Combining Classifiers based on Gaussian Mixtures

Luis Daza and Edgar Acuña
Department of Mathematics, University of Puerto Rico at Mayagüez, Mayagüez, Puerto Rico 00680.

ABSTRACT

A combination of classification rules (classifiers) is known as an Ensemble, and in general it is more accurate than the individual classifiers used to build it. Two popular methods to construct an Ensemble are Bagging (Bootstrap aggregating) introduced by Breiman, [4] and Boosting (Freund and Schapire, [11]). Both methods rely on resampling techniques to obtain different training sets for each of the classifiers. Previous work has shown that Bagging as well as Boosting are very effective for unstable classifiers. In this paper we present some results in application of Bagging and Boosting to classifiers where the class conditional density is estimated using Gaussian mixtures. The effect of feature selection on the Ensemble is also considered.

Keywords: Gaussian mixtures, Supervised pattern recognition, Ensembles, Bagging, Boosting.

1. INTRODUCTION

Many researches have investigated the technique of combining the predictions of multiple classifiers to produce a single classifier: Breiman [4] [5], Quinlan [17], Freund and Schapire [11], Maclin and Optiz [15], Bauer and Kohavi [2], Acuña and Rojas [1] among others. Breiman [4] heuristically defines a classifier as unstable if an small change in the learning data L can make large changes in the classification. Unstable classifiers have low bias but high variance, meanwhile the opposite occurs for stable classifiers. CART and neural networks are not stable classifiers, linear discriminant analysis and k-nearst neighbor classifiers are stable. It is expected a reduction of the bias and variance after the classifiers are combined. The most well known techniques to combine classifiers are Bagging and Boosting. Both of them are very effective for unstable classifiers such as decision trees: CART, C4.5 and MC4 (see Breiman [4] [5], Quinlan [17], Freund and Schapire [11], Bauer and Kohavi [2], Dietterich [10] and neural networks (see Maclin and Optiz [15]). Ormoneit. and Tresp [16] applied Bagging to

gaussian mixtures classifiers, but they only tested it in two small datasets. Boosting applied to decision trees and Naïve-Bayes performs generally better that Bagging, but not uniformly better, sometimes they degraded compared to the single classifier. The same conclusions were obtained for neural networks classifiers. Bagging mainly reduces the variance, whereas boosting reduces, both the bias and the variance.

In this paper we present some experimental results on the application of Bagging and Boosting to classifiers where the class conditional density is estimated using Gaussian mixtures. We also discuss the effect of perform feature selection before building the Ensemble. Our experiments are applied to 11 datasets coming from the Machine Learning database repository at the University of California, Irvine.

2. CLASSIFIERS BASED ON GAUSSIAN MIXTURES

In the classical Fisher’s linear discriminant analysis (LDA) one assumes that the class conditional density is a multivariate normal, but in practice this assumption rarely holds. Therefore it is natural generalize the LDA by assuming that the class conditional density is a finite mixture of multivariate gaussian (normal) distributions with unknown parameters, Hastie and Tibshirani [12] named this generalization as Mixture discriminat analysis (MDA). This method also can be seen as a smoothed version of Learning Vector Quantisation (LVQ).

From a Bayesian point of view, supervised classification is equivalent to compare estimates of the probabilities of \( x \) belonging to each class with each other, and an object with measurement vector \( x \) is assigned to the class with the largest \( \hat{f}(j|x), j = 1, 2, \cdots, g \). In order to obtain such estimates, one can estimate them indirectly via the class conditional density \( f(x/j) \) using the Bayes’ theorem. Gaussian mixtures can be used to carry out
this task. For a given class $j$ with $n_j$ instances and a random sample $x_1, x_2, \ldots, x_n$ of the $p$-dimensional random vector $x$, the Gaussian mixture estimate of the class conditional density at the point $x$ is given by

$$
\hat{f}(x / j) = \sum_{k=1}^{K_j} \pi_{jk} \phi(x, \mu_{jk}, \Sigma) \quad j = 1, 2, \ldots, g
$$

