%	83.31%	97.42%
Oracle	97.33%	97.22%	95.64%	99.85%
DCS-LA: Local Class Acc.	-	88.49%	**89.38%**	**99.25%**
DCS-LA: Overall Accuracy	-	87.64%	88.57%	99.16%
Classifier Rank	-	87.31%	87.88%	98.85%
Modified Classifier Rank	-	**88.75%**	88.96%	98.47%
Behavior Knowledge Space	-	85.68%	86.75%	99.05%

performance bounds are identical, and there is no point in using a CMC algorithm. For the phoneme data, the Modified Classifier Rank algorithm performed marginally better than the DCS-LA algorithm using local class accuracy. The other CMC algorithms failed to improve upon the performance of the K-Nearest Neighbor classifier. Results for the satimage data show the DCS-LA algorithm with local class accuracy to be the best CMC algorithm while the BKS algorithm again fails to improve upon the best individual classifier. For the texture data, the DCS-LA algorithm using local class accuracy is best, while the Classifier Rank and Modified Classifier Rank methods degrade performance.

This initial set of experiments permits us to make a couple of interesting observations. First, the DCS-LA algorithm using local class accuracy is the only CMC algorithm that showed some performance improvement for all data sets (excluding the iris data for which it was not possible to improve upon the Linear Bayes classifier). Second, local class accuracy was better than overall local accuracy for the DCS-LA algorithm in all cases. Also, the Modified Classifier Rank method, which uses local class accuracy, generally outperformed the Classifier Rank method, which uses overall local accuracy.

3.4 Altering the Classifier Mix

To test the extent to which the CMC results depend on the mix of individual classifiers, we ran the DCS-LA algorithm for all possible combinations of four out of five classifiers on the three ELENA data sets for which CMC is beneficial. Results are summarized in Table 3.

Table 3: Classification accuracy for the DCS-LA algorithm using all possible combinations of four classifiers as input. KNN = K-Nearest Neighbor, ANN = Artificial Neural Network, LB = Linear Bayes, and QB = Quadratic Bayes.

Classifiers Used as Input to DCS-LA: Local Class Acc.	Data Set		
	phoneme_CR	satimage_CR	texture_CR
KNN, ANN, C4.5, QB	88.60%	**89.39%**	99.05%
KNN, ANN, C4.5, LB	**88.64%**	89.31%	98.84%
KNN, ANN, LB, QB	87.78%	89.28%	**99.34%**
KNN, C4.5, LB, QB	88.60%	89.02%	99.31%
ANN, C4.5, LB, QB	86.81%	88.68%	99.04%

The DCS-LA algorithm outperforms the best individual classifier in all cases. Even more interesting, there exists a combination of four classifiers that is slightly superior than the combination of five classifiers for all three data sets. This tells us that some strategy should be used when selecting the mix of classifiers to use as input to a CMC algorithm.

Not surprisingly, removing the best individual classifier from the combination of five classifiers results in the biggest drop in performance for all three data sets. Also note that removing the single worst classifier results in better performance than the combination of five classifiers in all cases. These results suggest that a sequential backwards search might be an effective technique [13]. The results of a sequential backwards search for the ELENA data sets are shown in Table 4. As redundant or detrimental classifiers are removed from the mix, performance gradually improves. Eventually, we begin removing useful classifiers, and performance gradually drops off.

Table 4: Results of a sequential backwards search technique for selecting a mix of individual classifiers to use as input for the DCS-LA algorithm.

Classifiers Used as Input to DCS-LA: Local Class Acc.	Data Set		
	phoneme_CR	satimage_CR	texture_CR
All 5 Classifiers	88.49%	89.38%	99.25%
Best 4 Classifiers	88.60%	**89.39%**	**99.34%**
Best 3 Classifiers	88.62%	88.94%	99.33%
Best 2 Classifiers	**88.66%**	88.61%	99.16%
Best Individual Classifier	87.76%	87.79%	99.04%

4 Empirical Comparison with ROC Analysis

Our next round of experiments uses a data set from an application in mammogram image analysis [14], summarized in the fifth row of Table 1. Unlike the ELENA data sets, this feature data did not undergo normalization preprocessing. We use well known ROC analysis techniques for performance evaluation.

