Decision Combination in Multiple Classifier Systems [1]

Tin Kam Ho[2], *Jonathan J. Hull, Sargur N. Srihari*

Center for Document Analysis and Recognition
State University of New York at Buffalo
Buffalo, NY 14260, USA

Abstract

A multiple classifier system is a powerful solution to difficult pattern recognition problems involving large class sets and noisy input, because it allows simultaneous use of arbitrary feature descriptors and classification procedures. Decisions by the classifiers can be represented as rankings of classes so that they are comparable across different types of classifiers and different instances of a problem. The rankings can be combined by methods that either reduce or rerank a given set of classes. An intersection method and a union method are proposed for class set reduction. Three methods based on the highest rank, the Borda count, and logistic regression are proposed for class set reranking. These methods have been tested in applications on degraded machine-printed characters and words from large lexicons, resulting in substantial improvement in overall correctness.

1 Introduction

Traditional pattern recognition systems use a single feature descriptor and a particular classification procedure to determine the true class of a given pattern. For problems involving a large number of classes and noisy inputs, perfect solutions are often difficult to achieve. Recently, it has been observed that features and classifiers of different types complement one another in classification performance [5] [12] [13] [20] [21] [30]. This has led to a belief that by using features and classifiers of different types simultaneously, classification accuracy could be improved. However, the combination of potentially conflicting decisions by multiple classifiers remains an unsolved problem. Ideally, the combination function should take advantage of the strengths of the individual classifiers, avoid their weaknesses, and improve classification accuracy. In this paper, we provide a set of methods useful for these purposes. While a thorough theoretical investigation is beyond the scope of this paper, we will demonstrate the effectiveness of the methods by experimental results.

Previous methods for classifier combination include intersection of decision regions [9], voting methods [28], prediction by top choice combinations [33], and use of Dempster-Shafer theory [27] [34]. In the methods of [9] [28] [33], only the top choice from each

[1] This paper summarizes the first half of [13]. Preliminary versions have appeared in [12] [14] [15] [17].

[2] Tin Kam Ho has been with AT&T Bell Laboratories (600 Mountain Ave., Murray Hill, NJ 07974, USA) since August 1992.

classifier is used, which is usually sufficient for problems with a small number of classes. But for problems with many classes, as correct rate at top choices drops, secondary choices often contain near-misses that should not be overlooked. In [33] all possible combinations of top choices for a given set of classes are examined. A test is proposed for the dominance of each true class occurring with each combination that would justify the final assignment of the input to that class. Reliable decisions are achieved by exhaustive enumeration, and therefore the method is expensive. For n classes and k classifiers, n^k combinations need to be covered by the training data to sufficient density and the method is impractical for large number of classes. Many confidence-based combination methods suffer from a lack of consistency in the definition of confidence measures for different instances of a given problem and different types of classifiers. The methods proposed in this paper are motivated by an attempt to overcome these difficulties. As we will see, the examination of the strengths and weaknesses of each method leads to the problem of determining classifier correlation, which is the central issue in deriving an effective combination method. We attempt to analyze classifier correlation by a statistical model based on logistic regression.

Our methods have been tested in several OCR applications including handwritten digit recognition [14] and degraded multifont machine-printed character and word recognition [18]. The strengths of the methods are best demonstrated in problems involving a large number of classes. In a word recognition experiment where four classifiers were used to discriminate between 1,365 classes, an improvement of 7.8 percent at the top choice over that of the best individual classifier was achieved.

Organization of Discussions. In Section 2 we discuss the reasons for our use of class rankings to represent class decisions, and two objectives of decision combination, namely, class set reduction and reordering. Methods for achieving each objective are proposed in Sections 3 and 4. The use of multiple classifiers allows dynamic classifier selection in response to each input and a multi-stage combination. Issues involved in dynamic classifier selection and multi-stage organizations are discussed in Section 5. Section 6 describes experimental results.

2 Inputs and Outputs of Decision Combination Functions

A decision combination function must receive useful representations of classifier decisions. We decide to use rankings of classes instead of unique class choices or numerical scores computed for each input. Rankings contain more information than unique choices for a many-class problem. For a mixture of classifiers of various types, numerical scores such as distances to prototypes, values of an arbitrary discriminant, estimates of posterior probabilities, and confidence measures are not directly usable because of incomparability of their scales, and in some cases inconsistence across different instances of a problem

[3]. Rankings are on a weaker scale to which all such scores can be easily converted. Combination methods based on rankings are therefore more general and applicable to a mixture of classifiers of arbitrary types.

For simplification, we assume that in each combination, the same number of distinct ranks are used by each classifier for a given set of classes. In practice, it is possible that ties exist in the rankings produced by some classifiers. If this happens, we suggest that the rank scores on the finer scales be converted to ranks on the coarsest scale. For example, suppose that four classifiers are to be combined, one of which can only accept a single class as being correct and rejects all other classes. In this case, even though the other classifiers can produce a complete ranking of the class set, these rankings should be converted to binary scores, with one score for the top choice and another for the rest. Alternatively, the classifiers using the same number of distinct ranks may be combined first, and the combined rankings can then be converted to the same scale as those used by the others, so that they can be combined at a second stage.

A multiple classifier system is justified only if the combined decisions are better than those of any single classifier in the system. The comparison of performance can be based on two criteria, which suggest two different approaches to decision combination. We will refer to these two approaches as *class set reduction* and *class set reordering*. In class set reduction, the objective is to extract a subset from a given set of classes, such that the subset is as small as possible yet still contains the true class. In class set reordering, the objective is to derive a consensus ranking of the given classes, such that the true class is ranked as close to the top as possible.

It is useful to differentiate between the two objectives because they can be achieved by different means. Some methods for decision combination produce a small subset that hardly ever misses the true class, but the classes within the subset are not ordered. Other methods may produce good rankings where the true class is often ranked close to the top, but, occasionally for a bad input, the true class may be far away from the top, so that it will be missed if only a small neighborhood is taken.

These two objectives are equivalent under special conditions: (1) If it is required that the result set derived by a reduction method always contains only one class, then it is the same as requiring a reordering method to rank the true class always at the top. (2) If the rankings derived by a reordering method are so good that the true class is always ranked above a certain position, then it is always possible to include the true class in a neighborhood up to that position. The two approaches can also be applied to the same problem, so that the set of classes may first be reduced and then reranked, or first reranked and then reduced to a small neighborhood near the top of the ranking.

[3]For instance, the distance between characters 'c' and 'o' could be smaller than the distance between two instances of the character 'w', yet the smaller distance does not necessarily lead to a more correct decision.

