On Multiple Classifier Systems for Pattern Recognition

Tin Kam Ho, Jonathan J. Hull, Sargur N. Srihari

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

Abstract

Difficult pattern recognition problems involving large class sets and noisy input can be solved by a multiple classifier system, which allows simultaneous use of arbitrary feature descriptors and classification procedures. Independent decisions by each classifier can be combined by methods of the highest rank, Borda count, and logistic regression, resulting in substantial improvement in overall correctness.

1 Introduction

A pattern recognition system is proposed that consists of a set of independent classifiers and a decision combination function [6]. We consider this as a preferred solution to a complex pattern recognition problem, because it allows simultaneous use of feature descriptors of many types, their corresponding measures of similarity, and any appropriate classification procedures. Classifiers using different feature descriptors and matching procedures often produce uncorrelated errors. Classification performance can be improved by obtaining a consensus of their decisions. It is also possible to dynamically select the most appropriate classifiers for inputs of a particular type.

An effective decision combination function is critical to the success of a multiple classifier system. The function should take advantage of the strengths of the individual classifiers, avoid their weaknesses, and improve classification correctness. Methods based on intersection and union of candidate sets have been proposed in [4]. Those techniques are sensitive to outlying worst cases and hence are unsatisfactory in a noisy environment. In this paper, we propose several new techniques to overcome this problem.

2 Decision Combination Methods

We assume that each classifier outputs its decisions as a ranking of a given set of classes with respect to an input pattern. A classifier believes most strongly that the input pattern belongs to the class ranked at the top, according to a certain similarity score it computes. Binary decisions (accepted as a particular class or rejected) output by some classifiers are considered as rankings of two levels.

Rankings are used for three reasons. (1) They contain more information than a unique class choice. This is desirable in a many-class discrimination problem, where the classifiers may not be able to uniquely identify the true class, but can include the true class in several guesses with high accuracy. (2) As a nonparametric method, it avoids making assumptions on the distribution of the numerical similarity scores computed by each classifier. (3) It is convenient to implement, and generally applicable to all types of classifiers, since a ranking can always be derived from a linear ordering of the similarity scores.

Ranking combination methods are intended to improve the rank of the true class. Their success is evaluated by the position of the true class in the resultant ranking. A combination method is considered successful if the true class appears close to the top of the output ranking more often than it does in any of the individual rankings that are input.

Three methods for ranking combination are proposed. In each method, a confidence score is defined which is a function of the ranks assigned to each class by the classifiers. Only the relative magnitudes of the confidence scores for different classes are important. The consensus ranking is obtained by sorting the classes by the confidence scores.

Let the ranks assigned to a class by classifiers $C_1, C_2, ..., C_m$ $(m \geq 2)$ be represented as a vector $R = < r_1, r_2, ..., r_m >$ associated with each class. $r_i = 1$ if a class is ranked at the top by classifier C_i, and $r_i = 2$ if ranked at the second position and so on. The confidence score S is a function of R:

$$S = f(R).$$

Three forms of $f(R)$ are proposed:

$$\begin{aligned} f_1(R) &= \min(r_1, r_2, ..., r_m) \\ f_2(R) &= r_1 + r_2 + ... + r_m \\ f_3(R) &= w_1 r_1 + w_2 r_2 + ... + w_m r_m. \end{aligned}$$

A method that uses f_1 is called the *highest rank* method. f_2 is used in the *Borda count* method, and f_3 is used in a generalized Borda count method where the weights w_i are estimated by *logistic regression*. Using each of f_1, f_2, and f_3, a new ranking is obtained by sorting the given classes according to the computed values of f_1, f_2, and f_3, respectively. The advantages and disadvantages of each method are discussed in Sections 3 to 5.

3 The Highest Rank Method

Assume that for each input pattern, M classifiers are applied to rank a set of classes. Each class is then assigned the minimum (highest) of the M rank positions it receives as its score. The set of classes is then sorted by these scores to yield a combined class ranking for the input pattern. There may be ties in this combined ranking, which may be broken arbitrarily to achieve a strict ordering.