4.1 ROC Analysis

The accuracy of a classifier (in a 2-class problem) can be characterized by a plot of the classifier's true positive detection rate versus its false positive rate, called a receiver operating characteristic (ROC) curve. The Area Under the ROC Curve (AUC) is an accepted way of comparing *overall* classifier performance [15, 16]. Hanley and McNeil [17] describe methods to determine if the observed difference between two AUCs is statistically significant. These standard statistical methods compare AUCs over the full range of TP rates. Our empirical ROC results only cover a portion of the full range, and so AUCs must be expressed as conditional probabilities prior to applying the methods of Hanley and McNeil.

First, the AUCs over the range of interest are estimated using the trapezoid rule for the discrete operating points. The area under a portion of an ROC curve can be expressed as a conditional probability via the following transformation:

$$AUC = \frac{A_p}{TP_2 - TP_1} \tag{1}$$

where A_p is the area under the ROC curve computed between TP rates TP_1 and TP_2. The formula for the z statistic is

$$z = \frac{AUC_1 - AUC_2}{\sqrt{{SE_1}^2 + {SE_2}^2}} \tag{2}$$

where AUC_1 and AUC_2 are the two estimated AUCs, and SE_1 and SE_2 are the estimated standard errors of each AUC. We use a two-tailed test for statistical significance. The null hypothesis is that the two observed AUCs are the same. The alternate hypothesis is that the two AUCs are different. A critical range of $z > 1.96$ or $z < -1.96$ (a level of significance $\alpha = 0.05$) indicates that the null hypothesis can be rejected.

A conservative estimate of the standard error of an AUC value (from [17]) is:

$$SE(AUC_i) = \sqrt{\frac{\theta(1-\theta) + (n_A - 1)(Q_1 - \theta^2) + (n_N - 1)(Q_2 - \theta^2)}{n_A n_N}} \tag{3}$$

where Q_1 and Q_2 are two distribution-specific quantities, θ is the "true" area under the ROC curve, and n_A and n_N are the number of abnormal and normal samples, respectively. The estimate AUC_i is used as an estimate of θ. The quantities Q_1 and Q_2 are expressed as functions of θ:

$$Q_1 = \frac{\theta}{2-\theta} \quad and \quad Q_2 = \frac{2\theta^2}{1+\theta} \tag{4}$$

Each of the individual classifiers is usually able to generate operating points running from 0% to 100% with fairly small increments between consecutive points. To generate a single operating point for a CMC algorithm, the individual classifiers are all set to approximately the same level of sensitivity, and the CMC is executed. This procedure is repeated with the individual classifiers set to other sensitivity levels, resulting in a series of operating points for each CMC algorithm. The ROC curves generated for each CMC algorithm will not cover the full range of TP rates. Therefore, in a test for statistical significance, two ROC curves are compared *only* over the range of TP rates that are common to both curves.

4.2 DCS-LA Implementation and Application

For the mammography data, results of a CART decision tree classifier [18] were available in addition to those of the other five classifiers that were used for the ELENA data sets. For the most part, feature selection, classifier parameter optimization, and the DCS-LA implementation are done same as before. One exception is that the Mahalanobis [10] distance metric is used for the K-Nearest Neighbor classifier and the DCS-LA algorithm since the data has not been normalized.

Also, we would like to investigate the effect of setting the individual classifiers to various sensitivity levels prior to applying CMC. We tested all CMC algorithms with the individual classifiers set to 6 different TP rates: 70%, 75%, 80%, 85%, 90%, and 95%. If a classifier could not be set exactly to the desired TP rate desired, it was set as close as possible. As before, we ran experiments for various region sizes ranging from $K = 1$ to $K = 51$ for each of the 6 levels of individual classifier sensitivity.

4.3 Results

We show only those results obtained when the first half of the data set is used as training data. Nearly identical results were obtained for the experiments which utilized the other half of the data set in the training capacity.