The rest of this paper details methods that are useful for reducing or reordering decisions. In Section 3, two approaches to class set reduction as well as some approximation methods are discussed. In Section 4, three methods for class set reordering are presented. Certain classifiers may be redundant in a combination and can be eliminated for efficiency. A simple case is that two classifiers producing identical decisions for every input. In other cases, the significance of each classifier depends on the particular decision combination method. We will discuss the conditions under which a classifier is redundant in the context of each combination method.

3 Methods for Class Set Reduction

Class set reduction is aimed at reducing the number of classes in the output list without losing the true class. The criteria for success are therefore two-fold: the size of the result set should be minimized, and the probability of inclusion of the true class should be maximized. Two simple and direct methods can be used for these purposes. Both methods attempt to derive a threshold on the ranks according to the worst case ranks of the true classes.

The first method computes the intersection of large neighborhoods taken from each classifier. The sizes of the neighborhoods are determined by the ranks of the true classes in the worst cases in the training set. After rankings are obtained for all training patterns, the lowest rank ever given by each classifier to any true class is determined. These lowest ranks are taken as thresholds on the ranks. An example of threshold computation is shown in Table 1. For a test pattern, classes ranked above the thresholds are selected and intersected. In this method, a classifier is redundant if its threshold is equal to the size of the class set.

Table 1: An example of determining neighborhood sizes for an intersection.

	$rank_j^i$ of classifier C_j			
input I_i	C_1	C_2	C_3	C_4
I_1	3	12	1	24
I_2	1	5	29	12
I_3	34	3	4	6
I_4	9	7	6	7
I_5	4	36	5	5
I_6	16	2	3	4
thresholds ($colmax_j$)	34	36	29	24

The second method computes the union of small neighborhoods taken from each classifier. The thresholds on the ranks are selected by a max-min procedure illustrated in Table 2. The left half of Table 2 shows the ranks of the true class of each training pattern.

The best (minimum) rank in each row is determined, and entered under the classifier that produces it in the right half of the table. The maximum of all these minima is computed for each column. It can be shown that, if a neighborhood is obtained from each classifier by thresholding the ranks using these maxima, a union of the neighborhoods for each training pattern always contains its true class.

Table 2: An example of determining neighborhood sizes for a union.

	$rank_j^i$ of classifier C_j				$rowmin_j^i$			
input I_i	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
I_1	3	12	1	24	0	0	1	0
I_2	1	5	29	12	1	0	0	0
I_3	34	3	4	6	0	3	0	0
I_4	9	7	6	7	0	0	6	0
I_5	4	36	5	5	4	0	0	0
I_6	16	2	3	4	0	2	0	0
$thresholds\ (colmax_j)$					4	3	6	0

With the union method, any classifier j with threshold $colmax_j$ being 0 is redundant, meaning that its decision is always inferior to some other classifier's and should not be included in the union. Classifier C_4 in Table 2 is redundant in this sense. Other than the obvious redundancy indicated by a threshold of zero, a classifier can also be redundant in the sense that the union size can be smaller if that classifier is not used. For instance, using only the classifiers $\{C_1,C_3\}$ in the above example could result in a smaller union size. However, to determine the subset of classifiers optimal in this sense requires an exhaustive search over all the subsets [17].

Obviously, the intersection approach is useful only when all the classifiers have moderate worst case performance, so that small neighborhoods are obtained without missing the true classes. However, this is usually not the case for a set of specialized classifiers using a small number of features, so that each classifier may be excellent for certain types of inputs but poor for others. The neighborhoods may be undesirably large, since they are determined by the worst case behavior of the classifiers.

The union approach is preferred if the classifiers are specialized on different types of inputs. The ideal case is that the set of classifiers is sufficiently rich and all types of inputs are included in their specialties. That is, for each pattern, there is always one classifier that recognizes it well. Referring back to Table 2, this corresponds to the case when the row minima are small and so are the column maxima. Small neighborhoods and hence a small union are obtained in such cases. Essentially, the union approach focuses on the best case behavior of each classifier.

The thresholds given by the max-min procedure are absolute, in the sense that they guarantee 100 percent success in including the true class for every training pattern. Because of this, the effectiveness of the method is sensitive to outlying worst cases. This

is also true for the intersection method. In practical applications, if there are few outlying cases, and if the cost of a small number of errors is affordable, an approximation method is preferred. Approximations can be made by removing extremely bad cases from the training set according to the desired accuracy, or by using votings instead of intersections or unions.

4 Methods for Class Set Reordering

Class set reordering attempts to improve the rank of the correct class. The criterion for success is the position of the true class in the resultant ranking, as compared to its position in the rankings before combination. A method is considered successful if the probability of having the true class near the top of the combined ranking is higher than that in each of the original rankings. We give three methods for this purpose in the followings.

The Highest Rank Method

Similar to the union approach, the highest rank method is good for combining a small number of classifiers each of which specializes on inputs of a particular type. Assume that, for each input pattern, m classifiers are applied to rank a given set of classes. Thus each class receives m ranks. The minimum (highest) of these m ranks is assigned to that class as its score. The classes are then sorted by these scores to derive a combined ranking for that input. Ties in the combined ranking may be broken arbitrarily to achieve a strict linear ordering.

This reordering method is particularly useful in a problem involving a large number of classes and few classifiers. The advantage is its ability to utilize the strength of each classifier. For any input pattern, as long as there is one classifier that performs well and ranks the true class near the top, say, at rank k, no matter how the other classifiers perform, the true class will be at a position no farther than $k \times m$ from the top in the combined ranking, where m is the number of classifiers. Using this method, a classifier is redundant if the rank it assigns to a true class is always lower than those assigned by other classifiers.

One disadvantage with this method is that the combined ranking may have many ties. The number of classes sharing the same ranks depends on the number of classifiers used. Therefore, this method is useful only if the number of classifiers is small relative to the number of classes. Otherwise, most of the classes are involved in ties and the final ranking is not interesting.

The Borda Count Method

In the context of group decision theory, the mapping from a set of individual rankings to a combined ranking is referred to as a *group consensus function*. One useful group consensus function is referred to as the *Borda count* [4], which is a generalization of the majority vote. The Borda count for a class is the sum of the number of classes ranked below it by each classifier. The consensus ranking is given by arranging the classes so that their Borda counts are in descending order.

The magnitude of the Borda count for each class measures the strength of agreement by the classifiers that the input pattern belongs to that class. For a two-class problem, the Borda count is equivalent to the simple majority vote. Variations of the Borda count function, such as those for handling ties in the rankings, are discussed in [4].

The Borda count function assumes additive independence among the contributions of the individual classifiers. Using this method, a classifier is redundant if it always reinforces errors made by the others, that is, if all the classes it ranks above a true class are always contained in some other classifier's choices above the true class.