where $\phi$ represents the multivariate normal density with mean vector $\mu_{jk}$ and covariance matrix $\Sigma$, and $K_j$ is the number of subclasses of the $j$-th class. As in the LDA it is assumed that all the subclasses have the same covariance matrix independently of the classes where they belong. Also, the nonnegative coefficients $\pi_{jk}$ must satisfy $\sum_{k=1}^{K_j} \pi_{jk} = 1$. The parameters $\mu_{jk}$, $\Sigma$ and $\pi_{jk}$ are estimated by maximum likelihood based on $x_1, x_2, \ldots, x_n$ using the EM algorithm. Naturally this is not the only mixture model that can be considered, one alternative is to assume that subclass has different covariance matrix, another alternative is to consider that in each class all the subclasses have equal covariance matrix. The chosen model is attractive because maintains the number of parameters under control and its structure allows other generalizations (Hastie and Tibshirani, [12]).

Let $\Pi_j$ be the prior probability for the $j$-th class, then using the model (1) the posterior probability of $x$ belonging to the $j$-th class is given by

$$
P(G = j/X = x_i) = \frac{\sum_{k=1}^{K_j} \pi_{jk} \phi(X; \mu_{jk}, \Sigma) \Pi_j}{\sum_{c=1}^{g} \sum_{k=1}^{K_c} \pi_{ck} \phi(X; \mu_{ck}, \Sigma) \Pi_c} \quad (2)
$$

Let $D(x, \mu) = (x - \mu)^T \Sigma^{-1} (x - \mu)$ be the Mahalanobis distance between $x$ and $\mu$, then the gaussian mixture model for the $j$-th class can be written as

$$
m_{x_i}(x) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \sum_{k=1}^{K_j} \pi_{jk} e^{-D(x, \mu_{jk})/2} \quad j = 1, \ldots, g \quad i = 1, \ldots, n
$$

The likelihood function for the parameter estimation of the mixture is given by

$$
L(\mu_{jk}, \Sigma, \pi_{jk}) = \prod_{i=1}^{n} m_{z_i}(x_i)
$$

$$
= \prod_{j=1}^{g} \left[ (2\pi)^{-j/2} |\Sigma|^{-1/2} \sum_{k=1}^{K_j} \pi_{jk} e^{-D(x_i, \mu_{jk})/2} \right]^{z_{ij}}
$$

Therefore the log-likelihood can be expressed as:

$$
\ell(\mu_{jk}, \Sigma, \pi_{jk}) = \sum_{i=1}^{n} \sum_{j=1}^{g} z_{ij} \log \left( \sum_{k=1}^{K_j} \pi_{jk} \phi(x; \mu_{jk}, \Sigma) \right) \quad (4)
$$

where $n$ is the number of training samples and $z_{ij}$ is the membership value of $x_i$ to the $j$-th group. Notice that $Z=(z_{i1}, z_{i2}, \ldots, z_{ig})$ has a multinomial distribution with parameters $(1; \pi_1, \pi_2, \ldots, \pi_g)$. Maximizing the log-likelihood with respect to the parameters $\theta = (\mu_{jk}, \Sigma)$, and under the restriction that $\sum_{k=1}^{K_j} \pi_{jk} = 1$ we obtain

$$
\hat{\mu}_{jk} = \frac{\sum_{i=1}^{n} z_{ij} x_i}{\sum_{i=1}^{n} z_{ij}}, \quad j = 1, 2, \ldots, g \quad k = 1, 2, \ldots, K_j
$$

and,

$$
\hat{\Sigma} = \frac{1}{n} \sum_{j=1}^{g} \sum_{i=1}^{n} z_{ij} (x_i - \hat{\mu}_{jk})(x_i - \hat{\mu}_{jk})^T \quad (5)
$$

