# A Comparison Study of Some Combined Classifiers

By: Majid Mojirsheibani, School of Mathematics & Statistics, Carleton University, Ottawa, Ontario, K1S 5B6, Canada.

## Abstract

In this article we propose two procedures for combining a number of individual classifiers in order to construct more effective classification rules. The effectiveness of the new procedures, as compared to those of the existing methods, is assessed via detailed simulation studies.

Keywords: classification, Bayes error, combined classifier.

## 1. Introduction

In this article we consider the following standard multigroup classification problem. Let

$$\mathbf{T}_n = \{(\mathbf{X}_1, Y_1), \cdots, (\mathbf{X}_n, Y_n)\}$$

be a training sample of size $n$, where $\mathbf{X}_i \in \Re^d$ is called the feature or predictor vector and $Y_i \in \{1, \cdots, K\}$ is the class membership associated with $\mathbf{X}_i$. Here we assume that $(\mathbf{X}_i, Y_i)$'s are iid random vectors with a common distribution $\mathbf{F}_{\mathbf{X},\mathrm{Y}}$. Let $(\mathbf{X}, Y)$ be a new observation from $\mathbf{F}_{\mathbf{X},\mathrm{Y}}$ whose class membership $Y$ is to be predicted. When the parent distribution $\mathbf{F}_{\mathbf{X},\mathrm{Y}}$ is completely known, one would predict $Y$ by finding a classifier $g$ (i.e., a map from $\Re^d$ to $\{1, \cdots, K\}$) for which the misclassification error $\text{err}(g)$,

$$\text{err}(g) = P\{g(\mathbf{X}) \neq Y\},$$

is as small as possible. Let $g_{\text{B}}$ be defined by:

$$P\{g_{\text{B}}(\mathbf{X}) \neq Y\} = \inf_{g:\ \Re^d \to \{1, \cdots, K\}} P\{g(\mathbf{X}) \neq Y\},$$

i.e., $g_{\text{B}}$ attains the smallest misclassification error rate. Then $g_{\text{B}}$ is called the Bayes classifier. Unfortunately, in practice, $\mathbf{F}_{\mathbf{X},\text{Y}}$ is unknown and therefore $g_{\text{B}}$ cannot be found. The goal is then to find a data-based classification rule $g_n$, based on the training sample $\mathbf{T}_n$, whose conditional misclassification error

$$\text{err}_n(g_n) = P\{g_n(\mathbf{X}) \neq Y \mid \mathbf{T}_n\}$$

is somehow as small as possible. If $g_n$ satisfies the condition

$$\text{err}_n(g_n) \longrightarrow_p \text{err}(g_{\text{B}}), \quad \text{as} \ \ n \to \infty,$$

then it is said to be Bayes consistent. When the above convergence holds w.p.1, then $g_n$ is said to be strongly consistent.

In practice it is not clear as to how one should choose a classifier; this is particularly true in nonparametric situations. Of course, it is true that a uniformly consistent classification rule $g_n$ (i.e., $g_n$ satisfies $\sup_{\mathbf{F}_{\mathbf{X},\text{Y}}} \text{err}_n(g_n) \to_p \text{err}(g_{\text{B}})$) tends to perform reasonably well in most cases; however, given a few different such consistent rules, it is not clear at all as to how to choose the "best" one. An even more realistic difficulty in choosing a classifier deals with the fact that different classifiers have different merits and, as a result, in a given situation, one classifier can perform better than another one. More specifically consider the following typical situation. Suppose that there are 3 classes, two of which are approximately multivariate normal distributions, and the third class is non-normal. Then Fisher's linear or quadratic discriminant analysis might work best for separating the first two classes (the normal distributions), while a k-Nearest Neighbor rule is perhaps more appropriate in the non-normal case. This example suggests that perhaps one should consider methods that somehow combine (implicitly) the best features of different individual classifiers.

In this article we advocate methods for combining different classifiers in order to develop more effective classification rules. Here, more effectiveness means (at least in an asymptotic sense) higher predictive power or, equivalently, lower misclassification error rate. In the next section we consider a number of procedures for combining different classifiers. Some relevant work are those of Breiman (1995, 1997), Mojirsheibani (1997, 1998a,b), LeBlanc and Tibshirani (1996), and Wolpert (1995).