The Borda count method is simple to implement, and requires no training. However, it does not take into account the differences in the individual classifier capabilities. All classifiers are treated equally, which may not be preferable when we know that certain classifiers are more likely to be correct than the others.

Logistic Regression

In order to combine classifiers with nonuniform performances, the Borda count method needs to be modified by assigning weights to the rank scores produced by each classifier. The weights should reflect the relative significance of each classifier evaluated in the context of the combination. Moreover, it will be useful to measure the confidence of the combined decisions given by the Borda count method. Possible measures include a statistic derived from the distribution of sums of a given range of ranks, and the intervals between the computed Borda counts for a given set of classes [11] [29].

However, the distribution of rank sums is affected by the correlation between the classifiers and does not necessarily indicate classification correctness. For instance, if rankings by two identical classifiers are combined, the rank sum for the class that is their common top choice falls on an extreme of the distribution, whether that decision is correct or not. The two effects, namely, classification correctness and classifier correlation, must be distinguished and modeled separately.

Motivated by the need to distinguish the correct classes from the incorrect ones, we associate a binary variable Y_c to each class c for a given pattern. Y_c has the value 1 if c is the true class of that pattern, and 0 otherwise. The goal of recognition is therefore

to predict the value of Y_c for each class c. Hence the decision combination problem can be reformulated in the context of regression analysis. The rank scores produced by each classifier are considered as random variables that are used to predict the value of Y_c for each class c, and their effects on Y_c can be modeled by a multiple regression function. Since Y_c is binary, a logistic response function is useful in this context [1] [6].

For simplicity in notation, we denote the response variable Y_c by Y, which has a value for each class with respect to each input pattern. $Y = 1$ for the true class and $Y = 0$ for other classes. For a training pattern, the true class is known and therefore each class has a known value of Y. For an unseen pattern, the value of Y for each class has two possible outcomes.

Denote the probability $P(Y = 1|\mathbf{x})$ by $\pi(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, ..., x_m)$ represents the rank scores assigned to that class by classifiers $C_1, C_2, ..., C_m$. For convenience in discussion, we assume that x_i has the largest value if the class is ranked at the top by C_i. Using the logistic response function,

$$\pi(\mathbf{x}) = \frac{\exp\ (\alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m)}{1 + \exp\ (\alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m)},$$

and

$$log\ \frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})} = (\alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m),$$

where α, $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_m)$ are constant parameters.

The transformation $L(\mathbf{x}) = log\ \frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})}$ is referred to as the *log-odds*, or the *logit*, and is linearly related to $\mathbf{x}$. The logit transformation links the problem to linear regression analysis. Methods based on maximum likelihood or weighted least-squares can be used to estimate the model parameters $\alpha, \beta_1, \beta_2, ..., \beta_m$ [1] [6]. The relative magnitudes of the parameters indicate the relative significances of the classifiers in their marginal contribution to the logit. Hence the parameters can be used as weights for the rank scores.

For each test pattern, the logit for each class is predicted by the estimated model. If only a ranking of the class set is needed, the classes can simply be sorted by the predicted logits in descending order. The class with the largest logit is then considered as most likely to be the true class. The values of $\pi(\mathbf{x})$ or the logit can also be used as a confidence measure. A threshold on these values can be determined experimentally, so that classes with confidences lower than the threshold can be rejected.

Example. Weight estimation by logistic regression is illustrated with an example application in word recognition, where two classifiers were used to recognize an image as one of 67,305 classes.

We consider only the top 10 choices from each classifier, and use the largest score to represent a top choice. That is, a class receives a 10 if it is ranked at the top. A class receives a 0 if it is ranked below the 10th position. Using a union method together with three other classifiers, a neighborhood of up to 50 observations were taken for each image. A total of 43,422 observations were obtained using 1,055 training images.

The distribution of the rank scores in these observations and the empirical logits derived from this set of data are shown in Table 3. A plot of the empirical logits versus x_1 and x_2 is given in Figure 1. A regression plane was fit to these logits by the SAS procedure LOGISTIC [32]. Figure 2 shows the estimated regression plane. The parameter estimates are given in Table 4. Both x_1 and x_2 are significant according to the parameter estimates. The estimated regression model is

$$L(\mathbf{x}) = -5.8557 + 0.1965\, x_1 + 0.4008\, x_2.$$

Remarks. In an ordinary logistic regression analysis, the Chi-square value computed for each model parameter can be used to evaluate the statistical significance. However, caution has to be taken in this application. Because there is only one true class for each pattern, $Y = 1$ for one class implies $Y = 0$ for all the others. In other words, values of the response variable Y are related for each pattern but independent across different patterns. This may lead to the problem of *overdispersion* or underestimates of the standard errors [6] [26]. Nevertheless, the parameter estimates are unaffected, and the relative significance of each classifier can be told by the relative magnitude of those estimates. The validity of a model can be evaluated experimentally by observing its performance on a test set.

The residual plot can be used to examine if there is a systematic lack of fit between the estimated values and the actual values of the logits. Goodness of fit largely depends on whether the linearity assumption is satisfied by the empirical logits. The linearity assumption may be invalid if more ranks instead of a small number of top decisions are used. This is because, for most classifiers, $\pi(\mathbf{x})$ increases much more rapidly near the top of the rankings than it does at lower positions. The surface formed by the empirical logits is likely to be curvilinear on a large rank scale.

This problem can be overcome in three ways. A better fit may be obtained by using a nonlinear regression model. A second solution is to truncate the rankings at a certain threshold. The decisions at positions lower than the threshold may simply be grouped together and assigned a single rank, just like what we did in the previous example (assigning 0 to all classes ranked below 10). A third solution is to attempt a piecewise linear fit. For discussions on goodness-of-fit, diagnostic procedures, and model building techniques, interested readers are referred to [1] [2] [19] [25] [26].