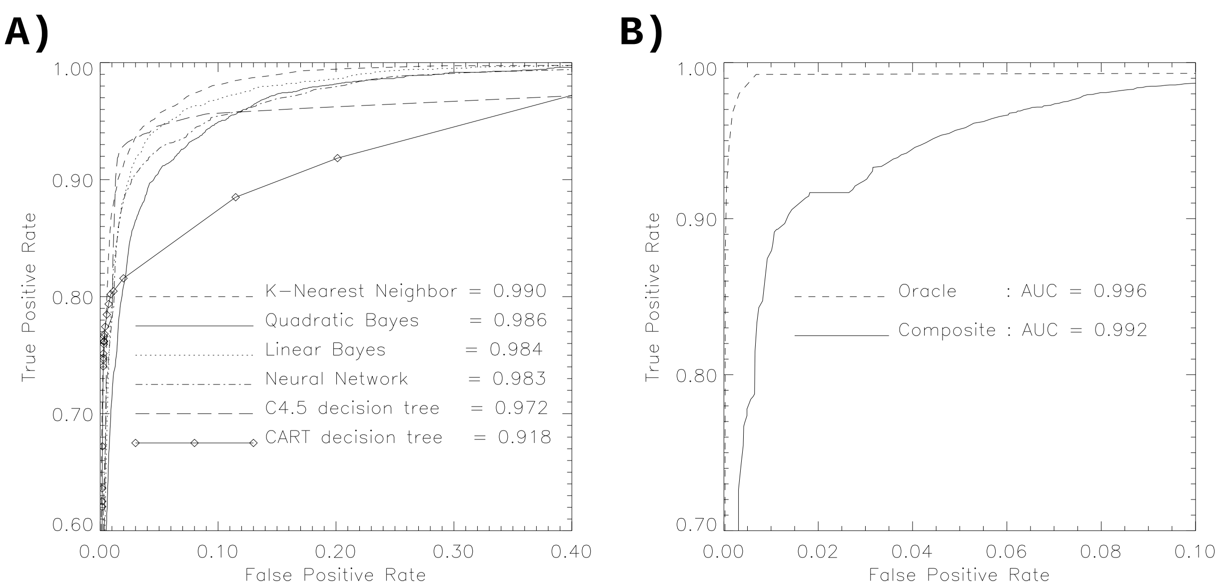

Figure 1: A) Partial ROC curves for 6 individual classifiers, and their AUCs. B) Composite ROC curve for the individual classifiers, and the ROC curve for an Oracle classifier.

Figure 1A shows partial ROC curves[3] plotted for all 6 individual classifiers. The best individual classifier is KNN if the overall AUC is considered. However, there is no single best classifier across all TP rates. As a benchmark for useful CMC performance, we consider a composite ROC curve consisting of the "best" parts of the individual ROC curves. The composite ROC is a lower bound for practical CMC performance. We also plot ROC curves for an Oracle classifier, the theoretical upper bound on CMC performance. The composite and Oracle ROC curves are shown in Figure 1B.

A comparison of the ROC curves generated by the DCS-LA algorithm using both methods of local accuracy estimation shows that local class accuracy is superior to overall local accuracy. The difference in AUCs, however, is not statistically significant ($z = 1.44$ for TP rates ranging from 78% to 94%). Further ROC analysis of the DCS-LA algorithm with various local region sizes shows that regions defined by $K = 10$ generally seem to result in the best performance for this data set.

Figure 2 compares the composite ROC curve with the results for DCS-LA using local class accuracy. We also show the results of the Behavior Knowledge Space, Classifier Rank, and the Modified Classifier Rank algorithms. To be fair, only the best single value of K (10) is used in the plot for the DCS-LA results. Thus, the ROC curves for all four CMC algorithms are composed of 6 operating points each.

The DCS-LA algorithm is better than the best individual classifier at all times. The difference between the AUCs, computed over the range of common TP points (from 82% to 93%), for DCS-LA ROC curve and the Composite ROC curve is statistically significant ($z = 3.51$). The Modified Classifier Rank method performs nearly as well as DCS-LA at lower sensitivities, but less so at higher levels. It is significantly better than the best individual classifier ($z = 2.71$) over the common TP range (82% to 88%). The Classifier Rank method provides improvement, though not statistically significant, at some levels of sensitivity. As with our initial set of experiments, the Behavior Knowledge Space method is not able to

[3]Partial ROC curves are plotted in order to focus on a region of interest. In a medical application such as ours, high sensitivity levels are required.

improve upon the performance of the optimized individual classifiers. The DCS-LA method performed significantly better than the Behavior Knowledge Space method ($z = 4.91$ for TP rates ranging from 84% to 92%), and the Classifier Rank method ($z = 3.81$ for TP rates ranging from 82% to 91%).