## 2. Combining Classifiers

Suppose that we have $M$ individual data-based classifiers $g_{n,1}, \cdots, g_{n,\mathrm{M}}$. Observe that each $g_{n,j}, \quad j = 1, \cdots, M$ is a map of the form:

$$g_{n,j} : \ \{\Re^d \times \{1, \cdots, K\}\}^n \times \Re^d \longrightarrow \{1, \cdots, K\}.$$

The idea of combining $g_{n,1}, \cdots, g_{n,\mathrm{M}}$ is to find a new classifier $\psi_n(g_{n,1}, \cdots, g_{n,\mathrm{M}})$ for which the misclassification error $\mathrm{err}_n(\psi_n) = P\{\psi_n(g_{n,1}(\mathbf{X}), \cdots, g_{n,\mathrm{M}}(\mathbf{X})) \neq Y \mid T_n\}$ is, in some sense, smaller than that of each constituent classifier. In a recent article Mojirsheibani (1998a) proposed a discretization method for combining classifiers. The resulting rule, denoted by $\psi_n^{\mathrm{comb1}}$, works as follows. Let $g_{n,1}, \cdots, g_{n,\mathrm{M}}$ be as before. Then

$$\psi_n^{\mathrm{comb1}}(g_{n,1}(\mathbf{X}), \cdots, g_{n,\mathrm{M}}(\mathbf{X}) = \mathrm{argmax}_{1\leq k\leq K} \sum_{i:\ Y_i=k} \mathbf{I}\{A_{n,\mathrm{M}}(\mathbf{X}, \mathbf{X}_i)\}, \tag{1}$$

where

$$A_{n,\mathrm{M}}(\mathbf{X}, \mathbf{X}_i) = \{g_{n,1}(\mathbf{X}_i) = g_{n,1}(\mathbf{X}), \cdots, g_{n,\mathrm{M}}(\mathbf{X}_i) = g_{n,\mathrm{M}}(\mathbf{X})\},$$

and $\mathbf{I}\{A_{n,\mathrm{M}}(\mathbf{X}, \mathbf{X}_i)\}$ is the indicator function of the set $A_{n,\mathrm{M}}(\mathbf{X}, \mathbf{X}_i)$. The combined classifier (1) is motivated by the fact that when the individual classifiers $g_1, \cdots, g_M$ are nonrandom (do not depend on the data), then (1) is simply a multinomial discrimination rule based on the discretized "data": $(\mathbf{W}_1, Y_1), \cdots, (\mathbf{W}_n, Y_n)$, where $\mathbf{W}_i = (g_1(\mathbf{X}_i), \cdots, g_M(\mathbf{X}_i))$. Of course, in practice the individual classifiers depend on the data (and, as a result, $\mathbf{W}_i$'s are no longer iid). Observe that (1) may be viewed as a class-majority vote where the voters are the individual observations falling in different classes and the vote associated with $\mathbf{X}_i$ is either 0 or 1 according to whether the indicator function

$$\mathbf{I}\{g_{n,1}(\mathbf{X}_i) = g_{n,1}(\mathbf{X}), \cdots, g_{n,M}(\mathbf{X}_i) = g_{n,M}(\mathbf{X})\}$$

is 0 or 1. Note that the combining procedure given by (1) is not smooth. It lacks smoothness in the sense that the vote associated with each $\mathbf{X}_i$ is either 0 or 1 with no intermediate values. Although the combined classifier $\psi_n^{\mathrm{comb1}}$ has strong asymptotic optimality properties (see Mojirsheibani (1998a)), the lack of smoothness described above can distort its performance in the case of sparse data. To deal with the non-smoothness problem, Mojirsheibani(1998b) proposed a flexible class of kernel-based combined classifiers, where the optimal member of the class is empirically chosen by a data-splitting approach. The steps may be summarized as follows. Start by randomly splitting the data into a training sample of size $n_1$, and a testing sequence of size $n_2$; here $n_1 + n_2 = n$. Let $g_{n_1,1}, \cdots, g_{n_1,\mathrm{M}}$ be $M$ individual classifier based, only, on the training set of size $n_1$. For fix $\alpha > 0$, let $\mathbf{K}_h(x)$ be the exponential kernel:

$$\mathbf{K}_h(x) = e^{-(x/h)^{\alpha}}, h > 0$$

Also, let

$\psi_{n_1,h}(g_{n_1,1}(\mathbf{X}), \cdots, g_{n_1,\mathrm{M}}(\mathbf{X})) =$

$$\mathrm{argmax}_{1\le k\le K} \sum_{\{i:\ 1\le i\le n_1,\ Y_i=k\}} \mathbf{K}_h\left(\sum_{j=1}^{M} \mathbf{I}\{g_{_{n_1,j}}(\mathbf{X}_i) \neq g_{_{n_1,j}}(\mathbf{X})\}\right), \quad \text{if } h > 0,$$

and

$\psi_{n_1,h}(g_{n_1,1}(\mathbf{X}), \cdots, g_{n_1,\mathrm{M}}(\mathbf{X})) =$

$$\mathrm{argmax}_{1\le k\le K} \sum_{\{i:\ 1\le i\le n_1,\ Y_i=k\}} \mathbf{I}\{A_{n,\mathrm{M}}(\mathbf{X}, \mathbf{X}_i)\}, \quad \text{if } h = 0,$$

where $\mathbf{I}\{A_{n,\mathrm{M}}(\mathbf{X}, \mathbf{X}_i)\}$ appears in (1). Now, consider the class of combined classifiers

$$\Psi_{n_1,h} = \{\psi_{n_1,h} \mid h \ge 0\}.$$

The proposed kernel-based combined classifier $\psi_n^{\text{comb2}}$ is the classifier, selected from $\Psi_{n_1,h}$, that minimizes the error committed on the testing sequence, i.e., $\psi_n^{\text{comb2}}$ minimizes

$$\widehat{\text{err}}_n(\psi_{n_1,h}) = \frac{1}{n^2}\sum_{i=1}^{n_2}\mathbf{I}\{\psi_{n_1,h}(\mathbf{X}_{n_1+i}) \neq Y_{n_1+i}\}. \tag{2}$$