This method is particularly useful in a problem involving a large number of classes and a small number of uncorrelated classifiers. The advantage is with its ability to focus on 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 N, no matter how the other classifiers perform, the true class will be at a position no farther then $N \times M$ from the top in the combined ranking, where M is the number of classifiers.

One disadvantage with this method is that the combined ranking may have many ties. The number of classes in ties depends on the number of classifiers used. For example, if five classifiers are used, there are at least one and at most five distinct classes that are ranked at the top by a classifier. Those five classes are in a tie in the final ranking, since they all receive the score 1. Therefore, this method is useful only if the number of classifiers is small relative to the number of classes.

4 The Borda Count Method

Decision combination in a multiple classifier system can be viewed as a committee voting problem in group decision theory and accomplished by a *group consensus function*. One useful group consensus function is the *Borda count* [2]. For any particular class c, the Borda count is the sum of the number of classes ranked below c by each classifier. It is defined as follows:

For any class c in a set S, let $B_j(c)$ be the number of classes in S which are ranked below the class c by classifier C_j $(j = 1, ..., m)$. The Borda count for class c is

$$B(c) = \sum_{j=1}^{m} B_j(c).$$

The consensus ranking is given by sorting the classes in descending order of their Borda counts.

Intuitively, if the class c is ranked near the top by more classifiers, its Borda count tends to be larger. The magnitude of the Borda count for each class measures the strength of agreement by the classifiers on the proposition that the input pattern belongs to that class. In a two-class problem, the Borda count is equivalent to a majority vote.

If there are no ties in the individual rankings, the Borda count method is equivalent to sorting the classes by the sum of their ranks (f_2 in Section 2). The Borda count method assumes an additive independence between the rankings.

The Borda count method is simple to implement, and requires no *a priori* knowledge of the classifier performance. 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 it is known for sure that certain classifiers perform better than others in most cases. Because of this, the method does not guarantee an improvement of the rank of the true class. A statistical approach that computes a weight for each classifier is described in the next section.

5 The Logistic Regression Method

A generalization of the Borda count is a weighted sum of the ranks. The weights reflect the relative significance of each classifier, which are estimated based on observations of the classifier performance on a training set. The weights can be estimated by a logistic regression analysis [1][3][7].

Consider the ranks assigned to each class as random variables. For a set of training patterns, the true class identities are known. To represent such knowledge from the training set, we need to determine how the rank variables predict which class is the true class of a given pattern.

We define a binary response variable Y associated with each class, with $Y = 1$ meaning that the class is the true class, and $Y = 0$ otherwise. For a set of training patterns, the true classes are known and

therefore each class has a known value of Y. Y has a Bernoulli distribution with expected value

$$E(Y) \;=\; 1 \times P(Y=1) + 0 \times P(Y=0) = P(Y=1).$$

We denote the probability $P(Y=1)$ as $\pi(\mathbf{x})$, where $\mathbf{x} \;=\; < x_1, x_2, ..., x_m >$ represents the ranks assigned to that class by classifiers $C_1, C_2, ..., C_m$. For convenience in interpretation, we assume that x_j for a particular class has the largest value when that class is ranked at the top by classifier C_j. For instance, in a problem with 100 classes, $x_3 \;=\; 100$ for a class ranked by C_3 at the top, and $x_3 \;=\; 99$ for a class ranked next to it, and so on.