Let $\hat{p}(c_{jk} / x_i, j)$ be the estimated conditional probability of object $x$ belonging to the $k$-th subclass of the $j$-th class. The expectation step (E step) of the algorithm EM, is given by:

$$
\hat{\pi}_{jk}(s) = \frac{1}{n_j} \sum_{i=1}^{n_j} z_{ij} \hat{p}(c_{jk} / x_i, j) \quad j = 1, 2, \ldots, n
$$

and

$$
\hat{\pi}_{jk} = \frac{1}{n_j} \sum_{i=1}^{n_j} z_{ij} \hat{p}(c_{jk} / x_i, j) \quad j = 1, 2, \ldots, n
$$
\[
\hat{p}(c_{jk}/x, j)_{(s+1)} = \frac{\pi^{(s)}_k e^{-D(x, \mu^{(s)}_k)/2}}{\sum_{k=1}^{K_j} \pi^{(s)}_k e^{-D(x, \mu^{(s)}_k)/2}}
\]  

(9)

The maximization step (M step) of the EM algorithm is given by

\[
\hat{\mu}^{(s+1)}_{jk} = \frac{\sum_{i=1}^{n} x_i p(c_{jk}/x, j)_{(s+1)}}{\sum_{i=1}^{n} p(c_{jk}/x, j)_{(s+1)}}
\]

(10)

\[
\hat{\Sigma}^{(s+1)}_{ij} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{K_j} \sum_{k=1}^{K_i} (x_i - \mu^{(s+1)}_{jk}) (x_i - \mu^{(s+1)}_{jk})^T p(c_{jk}/x, j)_{(s+1)}
\]

(11)

The EM algorithm requires (i) the election of the sizes of the \( K_j \) subclasses, in the \( j \)-th class, and (ii) the initial values for the means \( \mu_{jk} \), the covariance matrix \( \Sigma \) and the proportions \( p(c_{jk}/x, j) \) for the subclasses. In order to do that we have two different strategies: The clustering algorithm k-means and the neural network for supervised classification called Learning Vector Quantization (LVQ) (Kohonen, [13]). Both methods require initial centers that are obtained at random, but it can causes high variability, To overcome this problem, 10 random centers are chosen at random and the one given the lowest classification rate is chosen.

2.2. On the selection of the number of subclasses.

The determination of the number of subclasses per class is equivalent to the choice of the number of components of a finite mixture. This is a very important problem that still does not have a definitive solution. The solution depends of several factors, among them: the separability of the groups, the relative size of the groups and the dimensionality of the data (Webb, [20]). Bozdogan [6] used a criterion based on the Akaike’s Information criterion (AIC) to select the numbers of components of the mixture, and Dasgupta and Raftery [8] choose the number of components using an approximation to the Bayesian information criteria (BIC). A mixture is better than another if it has a larger BIC. The number \( M \) is chosen when the BIC reaches maximum per first time. are the more adequate criteria to determine the number of components of the mixture. Roeder and Wasserman [18], proved that for gaussian mixtures the BIC criterion is consistent to estimate the number of components of the mixture. Celeux and Soromenho [7] also proposed a method based on entropy.

Tenmoto et al. [19] proposed the Minimum Description Length Principle (MDL) to determine the number of components of classifiers based in Gaussian mixture. They realized that the determination of a number of components separately in each class could be not appropriate, in particular for training samples with a small number of instances. In the Tenmoto’s approach the mixture models are evaluated according to their classification performance, and a classifier with a large number of components is penalized to avoid overfitting. However the authors themselves mentioned that their approach needs to be improved.

In this work we have chosen the number of components of the mixture in a hill climbing way trying to maximize the recognition rate of the gaussian mixture classifier. A more detailed discussion of this can be found in Daza [9].