Of course, in practice, choosing $\psi_n^{\text{comb2}}$ from the class $\Psi_{n_1,h}$ amounts to a search over a grid of values of $h$ in order to find the one for which (2) is minimized. Note that the subscript $n$ on $\psi_n^{\text{comb2}}$ signifies the fact that $\psi_n^{\text{comb2}}$ depends on the entire sample of size $n = n_1 + n_2$. How good are the combined rules $\psi_n^{\text{comb1}}$ and $\psi_n^{\text{comb2}}$ as compared to the individual classifiers? It truns out that, under some regularity conditions, both of these rules are asymptotically, (strongly) at least as good as the best individual classifier, i.e., for each constituent classifier $g_{n,j}$, one has

$$\limsup_{n\to\infty}\{\text{err}_n(\psi_n^\ell) - \text{err}_n(g_{n,j})\} \leq_{\text{a.s.}} 0, \quad \text{for} \ \ \ell = 1,\ 2.$$

Here

$$\text{err}_n(\psi_n^\ell) = P\{\psi_n^\ell(g_{n,1}(\mathbf{X}), \cdots, g_{n,\text{M}}(\mathbf{X})) \neq Y \mid T_n\}$$

and

$$\text{err}_n(g_{n,j}) = P\{g_{n,j}(\mathbf{X}) \neq Y \mid T_n\}.$$

For more on these results see Mojirsheibani (1998a,b).

In this article we propose two new procedures for combining classifiers. To motivate our first method, recall that we may think of (1) as a class-majority vote where the vote associated with $\mathbf{X}_i$ is 1 if

$$\mathbf{I}\{g_{n,1}(\mathbf{X}_i) = g_{n,1}(\mathbf{X}), \cdots, g_{n,M}(\mathbf{X}_i) = g_{n,M}(\mathbf{X})\} = 1,$$

or equivalently if

$$\mathbf{I}\{\sum_{j=1}^{M} \mathbf{I}\{g_{n,j}(\mathbf{X}_i) = g_{n,j}(\mathbf{X})\} = M\} = 1. \tag{3}$$

This condition is perhaps too restrictive since all the $M$ classifiers are required to classify both $\mathbf{X}_i$ and $\mathbf{X}$ as belonging to the same class. Thus, quite often, the vote associate with $\mathbf{X}_i$ may turn out to be zero because $g_{n,j}(\mathbf{X}_i) \neq g_{n,j}(\mathbf{X})$ for only one or two individual classifiers; this can reduce the effectiveness of the combined classifier $\psi_n^{\text{comb1}}$ in small samples. Our first (new) combined classifier of this article, which may be viewed as an improved version of $\psi_n^{\text{comb1}}$, is denoted by $\psi_n^{\text{comb3}}$ and works as follows: Let $L_M$ be a positive integer satisfying $L_M \leq M$, then

$$\psi_n^{\text{comb3}}(g_{n,1}(\mathbf{X}), \cdots, g_{n,M}(\mathbf{X})) =$$

$$\text{argmax}_{1\leq k\leq K} \ \textstyle\sum_{i:\ Y_i=k} \mathbf{I}\{\sum_{j=1}^{M} \mathbf{I}\{g_{n,j}(\mathbf{X}_i) = g_{n,j}(\mathbf{X})\} \geq L_M\}.$$

In other words, $\psi_n^{\text{comb3}}$ replaces (3) by a more flexible criterion: $\mathbf{X}_i$ favors a vote of 1 if

$$\mathbf{I}\{\sum_{j=1}^{M} \mathbf{I}\{g_{n,j}(\mathbf{X}_i) = g_{n,j}(\mathbf{X})\} \geq L_M\} = 1.$$

Here $L_M$ may be viewed as the tuning parameter of the combined classifier $\psi_n^{\text{comb3}}$. Note that when $L_M = M$, then $\psi_n^{\text{comb3}}$ is the same as $\psi_n^{\text{comb1}}$. Before studying the effectiveness of $\psi_n^{\text{comb3}}$, we describe the mechanics of our second combining method.