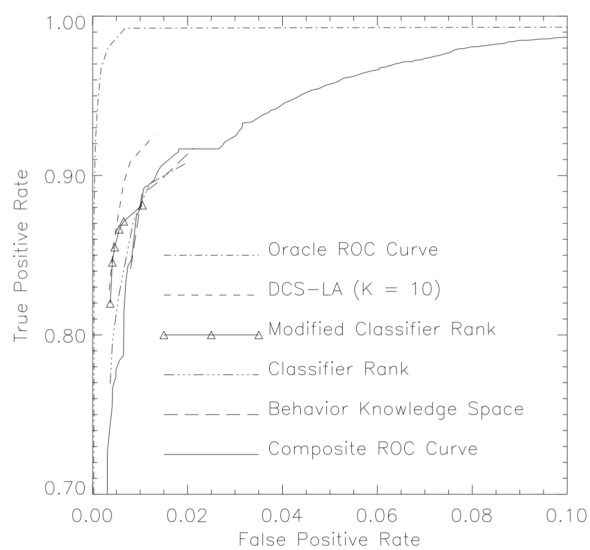

Figure 2: The composite and Oracle ROC curves for the 6 individual classifiers compared to the results for the DCS-LA, Behavior Knowledge Space, Classifier Rank, and Modified Classifier Rank methods.

Table 5 shows the results of the CMC algorithms when the individual classifiers are set (as close as possible) to a TP rate of 80%. The DCS-LA algorithm uses local class accuracy with $K = 10$. The number of times each individual classifier was selected by the DCS-LA algorithm is also shown. In this example, the DCS-LA algorithm finds operating points with higher TP rates *and* lower FP rates than points obtained by any individual classifier. All classifiers agree on the class assignment for a majority of the test samples (89.5%), and therefore any of the CMC algorithms are actually executed a relatively small percentage of the time. The number of times an individual classifier is selected by the DCS-LA algorithm seems closely correlated to the overall accuracy of the classifier. Results at other sensitivity levels show the same general trends.

Table 5: CMC results with individual classifiers set to TP rates as close to 80% as possible. All classifiers agree for 22,552 of the samples, or about 89.5% of the time.

Method of Classification	Set at (TP rate, FP rate)	Overall Accuracy	# times classifier selected by DCS-LA
Neural Network	(80.1, 0.85)	95.3%	1287
K-Nearest Neighbor	(79.8, 0.87)	95.2%	425
CART decision tree	(80.5, 1.11)	95.1%	444
C4.5 decision tree	(78.4, 0.57)	95.1%	287
Quadratic Bayes	(80.0, 1.84)	94.4%	125
Linear Bayes	(80.2, 1.97)	94.4%	67
Oracle	(94.7, 0.11)	98.8%	-
DCS-LA: Local Class Acc.	(87.7, 0.55)	97.0%	-
Behavior Knowledge Space	(89.7, 1.42)	96.8%	-
Classifier Rank Method	(82.6, 0.56)	96.0%	-
Modified Classifier Rank	(85.5, 0.46)	96.7%	-

In general, since the DCS-LA algorithm is attempting to lower the total number of misclassifications, it generates operating points which make the appropriate TP/FP trade-off in order to drive the *overall* error rate down. Consider when all the individual classifiers are set to lower sensitivities (approximately less than 90%). Given the number of test samples per class, it is possible to misclassify fewer total samples by trading off a higher TP rate for a corresponding higher FP rate. By contrast, when we set all classifiers to TP rates of approximately 95%, the DCS-LA algorithm usually generated an operating point with a TP rate lower than 95%. In this situation, trading off the lower TP rate for the corresponding lower FP rate resulted in fewer total classification errors, and therefore an improved overall accuracy.