3. EXPERIMENTAL METHODOLOGY

The effect of combining Gaussian mixture (GM) classifiers was evaluated using 11 datasets coming from the Machine Learning Database Repository at University of California Irvine (UCI). All of them have been used in previous studies to evaluate the effect of combining classifiers [1]. A summary of the datasets appears in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Instances</th>
<th>Features</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breastw</td>
<td>699</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>Bupa</td>
<td>345</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>German</td>
<td>1000</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>Crx</td>
<td>690</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Diabetes</td>
<td>768</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Heartc</td>
<td>294</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Segment</td>
<td>2310</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>Sonar</td>
<td>208</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>Vehicle</td>
<td>846</td>
<td>18</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1. Datasets used in this paper
The setup for Bagging was as follows: Each dataset is randomly divided in 10 parts, the first one is taken as the test sample and the remaining is considered as the learning sample. Next, 50 bootstrapped samples are taking from the learning sample and a Gaussian mixture classifier is constructed with each of them. Finally, each instance of the test sample is assigned to a class by voting using the 50 classifiers previously constructed. The proportion of instances incorrectly assigned will be the bagged misclassification error. We repeat the steps considering now the second part as the test set and the others as the training sample and in this way we continue until the tenth part is considered as the test set. The whole procedure is repeated 10 times and we compute the average of the bagged misclassification error. On the misclassification error of a single classifier is estimated by a 10-fold crossvalidation and averaged over 50 runs. We also computed the ratio of the misclassification errors of the bagged classifier versus the single one. The number of subclasses was chosen between 2 and 10 in such way that the ratio will be the smallest one. The results are shown in Table 2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Subclasses</th>
<th>Single</th>
<th>Bagged</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>2</td>
<td>2.33</td>
<td>2.00</td>
<td>0.858</td>
</tr>
<tr>
<td>Sonar</td>
<td>3</td>
<td>24.24</td>
<td>18.90</td>
<td>0.780</td>
</tr>
<tr>
<td>Heart-c</td>
<td>5</td>
<td>18.46</td>
<td>16.57</td>
<td>0.898</td>
</tr>
<tr>
<td>Bupa</td>
<td>5</td>
<td>32.20</td>
<td>30.65</td>
<td>0.952</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>3</td>
<td>15.32</td>
<td>15.28</td>
<td>0.997</td>
</tr>
<tr>
<td>Crx</td>
<td>3</td>
<td>13.69</td>
<td>13.17</td>
<td>0.962</td>
</tr>
<tr>
<td>Breastw</td>
<td>3</td>
<td>4.30</td>
<td>3.70</td>
<td>0.860</td>
</tr>
<tr>
<td>Diabetes</td>
<td>5</td>
<td>25.50</td>
<td>24.09</td>
<td>0.945</td>
</tr>
<tr>
<td>Vehicle</td>
<td>4</td>
<td>20.18</td>
<td>17.53</td>
<td>0.869</td>
</tr>
<tr>
<td>German</td>
<td>5</td>
<td>24.33</td>
<td>23.7</td>
<td>0.974</td>
</tr>
<tr>
<td>Segment</td>
<td>6</td>
<td>7.19</td>
<td>5.71</td>
<td>0.794</td>
</tr>
</tbody>
</table>

Table 2. Comparison of misclassification error rates estimated by cross-validation for single and bagged GM classifiers.

The average of the misclassification error reduction for the 11 datasets after Bagging using the Gaussian mixture classifier was 10.1%. Notice that in two datasets: Sonar and Segment the misclassification decreased significantly whereas in Ionosphere the misclassification error remains the same after Bagging.

The setup for Boosting is quite similar to the one used in Bagging, the only difference is that a bootstrap sample depends on the misclassification errors on the previous one. In the first step a bootstrap sample is drawn from the original sample assigning equal weight to every instance. Then a classifier is built using the bootstrap sample and its misclassification error is computed. For the second bootstrap sample a instance has more weight if it was misclassified in the first step. The procedure continues until 20 bootstrap samples are drawn. Finally a weighted voting is applied to assign a object to a class.

The boosted misclassification error is averaged on 10 repetitions. We also computed the ratio of the misclassification errors of the boosted classifier versus the single one. The results appear in Table 3.