Our second proposed combining procedure of this article is based on the following framework. Put

$$\psi_k(g_{n,1}(\mathbf{X}),\cdots,g_{n,M}(\mathbf{X})) = \sum_{j=1}^{M} W_{jk}\ \mathbf{I}\{g_{n,j}(\mathbf{X}) = k\},$$

for suitable weights $W_{jk}$'s, and consider the combined classifier that works by classifying $\mathbf{X}$ as belonging to, say, class $k' \in \{1,\cdots,K\}$ if

$$k' = \text{argmax}_{1\leq k\leq K}\ \ \psi_k(g_{n,1}(\mathbf{X}),\cdots,g_{n,M}(\mathbf{X})).$$

The question is how to choose the weights $W_{jk}$'s. One sensible criterion is: $W_{jk}$ should be small if the $j$-th individual classifier is poor. A natural choice would then be

$$W_{jk} = P(g_{n,j}(\mathbf{X}) = k \mid Y = k),$$

that is, the conditional probability that $g_{n,j}$ classifies $\mathbf{X}$ correctly, given $Y = k$. This suggests replacing $W_{jk}$ with the estimates:

$$\widehat{W}_{jk} = \frac{\sum_{i=1}^{n}\ \mathbf{I}\{g_{n,j}(\mathbf{X}_i) = k, Y_i = k\}}{\sum_{i=1}^{n}\ \mathbf{I}\{Y_i = k\}}$$

Thus we have the combined classifier $\psi_n^{\text{comb4}}$:

$$\psi_n^{\text{comb4}}(g_{n,1}(\mathbf{X}),\cdots,g_{n,M}(\mathbf{X})) = \text{argmax}_{1\leq k\leq K}\sum_{j=1}^{M}\ \widehat{W}_{jk}\mathbf{I}\{g_{n,j}(\mathbf{X}) = k\}.$$

## 3. Examples

The following examples provide an empirical comparison of the combining procedures discussed in this article. These examples also show some advantages of working with combined classifiers as compared to the individual ones.

**Example 1.** This example deals with $K$=3 classes with five predictors: Two multivariate normals with mean vector (1, 1, 1, 1, 1) and (2, 1, 1, 1, 1), and the identity covariance matrices; the third class is a multivariate Cauchy distribution with independent components having location and scale parameters (1,1), (2,1), (1,1), (1,1), (1,1). It should be emphasized that we have deliberately selected 3 classes which are quite difficult to separate (regrdless of the type of the individual classifier used) and that one should expect large misclassification error rates. The training sample consists of 70 observations from each of the three classes, thus the entire data size is $n = 3 \times 70 = 210$. We consider combining $M$=5 classifiers: Two tree classifiers with 6 and 12 terminal nodes, a 5-NN classifier (short for 5-Nearest Neighbor), a 40-NN classifier, and the LDA (short for linear discriminant analysis). An additional 100 observations from each class was used to estimate the misclassification error rates of different classifiers. The results appearing in row A of Table 1 are the averages over 20 such Monte Carlo runs. The numbers shown in brackets are the standard errors. Here (and in the subsequent examples) we have used the combined classifiers $\psi_n^{\text{comb1}}$, $\psi_n^{\text{comb2}}$, $\psi_n^{\text{comb3}}$ with $L_M$=3 and 4, and $\psi_n^{\text{comb4}}$; these are referred to as comb1, comb2, comb$3_3$, comb$3_4$, and comb4 respectively in Table 1. Row A of Table 1 shows that $\psi_n^{\text{comb1}}$ is nearly as good as the best individual classifier, the 5-NN rule in this case. However, the ability of the combining procedure to outperform the individual classifiers is even more apparent in the case of $\psi_n^{\text{comb2}}$ and $\psi_n^{\text{comb3}}$; these appear in columns 3, 4, and 5 of Table 1. A side-by-side boxplot of the error rates is given in Figure 1(a). Next, the entire process was repeated for training samples of sizes 200 and 400 from each class.