5 Summary and Conclusions

We have shown that even if all the individual classifiers have been optimized, dynamic classifier selection by local accuracy is still capable of improving overall performance significantly. By contrast, simple voting techniques, and even a recently proposed CMC algorithm, were not able to show any significant improvement when the individual classifiers were sufficiently optimized. At times, some of the other CMC algorithms actually hurt performance. The proposed DCS-LA algorithm was *always* capable of improving performance.

In this work, we have attempted to address some issues relevant to the construction of a multiple classifier system which have not previously received attention. First, we have made efforts to optimize the individual classifiers with respect to the available feature data. Certainly it would be preferable to use a single classifier as opposed to a combination of several classifiers if the performance of the two systems is equivalent. Second, we have suggested a systematic procedure for determining if certain classifiers are redundant or detrimental, and could therefore be removed from the mix of individual classifiers prior to CMC. The end result is improved performance, and faster execution time. Finally, we observed the effect of varying the sensitivity of the individual classifiers on the CMC algorithm.

The benefits of a CMC approach may be limited when there is a very small amount of training data, or when the classification accuracy of an individual classifier is sufficiently high. Thus, we believe the greatest potential for CMC algorithms is for large data sets with data distributions that are too complex for most individual classifiers.

References

[1] L. Lam and C. Y. Suen, "A theoretical analysis of the application of majority voting to pattern recognition," in *Proceedings of the 12th International Conference on Pattern Recogntion and Computer Vision*, (Jerusalem, Israel), pp. 418–420, 1994.

[2] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[3] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 66–75, January 1994.

[4] F. Kimura and M. Shridar, "Handwritten numerical recognition based on multiple algorithms," *Pattern Recognition*, vol. 24, no. 10, pp. 969–983, 1991.

[5] C. Nadal, R. Legault, and C. Y. Suen, "Complementary algorithms for the recognition of totally unconstrained handwritten numerals," in *Proceedings of the 10th International Conference on Pattern Recogntion*, (Atlantic City, NJ), pp. 443–449, 1990.

[6] E. Mandler and J. Schurmann, "Combining the classification results of independent classifiers based on the dempster-shafer theory of evidence," in *Pattern Recognition and Artificial Intelligence*, (North-Holland), pp. 381–393, Elsevier Science Publishers B.V., 1988.

[7] Y. S. Huang and C. Y. Suen, “A method of combining multiple classifiers - a neural network approach,” in *Proceedings of the 12th International Conference on Pattern Recogntion and Computer Vision*, (Jerusalem, Israel), pp. 473–475, 1994.

[8] Y. S. Huang and C. Y. Suen, “A method of combining multiple experts for the recognition of unconstrained handwritten numerals,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90–94, 1995.

[9] M. Sabourin, A. Mitiche, D. Thomas, and G. Nagy, “Classifier combination for hand-printed digit recognition,” in *Proceedings of the 2nd International Conference on Document Analysis and Recognition*, (Tsukuba Saenie City, Japan), pp. 163–166, 20-22 Oct. 1993.

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

[11] K. Knight, “Connectionist ideas and algorithms,” *Commun. of the ACM*, vol. 33, no. 11, pp. 59–74, 1990.

[12] J. R. Quinlan, *C4.5 Programs for Machine Learning*. Morgan Kaufmann, 1992.

[13] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Prentice-Hall, 1982.

[14] W. P. Kegelmeyer, Jr. and M. C. Allmen, “Dense feature maps for detection of calcifications,” in *Digital Mammography: Proceedings of the 2nd International Workshop on Digital Mammography*, vol. 1069 of *International Congress Series*, (York, England), pp. 3–12, Elsevier Science B. V., July 10-12 1994.

[15] C. E. Metz, “ROC methodology in radiologic imaging,” *Investigative Radiology*, vol. 21, pp. 720–733, 1986.

[16] J. A. Swets, “ROC analysis applied to the evaluation of medical imaging techniques,” *Investigative Radiology*, vol. 14, pp. 109–121, 1979.

[17] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a reciever operating characteristic (ROC) curve,” *Radiology*, vol. 143, pp. 29–36, 1982.

[18] L. Breiman, J. H. Friedman, R. A. Olsen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984.