Using the same definition of Y, the problem can be reformulated as a two-class discrimination problem and solved by discriminant analysis. However, it has been ob-

Table 3: Example distribution of rank scores and the empirical logits.

x_1:	rank by classifier 1
x_2:	rank by classifier 2
N:	number of classes receiving (x_1, x_2)
$N_{(Y=1)}$:	number of true classes at (x_1, x_2)
$\pi(\mathbf{x})$:	$N_{(Y=1)}/N$

x_1	x_2	N	$N_{(Y=1)}$	$\pi(\mathbf{x})$	$log_e(\pi(\mathbf{x})/(1-\pi(\mathbf{x})))$	x_1	x_2	N	$N_{(Y=1)}$	$\pi(\mathbf{x})$	$log_e(\pi(\mathbf{x})/(1-\pi(\mathbf{x})))$
0	0	25145	179	0.007	-4.938	6	0	802	3	0.004	-5.585
0	1	918	6	0.007	-5.024	6	3	21	1	0.048	-2.996
0	2	912	1	0.001	-6.815	6	6	18	1	0.056	-2.833
0	3	889	5	0.006	-5.175	6	8	34	2	0.059	-2.773
0	4	861	5	0.006	-5.143	6	9	44	2	0.045	-3.045
0	5	852	8	0.009	-4.659	6	10	20	9	0.450	-0.201
0	6	832	14	0.017	-4.068	7	0	761	4	0.005	-5.243
0	7	748	16	0.021	-3.823	7	1	14	1	0.071	-2.565
0	8	704	19	0.027	-3.585	7	5	24	1	0.042	-3.135
0	9	601	39	0.065	-2.668	7	6	28	3	0.107	-2.120
0	10	410	94	0.229	-1.212	7	7	32	2	0.062	-2.708
1	2	18	1	0.056	-2.833	7	8	43	1	0.023	-3.738
1	4	12	1	0.083	-2.398	7	9	47	4	0.085	-2.375
1	5	13	2	0.154	-1.705	7	10	41	25	0.610	0.446
1	9	13	1	0.077	-2.485	8	0	707	2	0.003	-5.865
1	10	10	6	0.600	0.405	8	3	19	1	0.053	-2.890
2	0	896	4	0.004	-5.407	8	7	39	1	0.026	-3.638
2	3	13	1	0.077	-2.485	8	8	56	3	0.054	-2.872
2	4	16	1	0.062	-2.708	8	9	56	6	0.107	-2.120
2	7	15	1	0.067	-2.639	8	10	58	33	0.569	0.278
2	9	23	2	0.087	-2.351	9	0	625	7	0.011	-4.481
2	10	10	1	0.100	-2.197	9	2	13	1	0.077	-2.485
3	0	879	2	0.002	-6.083	9	5	30	1	0.033	-3.367
3	3	14	2	0.143	-1.792	9	6	29	4	0.138	-1.833
3	8	16	1	0.062	-2.708	9	8	58	6	0.103	-2.159
3	9	31	3	0.097	-2.234	9	9	132	16	0.121	-1.981
3	10	19	9	0.474	-0.105	9	10	78	48	0.615	0.470
4	0	849	1	0.001	-6.743	10	0	469	6	0.013	-4.346
4	1	21	1	0.048	-2.996	10	1	12	1	0.083	-2.398
4	8	32	3	0.094	-2.269	10	3	12	1	0.083	-2.398
4	9	24	1	0.042	-3.135	10	4	14	1	0.071	-2.565
4	10	15	6	0.400	-0.405	10	5	23	2	0.087	-2.351
5	0	821	1	0.001	-6.709	10	6	20	5	0.250	-1.099
5	2	12	1	0.083	-2.398	10	7	24	6	0.250	-1.099
5	7	40	1	0.025	-3.664	10	8	36	8	0.222	-1.253
5	8	31	4	0.129	-1.910	10	9	63	22	0.349	-0.623
5	9	21	2	0.095	-2.251	10	10	375	331	0.883	2.018
5	10	19	8	0.421	-0.318						

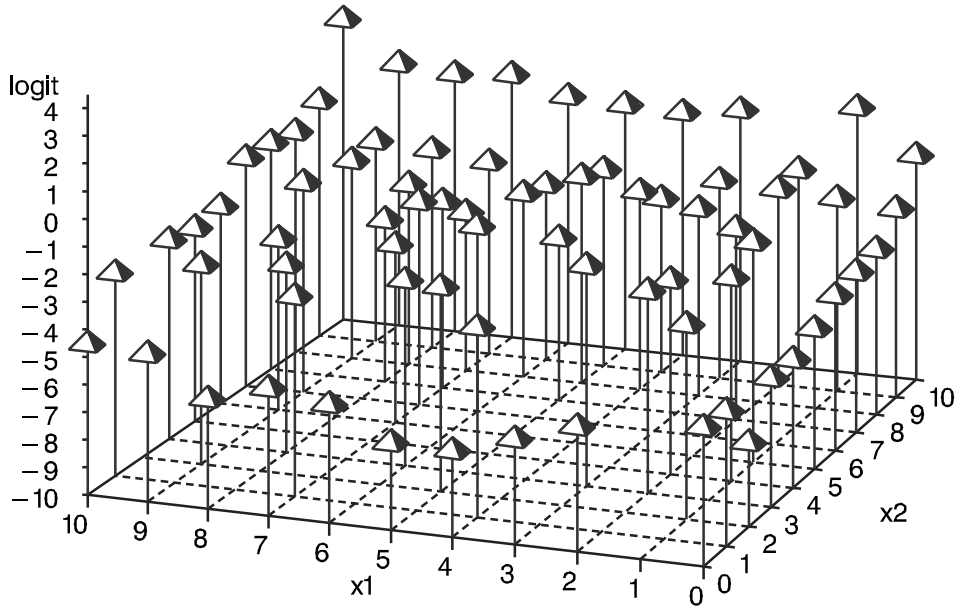


Figure 1: Plot of empirical logits versus ranks by two classifiers.

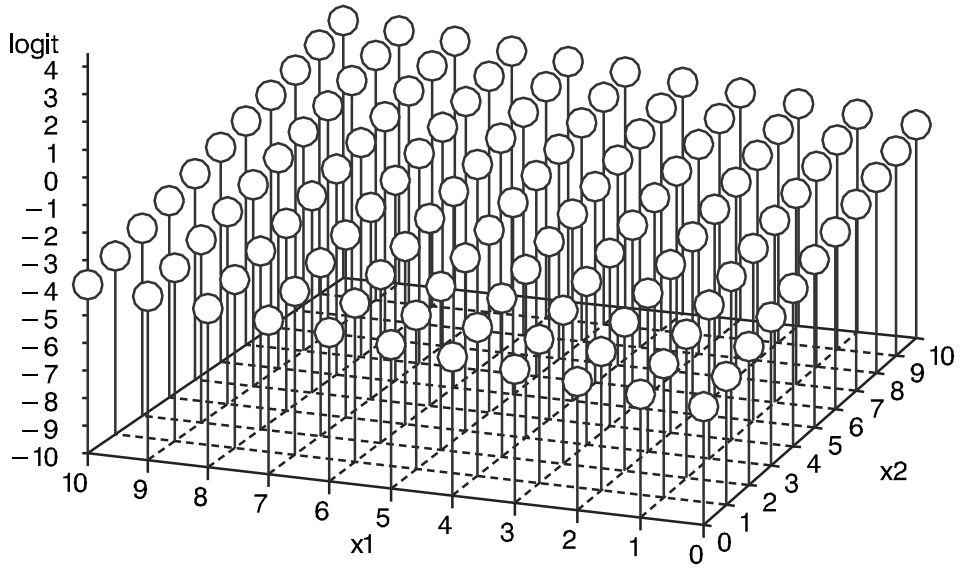


Figure 2: Plot of fitted logits versus ranks by two classifiers.