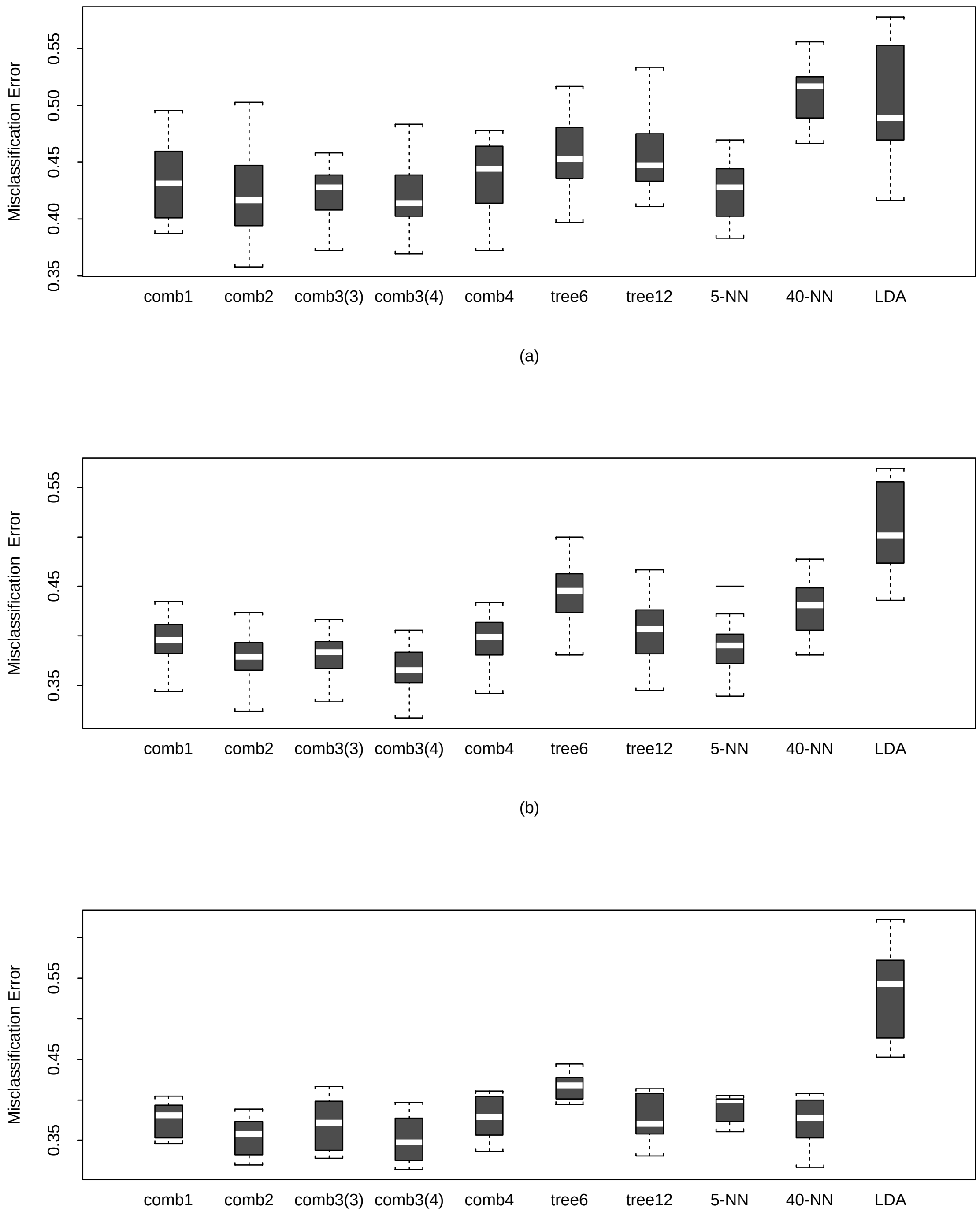
The corresponding results are shown in rows B and C of Table 1, and the side-by-side boxplots are in Figure 1(b), (c).

Table 1: Error rates for Example 1

| | comb1 | comb2 | $\text{comb3}_3$ | $\text{comb3}_4$ | comb4 | tree6 | tree12 | 5-NN | 40-NN | LDA |
|---|---|---|---|---|---|---|---|---|---|---|
| A | .430<br>(.033) | .421<br>(.041) | .424<br>(.024) | .418<br>(.030) | .436<br>(.033) | .456<br>(.032) | .451<br>(.031) | .426<br>(.026) | .509<br>(.024) | .504<br>(.051) |
| B | .393<br>(.024) | .377<br>(.026) | .381<br>(.022) | .367<br>(.025) | .397<br>(.026) | .445<br>(.029) | .405<br>(.031) | .389<br>(.028) | .429<br>(.027) | .507<br>(.044) |
| C | .375<br>(.022) | .354<br>(.025) | .370<br>(.033) | .351<br>(.031) | .378<br>(.029) | .417<br>(.018) | .377<br>(.030) | .389<br>(.017) | .373<br>(.033) | .532<br>(.060) |

Note that, except for the LDA classifiers, all individual and the combined classifiers tend to perform better for large sample sizes. The LDA is clearly not suitable due to the presence of the Cauchy class. Observe that the combined classifiers comb2 and $\text{comb3}_4$ have consistently performed quite well, regardless of the training sample size.

Figure 1: Side-by-side boxplots of the error rates for Example 1



(a)

(b)