Intuitively we expect that the likelihood of a class being the true class $P(Y \;=\; 1)$ is larger when it is ranked higher by the classifiers. The relationship between the ranks and the tendency towards being a true class is expected to be monotonic, but not necessarily linear. In fact, we expect $\pi(\mathbf{x}) \rightarrow 0$ when component values of $\mathbf{x}$ are small, and $\pi(\mathbf{x}) \rightarrow 1$ when component values of $\mathbf{x}$ are large. These conditions on the response function suggest the use of the logistic response function [1]. For the case with one independent variable, the function has the form

$$\pi(x) \;=\; \frac{\exp\;(\alpha \;+\; \beta x)}{1 \;+\; \exp\;(\alpha \;+\; \beta x)}$$

where x is the independent variable, α, β are parameters. Using this function as a model, the odds of having response 1 ($P(Y=1)$) are

$$\frac{\pi(x)}{1 \;-\; \pi(x)} \;=\; \exp\;(\alpha \;+\; \beta x).$$

The log-odds have a linear relationship with the independent variable x

$$\log \frac{\pi(x)}{1 \;-\; \pi(x)} \;=\; (\alpha \;+\; \beta x),$$

which can be generalized to the multivariate case with m independent variables

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

Such a log-odds (*logit*) transformation links the problem to linear regression analysis [1]. Maximum likelihood methods can be used to estimate the model parameters. The relative magnitudes of the estimated parameters indicate the relative significance of the classifiers in their marginal contributions to the logit. Hence insignificant classifiers may be identified and removed from the system.

Estimation of the weights is illustrated with an example application in word recognition, where two classifiers are used to discriminate between 33,850 classes (words). 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. Only the top ten decisions from each of the two rankings were used in model estimation. That is, a class receives a 10 if it is ranked at the top, and a 0 if it is ranked below the 10th position. Figure 1 shows the plot of the computed logits versus x_1 and x_2. A regression plane was fit to these logits by the SAS procedure LOGISTIC [9]. The estimated model is

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

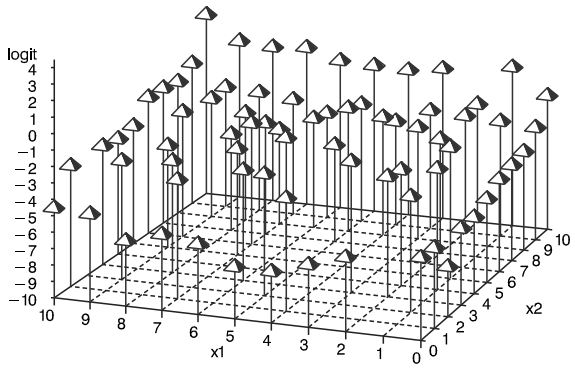


Figure 1: Plot of empirical logits versus ranks by two classifiers.

For an unseen pattern, the true class is unknown, and the logit of each class is predicted by the estimated model. Since the odds of having response 1 (i.e., being the true class) increase monotonically with the logit, the given classes can be ordered by descending values of the logit. The class with the largest logit is then considered as most likely to be the true class.

In parameter estimation, all combinations of the ranks have to be considered. Therefore the method is computationally expensive, and is practical only for rankings of a small number of classes. In problems involving many classes, one may consider only those classes ranked very close to the top, say, only the top ten decisions in each ranking.

By using a specific Y for each decided class, and using ranks for all classes simultaneously as regressors [8], a regression model can be estimated for each decided class. Decision for an unseen pattern can then be made by comparing the predicted logit for each class. This is a more refined model, which is useful when there is a large amount of training data.

6 Incremental Refinement

In applications where only a few top choices are desired, the logistic regression method can be applied to obtain a consensus. However, in an application where a complete ranking of a large number of given classes is desired, estimation of the model parameters may become too expensive. In such a case, the three combination functions can be used in turn to improve the rankings incrementally. The highest rank method can be applied first to obtain a combined ranking. A number of choices can then be extracted, which can be reranked using the Borda count method or the logistic regression method. This is a process of incremental refinement, starting with the simplest method and finishing with the most expensive method.

7 Dynamic Classifier Selection

Classification performance of a multiple classifier system may also be improved by dynamically selecting the most appropriate classifiers for inputs of a particular type.

Consider an *oracle* that will always select the best classifier for each image. If such an oracle is available, we can take the decisions from the selected classifier, and ignore the decisions by other classifiers. This is an ideal case where we can apply *dynamic classifier selection.*

One way to approximate such an oracle is to compute some mutually exclusive conditions that partition the set of all possible patterns in a particular domain. A training set is partitioned according to the computed conditions. Classifier performance is measured separately on each partition. The best classifier for each partition is hence determined. For the test set, similar conditions are computed and the best classifier for the corresponding partition is selected. This is a divide-and-conquer strategy.