Table 4: Analysis of maximum likelihood estimates in logistic regression.

Variable	Parameter Estimate	Standard Error	Wald Chi-Square	Pr > Chi-Square	Standardized Estimate
INTERCEPT	-5.8557	0.0779	5656.2065	0.0001	.
X1	0.1965	0.00883	495.1244	0.0001	0.297987
X2	0.4008	0.0103	1516.6643	0.0001	0.607888

served that, if the normality assumption for $\mathbf{x}$ is invalid, which is true in our case, logistic regression is preferred over discriminant analysis [8] [31].

Our use of a single response variable Y for all the classes is a simplification that limits the number of parameters to be estimated. In cases when the number of classes is small and sufficient training data are available, a more elaborated model can be constructed by using a different response variable Y_c for each class c[10]:

$$
\begin{aligned}
log\ \frac{\pi_1(\mathbf{x})}{1\ -\ \pi_1(\mathbf{x})} &= \alpha_1\ +\ \sum_{j=1}^{m}\beta_{1j}x_j, \\
log\ \frac{\pi_2(\mathbf{x})}{1\ -\ \pi_2(\mathbf{x})} &= \alpha_2\ +\ \sum_{j=1}^{m}\beta_{2j}x_j, \\
&\ldots \\
log\ \frac{\pi_n(\mathbf{x})}{1\ -\ \pi_n(\mathbf{x})} &= \alpha_n\ +\ \sum_{j=1}^{m}\beta_{nj}x_j
\end{aligned}
$$

where $\pi_i(\mathbf{x}) = P(Y_i = 1|\mathbf{x})$, $i = 1, ..., n$ and n is the number of classes. For an unseen input, the output ranking is given by arranging the classes in descending order of predicted logit. The model has $n(m+1)$ free parameters, and is therefore good only for small n and m with sufficiently large training sets.

5 Alternatives to Parallel Application and Combination

A set of classifiers can cooperate in other ways besides a simultaneous and parallel application. These include the possibilities of dynamically selecting the most appropriate classifier for each input, and multi-stage combination.

Dynamic Classifier Selection

Consider an *oracle* that always predicts the best classifier for each pattern. If such an oracle is available, we can take the decisions only from the selected classifier and ignore those by others. This is an ideal case of *dynamic classifier selection.*

The decision of such an oracle could be based on confidence of feature detection, or the correlation of classifier performance with measurable characteristics of the pattern, or estimation of the type of degradation in a pattern. Dynamic selection can be applied on a set of classifiers that are statically determined to be useful when all possible cases are considered.

One way to approximate such an oracle is to specify a set of mutually exclusive conditions that divides a training set into several partitions. Classifier performance is measured separately on each partition so that the best classifier for each partition is determined. Each test pattern will be categorized first into a partition and then classified by the corresponding best classifier.

Dynamic selection can also be applied at the decision combination level. After the training set has been partitioned by a computable condition, the significance of each classifier's contribution can be estimated using the logistic model. The estimated model gives the decision combination function for the type of inputs represented by that partition. After the model is estimated for each partition, a suitable decision combination function can be selected dynamically for each test case.

One method to do this is given as follows. Suppose that a set of independent classifiers are used. The quality of an input pattern can then be characterized by examining the class rankings produced by these classifiers. Intuitively, the classifiers tend to agree on the top choice for patterns that are easy to recognize, and tend to disagree for difficult cases. Therefore, whether the top choices are the same indicates the difficulty of recognizing a particular input. A training set can be partitioned according to the state of agreement by the classifiers on the top choices. A regression model can then be estimated separately for each partition. In test runs, a combination function can be dynamically selected according to such a state of agreement.

It is important that the selecting conditions must be cheaply computable from the inputs. The importance can be seen if this selection is viewed as an intermediate method between two extremes, one using a static single classifier and the other using one classifier for each class that responds well for patterns of only that class. In the latter case, the partitioning condition is the true class identity of the input, and hence the selection of the best classifier is equivalent to the original recognition problem in difficulty.

Multiple Stage Organization

A number of classifiers and their combination functions may be organized in many different ways. One possibility is that all the classifiers are connected in parallel, and their decisions are combined by one or several methods applied serially. Alternatively, the classifiers may be organized in groups, with a combination function applied to each group; and the combined decisions from each group are recombined later to derive a final decision. Certain classifiers may be combined using the reduction methods, so that other

classifiers can be applied to the reduced set of classes. There are even more possibilities if dynamic selection is applied.

In general, the classifiers can be organized in a multi-stage structure. At each stage, a group of classifiers operate in parallel and their decisions are combined by any one of the methods proposed here. A dynamic selector decides which classifiers are to be activated at each stage. The set of classes is then gradually reduced and reordered as it goes through each stage.

The optimal design of such a multi-stage organization is likely to be specific to a particular application. Factors to consider include the performance of each classifier, the correctness of the combined rankings given by each method, and the cost of errors.

6 Experimental Results

Machine-Printed Word Recognition

The three reranking methods were tested in a word recognition application, where the objective was to classify a word image as one of 1,365 words in a lexicon. The images were collected from machine-printed addresses taken from live mail in a post office. The words are in unrestricted font types and are of highly variable quality. The lexicon was compiled from a database of postal words and aliases.

Four classifiers were designed to rank the lexicon according to different features and matching procedures [13] [18]. Classifier 1 uses a polynomial discriminant for character recognition and confidence-based contextual postprocessing. Classifier 2 is a segmentation-based method using binary pixel values as features. Classifiers 3 and 4 are wholistic approaches that treat a word as a single symbol [16].

The four classifiers were applied to a set of 4,624 training images, and the top ten choices from the class rankings were used to estimate a logistic regression model. Besides, for each image, the top choice by each classifier was compared to the top choices of the other classifiers and an *agreement grade* was assigned accordingly.

The agreement grade is assigned by observing if there are two or more classifiers making the same top choice, and which classifiers they are. This is motivated by the observation that the classifiers tend to agree at the top choice for good quality images, and tend to disagree for bad images. Therefore the top choice agreement can be used as an indicator of image quality.

There are 12 ($C_4^4 + C_4^3 + C_4^2 + C_4^0$) possible states of agreement among the four top choices, ranging from "all four agree" to "all four disagree". Hence 12 agreement grades are defined. The training set is then partitioned into 12 subsets accordingly. A

logistic regression model is estimated for each subset. The definitions of the agreement grades, as well as the estimated model parameters are given in Table 5.