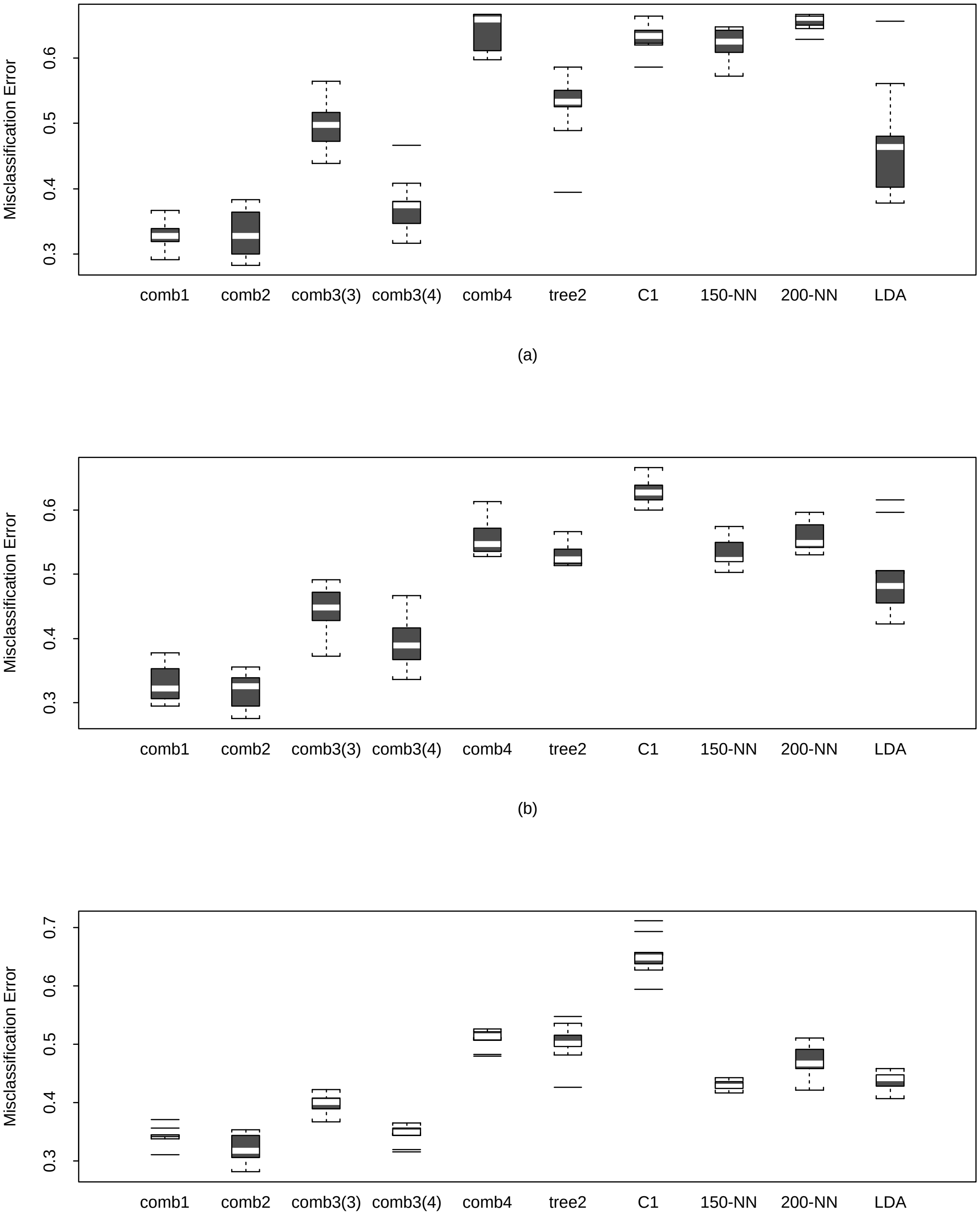
(c)

**Example 2.** In this example we consider combining a number of relatively useless classifiers. These are a 150-NN classifier, a 200-NN classifier, a tree classifier with just 2 terminal nodes, the LDA and a new classifier $C(\cdot)$, defined by

$$C(\mathbf{x}) = \begin{cases} 1 & \text{if } \parallel \mathbf{x} \parallel^2 < 10, \\ 2 & \text{if } \parallel \mathbf{x} \parallel^2 \geq 30, \\ 3 & \text{if } 10 \leq \parallel \mathbf{x} \parallel^2 < 30. \end{cases}$$

There are $k$=3 classes. The first class is a multivariate normal with the mean vector (1, 1, 1, 1, 1) and the diagonal covariance matrix diag$\{0.5,\ 0.5,\ 0.5,\ 0.5,\ 0.5\}$; the other two classes are the same as classes 2 and 3 of Example 1. Once again, we have considered three different sample sizes: A) 70 from each class, B) 200 from each class, and C) 400 from each class. Observe that both the 150-NN and the 200-NN rules are poor classifiers, (recall that a $k_n$-NN rule is consistent when $k_n/n \to 0$, as $n \to \infty$). Also, note that the new rule $C(\mathbf{x})$ is a wrong classifier because it confuses classes 2 and 3. A tree classifier with just 2 terminal nodes is not appropriate for separating three classes. Finally, the LDA is clearly not suitable for separating normal populations (with different covariance matrices) and a Cauchy population. The results are summarized in rows A, B, and C of Table 2; these correspond to different training sample sizes. Side-by-side boxplots are in Figure 2. Table 2 shows that combining a number of incorrect or useless classifiers can still yield substantial improvement over the individual classifiers; this is particularly true in the case of comb1, comb2, and comb$3_4$. The combined classifier comb4 fails to produce any good results no matter how large the sample size is.