Dynamic selection can also be applied at the classifier set level. After the training set has been partitioned, the significance of each classifier's contribution can be evaluated using logistic regression analysis. The estimated model suggests the decision combination function for the type of patterns in that partition. After the model is estimated for each partition, the appropriate decision combination function can be selected dynamically for each unseen pattern by evaluating the partitioning condition.

Possible partitioning conditions include characterizations of input quality and agreement of decided classes. For example, a character image can be described by the density of black pixels and a measure of image fragmentation. The agreement of, say, the top choice by each classifier, often indicates the difficulty of a particular instance of a given problem. Classifiers tend to disagree with one another on a difficult input. Therefore some hints on how the decisions should be combined can be obtained by observing the top choice agreement by particular classifiers.

8 Experimental Results

The three combination methods were tested in a word recognition application, where the objective was to classify a word image as one of 33,850 words in a given lexicon. The images were collected from machine-printed addresses taken from live mail, scanned at 212 pixels per inch and binarized. The words had been printed in highly variable qualities and fonts. The lexicon was compiled from a database of postal words and aliases.

Five classifiers were designed to rank the lexicon according to different features and matching procedures [5]. Classifiers 1 and 2 use two different techniques to postprocess decisions based on isolated character recognition. Classifier 3 is based on character segmentation and character recognition in context. Classifiers 4 and 5 are based on word shape analysis that treats a word as a single symbol. The classifiers and the combination methods were trained using 1,055 sample images.

Table 1 summarizes the performance of the five classifiers and their combinations by the three methods over a set of 1,671 testing images, without dynamic classifier selection. Substantial improvements in the correct rates are observed.

9 Conclusions

A multiple classifier system was proposed to solve complex pattern recognition problems. Its advantages include robustness given by simultaneous use of complementary recognition methods and flexibility in dynamic adaptation. Recognition decisions are represented as rankings of a given class set. Consensus in decisions is obtained by three different methods, whose effectiveness is demonstrated in a word recognition application.

Table 1: Summary of performance on 1,671 degraded word images using a 33,850 word lexicon.

Descriptions	% Correct in Top N Decisions N = 1	2	3	10
Classifier 1	79.2	86.1	88.2	90.5
Classifier 2	76.9	83.2	85.4	88.3
Classifier 3	74.8	84.1	86.3	90.5
Classifier 4	42.4	53.7	59.5	72.1
Classifier 5	58.9	70.0	74.5	82.9
Combinations by				
highest rank	46.6	73.8	87.0	94.9 *
Borda count	83.1 *	88.1	90.7	94.6
logistic regression	88.4 *	91.2	92.7	95.1

* : significant improvements over individual performance.

The proposed methodology is applicable to any pattern recognition problem, as long as multiple solutions exist for that problem, and each of them provides a ranking of a given set of classes with respect to an input pattern.

The advantage of using multiple classifiers is best demonstrated in problems that involve a large set of classes and complex inputs. Word recognition is one such problem, where the size of the lexicon determines the number of classes. Other problems with these characteristics include the recognition of Chinese characters, fingerprints, human faces, speech, as well as certain areas of medical diagnosis.

References

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

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

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

[4] 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, 1990, 123-136.

[5] T.K. Ho, J.J. Hull, S.N. Srihari, "Word Recognition with Multi-Level Contextual Knowledge", *Proceedings of the First International Conference on Document Analysis and Recognition*, Saint-Malo, 1991, 905-915.

[6] T.K. Ho, *A Theory of Multiple Classifier Systems And Its Application to Visual Word Recognition*, Doctoral Dissertation, Department of Computer Science, State University of New York at Buffalo, 1992.

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

[8] T. Pavlidis, T. Hastie, personal communications.

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