Table 5: Definitions of agreement grades and estimated model parameters

Agreement	Which Classifiers Agree (x)				Size of	Estimated Model Parameters			
Grade	poly	segb	sfv	bfv	Partition	poly	segb	sfv	bfv
1	x	x	x	x	1320	0.3584	1.5048	2.8836	1.0190
2	x	x	-	x	325	0.3955	7.0355	0.1686	0.3563
3	x	x	x	-	136	5.7066	5.7528	0.1362	-0.1444
4	x	-	x	x	454	0.2327	0.0529	5.4321	0.1187
5	-	x	x	x	712	-0.1032	1.8564	0.8661	0.8590
6	x	-	x	-	204	0.4422	0.0001	2.5731	0.0288
7	x	-	-	x	166	0.5860	0.0421	0.1122	0.4395
8	-	x	x	-	80	0.0311	3.2486	0.5132	-0.0637
9	-	x	-	x	82	0.1268	0.9901	0.1603	0.3009
10	x	x	-	-	131	2.3105	1.0479	0.0983	0.0656
11	-	-	x	x	46	0.2261	0.1683	0.3747	0.3051
12	-	-	-	-	968	0.2964	0.3184	0.2418	0.2013
all					4624	0.3230	0.2850	0.2627	0.1905

Note that the agreement grade is defined with no reference to the correctness of the top choices. Therefore it is always obtainable even for an unseen image. A regression model estimated for the particular class can then be applied to combine the rankings.

A set of 1,384 images were used to test the decision combiners as well as the model selector. Table 6 summarizes the correct rates of the classifiers and their combinations by each method. In this test each combination method is applied independently.

Table 6: Comparison of results by individual classifiers and their combinations

	% Correct in Top N Choices				
Classifier / Combination	1	2	3	5	10
1) character recognition and postprocessing (poly)	84.9	88.4	90.3	91.2	92.3
2) segmentation-based method (segb)	86.1	90.0	90.9	91.8	92.8
3) word-shape with stroke direction features (sfv)	65.2	74.5	78.5	82.4	85.5
4) word-shape with Baird features (bfv)	50.9	59.0	62.2	66.3	70.9
5) combination by the highest rank	50.9	84.7	96.2	98.6	98.9
6) combination by the Borda count	87.4	95.8	97.2	98.2	99.0
7) combination by static regression model	90.7	96.2	97.5	98.5	99.0
8) combination by dynamically selected model	93.9	97.2	97.9	98.3	99.0
9) oracle	98.1	98.8	99.0	99.1	99.3

Substantial improvements in the combined decisions are observed. Line 5 shows that the highest rank method can improve the correct rate in the top 10 choices substantially. Because of arbitrarily broken ties, this method does not give a good top choice performance. Line 6 shows that the Borda count method improves the correct rate at all ranks. Line 7 shows the improvement achieved by logistic regression. Comparing

lines 7 and 8 in Table 6, we can see that the dynamic selector can further improve the performance over the static regression model. That means i) the agreement grade is a good indicator of the input condition; and ii) dynamic selection of a combination function using this condition is effective. Line 9 shows the possible correct rate were there an oracle that could predict which classifier among the four works the best for each image.

Identifying Redundant Classifiers

Another experiment shows how logistic regression can be used to identify redundant classifiers. The chosen domain is the recognition of degraded machine-printed characters in multiple font styles. The characters are in 62 classes, including the upper and lower case alphabet and 10 numerals. Sample images were collected from live mail, scanned on a postal OCR at a resolution of 212 pixels per inch. They were then binarized and normalized to 24×24 in size. The quality of the images is highly unstable, and in many cases the defects are severe. Examples of the images in the collection are shown in Figure 3.



Figure 3: Example images in the test set.

Due to similarities in shape after size normalization, several groups of classes are merged into single classes. These include the groups {o,0,O}, {1,l,I,i,j}, {c,C}, {p,P}, {s,S}, {u,U}, {v,V}, {w,W}, {x,X}, and {z,Z}. Therefore, only 48 distinct classes are considered.

Six classifiers were applied to recognize the characters. The features and classification procedures they use are summarized in Table 7. The pixel value vector has 576 binary components and contains the normalized 24×24 input image. The Baird feature vector has 288 components obtained by convolving 32 feature templates with the image [3]. Independence among feature components is assumed in the Bayesian classifier in this system [7]. The modified nearest-neighbor classifier uses the Hamming distance, and it ranks the classes according to the distance of the closest sample of each class

to the input. The modified 2-nearest-neighbor classifier ranks the classes according to the averaged distance of two closest samples of each class to the input. A set of 19,151 samples was used to train each of the six classifiers.

Table 7: A set of classifiers for character recognition.

Classifier	*Features*	*Classification Method*
PBC	pixel values	Bayesian with independence assumption
PNC	pixel values	modified nearest-neighbor
P2N	pixel values	modified 2-nearest-neighbor
BBC	Baird features	Bayesian with independence assumption
BNC	Baird features	modified nearest-neighbor
B2N	Baird features	modified 2-nearest-neighbor

The six classifiers were applied to a set of 8,000 sample images which are distinct from those used in classifier training. A logistic regression analysis was performed using the rankings of the 48 classes given by the six classifiers.

For each of the 8,000 samples, an observation vector of the form $< Y$, R_{PBC}, R_{PNC}, R_{P2N}, R_{BBC}, R_{BNC}, $R_{B2N} >$ was obtained for each class, where Y is 1 if that class is the true class for that image and 0 otherwise, and R_C is the rank assigned to that class by classifier C for that image. The ranks are represented by a descending number, so that R_C is 48 if that class is ranked at the top by classifier C, and R_C is 1 if it is considered the least similar to the input image by C. There are 48 such vectors for each training image. To simplify the analysis, only the top 10 decisions from each classifier were considered, that is, the vectors were used in the analysis only if any of the R_C's is larger than 38.

Nine different models were attempted in the analysis. Table 8 summarizes the results of an analysis using the SAS procedure CATMOD [32]. For the pixel features, the results indicate that the decisions of both PNC and PBC are significant when only these two classifiers are used (model 1). However, when P2N is introduced (model 2), PNC becomes insignificant. Since its weight estimate becomes close to zero, it has almost no influence in promoting the rank of any class, and can therefore be ignored. The results for the Baird feature based classifiers are similar (models 3,4). Model 9 shows that when all six classifiers are used, PNC and BNC become insignificant (estimated magnitude of the parameter is small and the standard error is large) and the combination should be based on the four other classifiers.

A set of 12,000 samples were used to test the performance of these models. Table 9 shows the performance of each of the six classifiers and their combinations by the regression method with parameters given in Table 8. The parameter was set to zero if it was determined to be insignificant.