Figure 2: Side-by-side boxplots of the error rates for Example 2



(a)

(b)

(c)

Table 2: Error rates for Example 2

| | comb1 | comb2 | comb3$_3$ | comb3$_4$ | comb4 | tree2 | C($\mathbf{x}$) | 150-NN | 200-NN | LDA |
|---|---|---|---|---|---|---|---|---|---|---|
| A | .331 | .331 | .496 | .370 | .645 | .531 | .634 | .621 | .657 | .464 |
| | (.021) | (.033) | (.032) | (.036) | (.027) | (.045) | (.019) | (.024) | (.011) | (.070) |
| B | .329 | .318 | .444 | .392 | .556 | .531 | .631 | .531 | .556 | .498 |
| | (.027) | (.029) | (.036) | (.038) | (.027) | (.019) | (.021) | (.021) | (.019) | (.061) |
| C | .341 | .320 | .399 | .345 | .509 | .501 | .651 | .428 | .470 | .438 |
| | (.015) | (.023) | (.015) | (.016) | (.015) | (.032) | (.032) | (.010) | (.030) | (.015) |

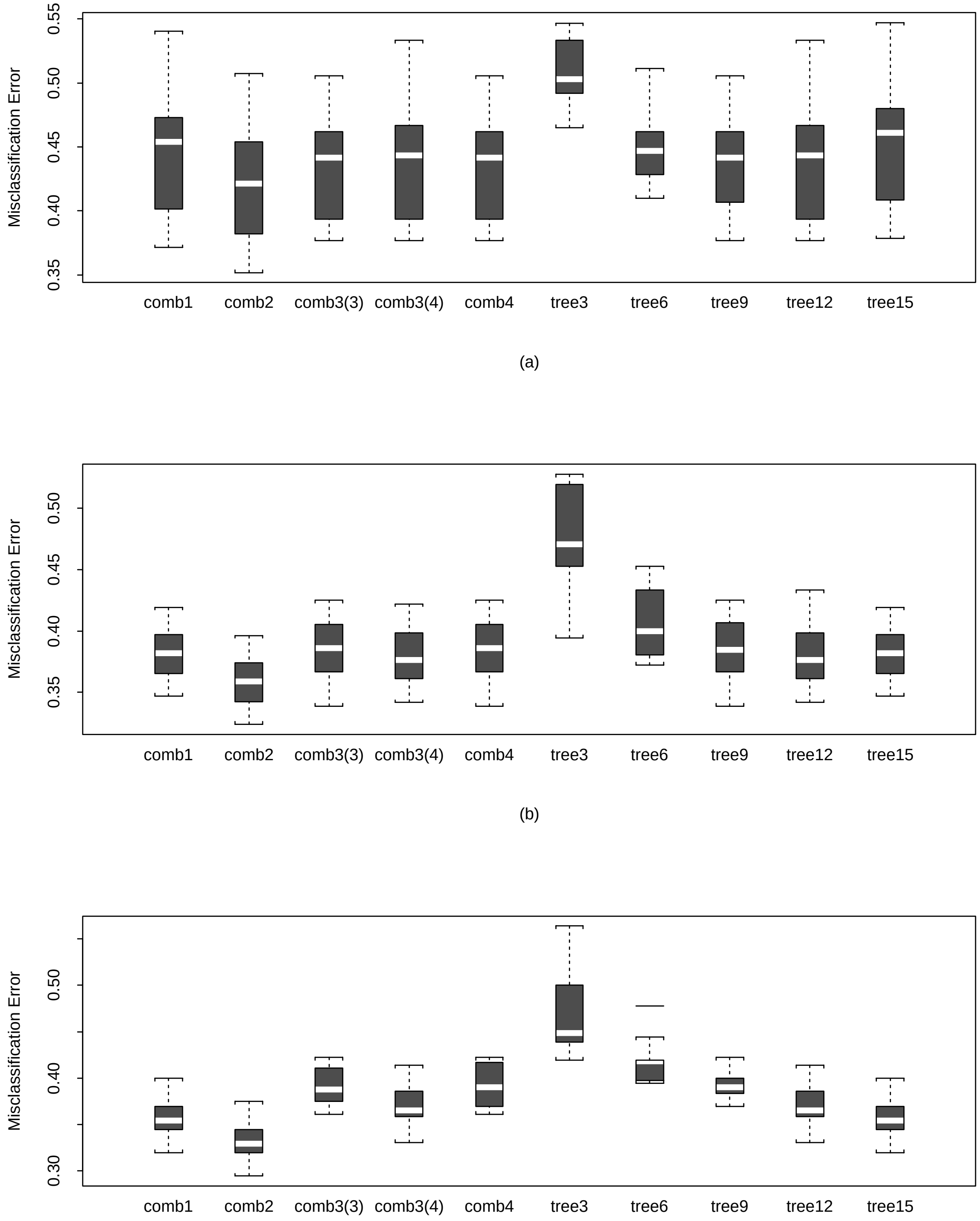
**Example 3.** In this example we consider using a combined classifier as an implicit model selection method. The idea may be summarized as follows. Quite often the classifier of interest depends on a tuning parameter $t$, whose value has to be selected from a set of candidate values $t_1, \cdots, t_p$. Examples of such classifiers are tree classifiers with $t=$ number of terminal nodes; generalized linear discriminant rules, where $t=$ number of fixed functions; and Nearest Neighbor classifiers with $t=$ number of nearest neighbors. Therefore, by combining all $p$ individual classifiers, one should expect to implicitly recover the one with the smallest error rate. In this example we propose to combine $M$=5 tree classifiers, where $t$ (the number of terminal nodes) takes the values 3, 6, 9, 12, 15. The classes are those of Example 1. The results are shown in Table 3, and the boxplots are