The average of the misclassification error reduction for the 11 datasets after Boosting using the Gaussian mixture classifier was almost none, only 0.80%. When C4.5 classifier was boosted (Quinlan, [17]) the average error reduction for the same datasets was 8.83%. Notice that the boosted classifier performed well only in the two smaller datasets Iris and Sonar.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sub-Classes</th>
<th>Single</th>
<th>Boosted</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>2</td>
<td>2.33</td>
<td>2.00</td>
<td>0.858</td>
</tr>
<tr>
<td>Sonar</td>
<td>3</td>
<td>24.24</td>
<td>20.86</td>
<td>0.861</td>
</tr>
<tr>
<td>Heart-c</td>
<td>4</td>
<td>17.83</td>
<td>19.53</td>
<td>1.095</td>
</tr>
<tr>
<td>Bupa</td>
<td>2</td>
<td>33.07</td>
<td>32.93</td>
<td>0.996</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>3</td>
<td>15.32</td>
<td>16.07</td>
<td>1.049</td>
</tr>
<tr>
<td>Crx</td>
<td>2</td>
<td>13.48</td>
<td>13.94</td>
<td>1.034</td>
</tr>
<tr>
<td>Breastw</td>
<td>5</td>
<td>4.57</td>
<td>4.33</td>
<td>0.947</td>
</tr>
<tr>
<td>Diabetes</td>
<td>5</td>
<td>25.50</td>
<td>24.97</td>
<td>0.979</td>
</tr>
<tr>
<td>Vehicle</td>
<td>4</td>
<td>20.18</td>
<td>20.71</td>
<td>1.026</td>
</tr>
<tr>
<td>German</td>
<td>2</td>
<td>24.62</td>
<td>26.07</td>
<td>1.059</td>
</tr>
<tr>
<td>Segment</td>
<td>6</td>
<td>7.19</td>
<td>7.25</td>
<td>1.008</td>
</tr>
</tbody>
</table>

Table 3. Comparison of misclassification error rates estimated by cross-validation for single and boosted GM classifiers.
4. EFFECT OF FEATURE SELECTION ON THE ENSEMBLE'S PERFORMANCE

To speed up the computation of the ensembles, first we perform feature selection. A sequential forward selection procedure was used and repeated 10 times for datasets with less than 20 features and 20 times for those with more than 20 features. First, we select the single feature that produces the highest classification rate estimated by 10-fold cross-validation using the classifier based on kernel density estimator. Once that this is done we search for the second feature that, together with the first one yields the highest classification rate. The procedure continues until the classification rate does not increase. After that we compute the average number of selected features for each dataset, rounding it if it is necessary. Finally, for each dataset, we select the features appearing more frequently in the 10 replications. In table 4 we show the effect of the forward feature selection.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Sub classes</th>
<th>Before</th>
<th>After</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breastw</td>
<td>3</td>
<td>4.30</td>
<td>4.19</td>
<td>0.974</td>
</tr>
<tr>
<td>Bupa</td>
<td>5</td>
<td>32.20</td>
<td>29.91</td>
<td>0.929</td>
</tr>
<tr>
<td>German</td>
<td>5</td>
<td>24.33</td>
<td>24.19</td>
<td>0.994</td>
</tr>
<tr>
<td>Crx</td>
<td>3</td>
<td>13.69</td>
<td>13.55</td>
<td>0.990</td>
</tr>
<tr>
<td>Diabetes</td>
<td>5</td>
<td>25.50</td>
<td>22.74</td>
<td>0.892</td>
</tr>
<tr>
<td>Heart-c</td>
<td>5</td>
<td>18.46</td>
<td>19.78</td>
<td>1.072</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>3</td>
<td>15.32</td>
<td>12.74</td>
<td>0.832</td>
</tr>
<tr>
<td>Segment</td>
<td>6</td>
<td>7.19</td>
<td>5.48</td>
<td>0.762</td>
</tr>
<tr>
<td>Sonar</td>
<td>3</td>
<td>24.24</td>
<td>22.18</td>
<td>0.915</td>
</tr>
<tr>
<td>Vehicle</td>
<td>4</td>
<td>20.18</td>
<td>21.54</td>
<td>1.067</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td>0.943</td>
</tr>
</tbody>
</table>

Table 4. Effect of forward feature selection on the performance of GM classifiers

Notice that in all the datasets, except two: Heartc and Vehicle, forward feature selection reduces the misclassification error. Once that the predictors are selected we generate the best subsets for each of the datasets and then we perform bagging and boosting using GM classifiers.