The fact that the combination (PBC,P2N) performs better than either PBC or

Table 8: Analysis of maximum likelihood estimates in logistic regression.

Model	Effect	Parameter	Estimate	Standard Error	Chi-Square	Prob
1	INTERCEPT	1	-23.7605	0.3833	3842.63	0.0000
	PNC	2	0.2951	0.00818	1302.55	0.0000
	PBC	3	0.1861	0.00625	887.27	0.0000
2	INTERCEPT	1	-25.0556	0.4121	3695.84	0.0000
	PNC	2	-0.0107	0.0215	0.25	0.6180
	PBC	3	0.1730	0.00615	790.01	0.0000
	P2N	4	0.3466	0.0238	212.39	0.0000
3	INTERCEPT	1	-24.9299	0.4149	3610.43	0.0000
	BNC	2	0.3425	0.00920	1384.96	0.0000
	BBC	3	0.1633	0.00612	712.72	0.0000
4	INTERCEPT	1	-26.9303	0.4614	3405.92	0.0000
	BNC	2	-0.0100	0.0205	0.24	0.6274
	BBC	3	0.1448	0.00593	597.07	0.0000
	B2N	4	0.4136	0.0235	308.93	0.0000
5	INTERCEPT	1	-21.8316	0.3559	3762.15	0.0000
	PBC	2	0.2277	0.00692	1082.12	0.0000
	BBC	3	0.2107	0.00684	948.43	0.0000
6	INTERCEPT	1	-25.4009	0.4204	3650.28	0.0000
	PNC	2	0.2186	0.00766	814.00	0.0000
	BNC	3	0.2974	0.00910	1068.49	0.0000
7	INTERCEPT	1	-27.5226	0.4652	3499.64	0.0000
	P2N	2	0.2275	0.00845	724.48	0.0000
	B2N	3	0.3337	0.0104	1028.18	0.0000
8	INTERCEPT	1	-26.1740	0.4099	4078.14	0.0000
	PBC	2	0.1085	0.00614	312.51	0.0000
	BBC	3	0.0836	0.00574	212.28	0.0000
	PNC	4	0.1405	0.00764	338.62	0.0000
	BNC	5	0.2076	0.00896	536.83	0.0000
9	INTERCEPT	1	-27.7886	0.4537	3751.44	0.0000
	PBC	2	0.0996	0.00608	268.56	0.0000
	BBC	3	0.0775	0.00563	189.54	0.0000
	PNC	4	-0.0579	0.0220	6.93	0.0085
	BNC	5	-0.0232	0.0212	1.20	0.2741
	P2N	6	0.2130	0.0244	76.40	0.0000
	B2N	7	0.2648	0.0243	119.12	0.0000

Table 9: Performance of character classifiers and their combinations on test set.

Model	Classifier(s)	Correct Rate (%) at Top N Choices					
		1	2	3	4	5	10
	PBC	79.3	87.8	91.2	92.8	94.3	97.6
	PNC	85.3	91.3	93.3	94.7	95.4	97.7
	P2N	85.8	91.9	93.9	94.9	95.7	97.9
	BBC	79.1	87.5	90.7	92.4	93.9	97.1
	BNC	84.3	90.8	93.3	94.6	95.4	97.7
	B2N	85.4	91.7	93.9	95.1	95.7	98.0
	Logistic Regression						
1	PBC,PNC	85.4	92.2	94.5	95.7	96.7	98.5
2	PBC,P2N	86.3	92.8	94.7	95.9	96.8	98.6
3	BBC,BNC	85.3	91.8	94.1	95.3	96.1	98.3
4	BBC,B2N	86.0	92.3	94.5	95.7	96.4	98.4
5	PBC,BBC	81.3	89.8	92.7	94.5	95.8	98.3
6	PNC,BNC	86.7	92.6	94.6	95.6	96.2	98.4
7	P2N,B2N	86.9	92.9	95.0	95.9	96.5	98.5
8	PBC,BBC,PNC,BNC	88.1	93.8	95.7	96.6	97.2	98.8
9	PBC,BBC,P2N,B2N	88.8	94.1	95.7	96.7	97.4	98.9

P2N individually indicates that even using the same feature set, different classifier designs give independent decisions that can be combined to achieve a higher performance level, though the improvement is not as remarkable as a combination of classifiers that use different feature sets. This suggests that different information contained in the feature vectors is utilized in each classification method. Such information is effectively used in a multiple classifier system.

From Table 9, we can observe improvements of the top choice correct rates in each combination over the individual classifiers. The most significant improvement is obtained by combining four classifiers (PBC, BBC, P2N, B2N), which gives a net increase of 3% in the top choice correct rate over the best individual classifier (P2N). Corresponding improvements are also observed in larger neighborhoods. In our experiments with other classifiers, improvement at top choice is usually achievable if three or more classifiers are combined. In cases where only two classifiers are combined and their errors are highly correlated, the top choice correct rate is not necessarily improved.

7 Conclusions

A multiple classifier system is suggested to solve complex pattern recognition problems. Its advantages include robustness given by simultaneous uses of complementary recognition methods, and flexibility in dynamic adaptation. Decisions are represented as rankings of a given class set. They can be combined by a number of methods that either reduce or rerank the class set. These methods are applicable regardless of the type of

similarity scores used by the individual classifiers, thereby allowing flexibility in selecting the best descriptors and similarity functions for each type of useful features for a particular problem. The effectiveness of the methods has been demonstrated in several applications with real-world data. It is expected that the methods are applicable to other problem domains as well, and that they will be most useful for recognition problems involving a large number of classes and at least several solutions. Examples of such domains include Chinese character recognition, fingerprint recognition, face recognition, and certain types of medical diagnosis.

Some other combination functions are useful in special types of multiple classifier systems. For systems where all the classifier decisions are binary, simple voting methods such as the majority vote may be satisfactory. In cases when reasonable and consistent confidence measures can be assigned to the decisions, heuristic functions or theories for confidence combination may be applicable. Other interesting alternatives for decision combination include custom-designed decision trees, neural networks as well as symbolic inference systems.

Selection of an optimal subset of classifiers, by methods other than combinatorial search and regression analysis, will also be interesting. This involves a systematic study of the correlation of errors made by the classifiers. The implications of each possible organization of a multiple classifier system are not yet clear. How to achieve an optimal organization is a challenging open problem.

Dynamic selection of classifiers is likely to be domain specific. Descriptors of input conditions are needed that can categorize the input with respect to their responses to each classifier. Such conditions may include features in the input patterns as well as characteristics of the classifier outputs.