in Figure 3. Table 3 shows that for larger training sample sizes (rows B and C) both comb1 and comb2 perform quite well. The rule comb2 is clearly the winner for all three sample sizes. In fact comb2 outperforms all the other combined and individual classifiers. The combined classifier comb$3_4$ has also performed well in rows A and B of Table 3, but not row C. It is important to mention that although using comb1 or comb2 or comb$3_4$ can produce error rates at least as good as the best individual, the resulting combined classifier does not retain the interpretability property enjoyed by the individual classifiers; this is one of the trade-offs associated with combined classifiers.

Table 3: Error rates for Example 3

| | comb1 | comb2 | comb$3_3$ | comb$3_4$ | comb4 | tree3 | tree6 | tree9 | tree12 | tree15 |
|---|---|---|---|---|---|---|---|---|---|---|
| A | .444<br>(.044) | .421<br>(.041) | .433<br>(.038) | .439<br>(.042) | .434<br>(.036) | .508<br>(.026) | .448<br>(.028) | .436<br>(.037) | .438<br>(.042) | .451<br>(.044) |
| B | .382<br>(.024) | .358<br>(.024) | .385<br>(.028) | .380<br>(.027) | .385<br>(.028) | .476<br>(.045) | .406<br>(.030) | .385<br>(.029) | .381<br>(.030) | .382<br>(.024) |
| C | .359<br>(.025) | .334<br>(.025) | .391<br>(.021) | .372<br>(.026) | .393<br>(.024) | .468<br>(.044) | .419<br>(.025) | .393<br>(.016) | .372<br>(.026) | .359<br>(.025) |

Figure 3: Side-by-side boxplots of the error rates for Example 3



(a)

(b)

(c)

## 3. Conclusion

In this article we have considered a number of procedure for combining different classifiers in order to construct more effective classification rules with lower misclassification error rates. The combined rules $\psi_n^{\text{comb1}}$ and $\psi_n^{\text{comb2}}$ have the ability to outperform the individual classifiers under certain regularity conditions; these results are established in Mojirsheibani(1998a,b). Examples 1, 2, and 3 show that $\psi_n^{\text{comb3}}$ performs well in cases where $L_{\text{M}}$=4. This is no fluke! In fact we are currently preparing a manuscript that deals with the asymptotic performance of $\psi_n^{\text{comb3}}$ for carefully chosen values of $L_{\text{M}}$. Because of its linear structure, the combined classifier $\psi_n^{\text{comb4}}$ is intuitively appealing. This classifier is a modified version of the one that appeared in Breiman's (1997) article on *Arcing Algorithms.* Unfortunately we have not been able to establish any optimality results for this combining procedure. Also, most of our examples show that $\psi_n^{\text{comb4}}$ is not as effective as $\psi_n^{\text{comb2}}$ or $\psi_n^{\text{comb3}}$

We would also like to point out that in all of our examples we have deliberately selected three populations/classes which are very difficult to separate. As a result, the misclassification error rates are all quite high; this is also true for the combined classifiers. The message of the aricle, however, is that a combined classifier such as $\psi_n^{\text{comb3}_4}$ (or $\psi_n^{\text{comb2}}$ or $\psi_n^{\text{comb1}}$) can be at least as good as the best individual classifier.

## References

Breiman, L. (1995), "Stacked Regression," *Machine Learning*, 24, 49-64.

Breiman, L. (1997) "Arcing the Edge,"
(available at `http://www.stat.Berkeley.EDU/tech-reports/`)

LeBlanc, M. and Tibshirani, R. (1996), "Combining Estimates in Regression and Rlassification," *Journal of the American Statistical Association,* 91, 1641-1650.

Mojirsheibani, M. (1998a), "A Kernel-based Combined Classification Rule". Submitted to the *Annals of Statistics*

Mojirsheibani, M. (1998b), Combining Classifiers via Discretization, *Journal of the American Statistical Association,* To appear.

Mojirsheibani, M. (1997), "A Consistent Combined Classification Rule," *Statistics & Probability Letters*, 36, 43-47.

Wolpert, D. (1992), "Stacked Generalization", *Neural Networks*, 5, 241-259.