5. CONCLUDING REMARKS

Our experiments lead us to the following conclusions:

a) On average, the use of bagging for gaussian mixtures classifiers uniformly reduces error misclassification rate and has a better performance than the Boosting algorithm when it is applied to classifiers based on Gaussian mixtures.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Sub classes</th>
<th>Single</th>
<th>Bagged</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breastw</td>
<td>3</td>
<td>4.19</td>
<td>3.56</td>
<td>0.851</td>
</tr>
<tr>
<td>Bupa</td>
<td>5</td>
<td>29.91</td>
<td>28.46</td>
<td>0.952</td>
</tr>
<tr>
<td>German</td>
<td>5</td>
<td>24.19</td>
<td>24.03</td>
<td>0.994</td>
</tr>
<tr>
<td>Crx</td>
<td>3</td>
<td>13.55</td>
<td>13.38</td>
<td>0.988</td>
</tr>
<tr>
<td>Diabetes</td>
<td>5</td>
<td>22.74</td>
<td>21.64</td>
<td>0.952</td>
</tr>
<tr>
<td>Heart-c</td>
<td>5</td>
<td>19.78</td>
<td>18.11</td>
<td>0.916</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>3</td>
<td>12.74</td>
<td>11.91</td>
<td>0.934</td>
</tr>
<tr>
<td>Segment</td>
<td>6</td>
<td>5.48</td>
<td>5.04</td>
<td>0.919</td>
</tr>
<tr>
<td>Sonar</td>
<td>3</td>
<td>22.18</td>
<td>21.92</td>
<td>0.988</td>
</tr>
<tr>
<td>Vehicle</td>
<td>4</td>
<td>21.54</td>
<td>20.18</td>
<td>0.937</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td>0.943</td>
</tr>
</tbody>
</table>

Table 5. Effect of forward feature selection on the Misclassification errors by Bagging.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sub classes</th>
<th>Single</th>
<th>Boosted</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast-w</td>
<td>3</td>
<td>4.19</td>
<td>4.36</td>
<td>1.042</td>
</tr>
<tr>
<td>Bupa</td>
<td>5</td>
<td>29.91</td>
<td>31.48</td>
<td>1.052</td>
</tr>
<tr>
<td>Credit-g</td>
<td>5</td>
<td>24.19</td>
<td>24.40</td>
<td>1.009</td>
</tr>
<tr>
<td>Crx</td>
<td>3</td>
<td>13.55</td>
<td>13.87</td>
<td>1.024</td>
</tr>
<tr>
<td>Diabetes</td>
<td>5</td>
<td>22.74</td>
<td>23.49</td>
<td>1.033</td>
</tr>
<tr>
<td>Heart-c</td>
<td>5</td>
<td>19.78</td>
<td>21.28</td>
<td>1.076</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>3</td>
<td>12.74</td>
<td>14.25</td>
<td>1.118</td>
</tr>
<tr>
<td>Segment</td>
<td>6</td>
<td>5.48</td>
<td>6.16</td>
<td>1.123</td>
</tr>
<tr>
<td>Sonar</td>
<td>3</td>
<td>22.18</td>
<td>25.87</td>
<td>1.166</td>
</tr>
<tr>
<td>Vehicle</td>
<td>4</td>
<td>21.54</td>
<td>22.67</td>
<td>1.053</td>
</tr>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td>1.07</td>
</tr>
</tbody>
</table>

Table 6. Effect of forward feature selection on the Misclassification errors by Boosting.

b) Boosting is does not seem to be adequate to be applied to the GM classifiers, it does not reduce the probability of bad classification and in most of the cases it increases it.

c) The forward feature selection method applied to Gaussian mixtures classifiers yields good results. In average, a reduction in misclassification error rate is obtained. but this improvement is not uniform In some cases, the misclassification error rate increases.
d) Bagged Gaussian mixtures classifiers improve after feature selection but as in the case of single classifiers this is not uniform. Boosting after feature selection performs poorly (see tables 5 and 6).

The S-plus programs used in this paper are available in www.math.uprm.edu/~edgar/research.htm.

**Acknowledgment**

This work was supported by the Office of Naval Research grant N00014-00-1-0360 and by the PRECISE group of the Engineering School at the UPR-Mayaguez funded by NSF grant EIA 99-77071.

6. REFERENCES