The performance of a multiple classifier system, though it can be better than those of each individual, will reach an upper limit if there are cases where none of the classifiers' decision is sufficiently close to correct. A question then arises: is it possible to systematically create a multiple classifier system for a given problem, so that for each possible input pattern, there exists one or a combination of several classifiers that can correctly identify its true class? If this could be done, a perfect solution could be obtained for any given recognition problem. Recently, the studies by Kleinberg [22][23][24] suggest a promising approach towards this goal.

Acknowledgements The support of the Office of Advanced Technology of the United States Postal Service is gratefully acknowledged. We would like to thank Henry Baird, Saskia le Cessie and David Sher for their helpful comments.

References

[1] A. Agresti, *Categorical Data Analysis*, John Wiley & Sons, 1990.

[2] J.A. Anderson, "Regression and Ordered Categorical Variables", *Journal of the Royal Statistical Society*, **B46**, 1, 1-30, 1984.

[3] H.S. Baird, H.P. Graf, L.D. Jackel, W.E. Hubbard, "A VLSI Architecture For Binary Image Classification", in J.C. Simon, ed., *From Pixels to Features* , North Holland, 1989, 275-286.

[4] D. Black, *The Theory of Committees and Elections*, Cambridge University Press, London, 1958, 2nd ed., 1963.

[5] R. Bradford, T. Nartker, "Error Correlation in Contemporary OCR Systems", *Proceedings of the First International Conference on Document Analysis and Recognition*, Saint-Malo, France, 1991, 516-523.

[6] D.R. Cox, E.J. Snell, *Analysis of Binary Data*, 2nd. ed., Chapman and Hall, 1989.

[7] R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Addison-Wesley, New York, 1973.

[8] B. Efron, "The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis", *Journal of the American Statistical Association*, **70**, 352, 892-898, December 1975.

[9] R.M. Haralick, "The Table Look-Up Rule", *Communications in Statistics - Theory and Methods*, **A5**, 12, 1163-1191, 1976.

[10] T. Hastie, personal communication, 1992.

[11] T.P. Hettmansperger, *Statistical Inference Based on Ranks*, John Wiley & Sons, 1984.

[12] T.K. Ho, J.J. Hull, S.N. Srihari, "Combination of Structural Classifiers", *Pre-Proceedings of the IAPR Syntactic and Structural Pattern Recognition Workshop*, Murray Hill, June 1990, 123-136.

[13] T.K. Ho, *A Theory of Multiple Classifier Systems and Its Application to Visual Word Recognition*, Doctoral Dissertation, Department of Computer Science, SUNY at Buffalo, 1992.

[14] T.K. Ho, J.J. Hull, S.N. Srihari, "A Regression Approach to Combination of Decisions by Multiple Character Recognition Algorithms", *SPIE Proceedings Vol. 1661, Machine Vision Applications in Character Recognition and Industrial Inspection*, San Jose, February 10-12, 1992.

[15] T.K. Ho, J.J. Hull, S.N. Srihari, "On Multiple Classifier Systems for Pattern Recognition", *Proceedings of the 11th International Conference on Pattern Recognition*, The Hague, The Netherlands, Aug 30-Sep 3, 1992, 84-87.

[16] T.K. Ho, J.J. Hull, S.N. Srihari, "A Word Shape Analysis Approach to Lexicon Based Word Recognition", *Pattern Recognition Letters*, **13**, 821-826, 1992.

[17] T.K. Ho, J.J. Hull, S.N. Srihari, "Combination of Decisions by Multiple Classifiers", in H. Baird, H. Bunke, K. Yamamoto (eds.), *Structured Document Image Analysis*, Springer-Verlag, 1992, 188-202.

[18] T.K. Ho, J.J. Hull, S.N. Srihari, "A Computational Model for Recognition of Multifont Word Images", *Machine Vision and Applications*, **5**, 157-168, 1992.

[19] D.W. Hosmer, S. Lemeshow, *Applied Logistic Regression*, John Wiley & Sons, 1989.

[20] J.J. Hull, A. Commike, T.K. Ho, "Multiple Algorithms for Handwritten Character Recognition", *Proceedings of the First International Workshop on Frontiers in Handwriting Recognition*, Montreal, April 1990, 117-124.

[21] F. Kimura, M. Shridhar, "Handwritten Numerical Recognition Based on Multiple Algorithms", *Pattern Recognition*, **24**, 10, 969-983, 1991.

[22] E.M. Kleinberg, "Stochastic Discrimination", *Annals of Mathematics and Artificial Intelligence*, **1**, 207-239, 1990.

[23] E.M. Kleinberg, "The Theory of Stochastic Modeling in Pattern Recognition", in preparation.

[24] E.M. Kleinberg, T.K. Ho, "Pattern Recognition by Stochastic Modeling", *Proceedings of the Third International Workshop on Frontiers in Handwriting Recognition*, Buffalo, May 1993.

[25] J.M. Landwehr, D. Pregibon, A.C. Shoemaker, "Graphical Methods for Assessing Logistic Regression Models", *Journal of the American Statistical Association*, **79**, 385, 61-71, March 1984.

[26] S. le Cessie, *Model Building Techniques for Logistic Regression, with Applications to Medical Data*, Doctoral Dissertation, University of Leiden, The Netherlands, 1991.

[27] E. Mandler, J. Schuermann, "Combining the Classification Results of Independent Classifiers Based on the Dempster/Shafer Theory of Evidence", in E.S. Gelsema and L.N. Kanal, eds., *Pattern Recognition and Artificial Intelligence*, North Holland, 1988, 381-393.

[28] V.D. Mazurov, A.I. Krivonogov, V.L. Kazantsev, "Solving of Optimization and Identification Problems by the Committee Methods", *Pattern Recognition*, **20**, 4, 371-378, 1987.

[29] R. Meddis, *Statistics Using Ranks, A Unified Approach*, Basil Blackwell, 1984.

[30] C. Nadal, R. Legault, C.Y. Suen, "Complementary Algorithms for the Recognition of Totally Unconstrained Handwritten Numerals", *Proceedings of Tenth International Conference on Pattern Recognition*, Atlantic City, 1990, 443-449.

[31] S.J. Press, S. Wilson, "Choosing Between Logistic Regression and Discriminant Analysis", *Journal of the American Statistical Association*, **73**, 364, 699-705, December 1978.

[32] SAS Institute Inc., *SAS/STAT User's Guide*, Version 6, Fourth Edition, Volume 2, SAS Institute Inc., 1989.

[33] K-D. Wernecke, "A Coupling Procedure for the Discrimination of Mixed Data", *Biometrics*, **48**, 497-506, June 1992.

[34] L. Xu, A. Krzyzak, C.Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition", *IEEE Transactions on Systems, Man, and Cybernetics*, **22**, 3, 418-435, May/June 1992.