Improving the Performance of the HONG Network with Boosting

Ajantha S. Atukorale, Tom Downs
School of Information Technology and Electrical Engineering
University of Queensland, St Lucia, QLD 4072, Australia.

P. N. Suganthan
School of Electrical and Electronic Engineering
Nanyang Technological University
Republic of Singapore, 639798

Abstract This paper gives a brief description of a hierarchical architecture (HONG) that has been described elsewhere. The learning algorithm it uses is a mixed unsupervised/supervised method with most of the learning being unsupervised. The architecture generates multiple classifications for every data pattern presented, and combines them to obtain the final classification. The main purpose of this paper is to show how boosting can be used to improve the performance of the HONG classifier.

1 INTRODUCTION

Ensemble learning has had considerable success in reducing classification error from learned systems [7, 13]. Combining multiple classifiers aims to mask the errors produced by individual classifiers in order to improve the performance of the overall system. It is often easier to apply several different types of classifiers in this way rather than trying to develop a single, more sophisticated architecture to achieve the same accuracy. This approach has quite a long history [11], although it has been receiving more attention in the machine learning community in recent times. In the last decade, many researchers have developed different procedures for ensemble learning and among these, boosting technique has been quite prominent [6].

The classifier used in this paper is based on the neural gas algorithm [9] using a hierarchical architecture (HONG) [2] whose individual classifiers are combined to provide the final classification. This paper shows how a boosting method can be used to improve the performance of the HONG classifier. The performance of the boosted HONG network is compared with that of other techniques on two well-known benchmark data sets.

The paper is organized as follows. In the next two sections we describe the HONG network architecture and how the AdaBoost algorithm can be applied to it. In the fourth section, the performance of the boosted HONG network is presented. The paper concludes with some brief remarks in section five.

2 THE HONG ARCHITECTURE

By retaining the essence of the original NG algorithm [9], we have been able to develop what we refer to as a hierarchical overlapped neural gas (HONG) network architecture for pattern recognition problems [2]. The basic aim in this type of structure is to build a hierarchy in which the higher levels specialize in distinguishing between patterns that belong to different classes but which are close together in pattern space. In order to use the original NG algorithm efficiently in an architecture like this, we modified the original NG algorithm, which has a time complexity of $O(N \log N)$, to run faster in its sequential implementation with a complexity of $O(N)$ [2]. A brief review of the HONG architecture is given next.

First a base network is initialized. Once the number of neurons in the base network has been selected their reference vectors (also known as codebook vectors) are initialized randomly, and our accelerated version of the NG algorithm is applied to adapt the reference vectors. Having completed the unsupervised NG learning, a labeled set of training patterns is presented to the network and the neurons in the base network are labeled according to the pattern class that they most frequently respond to. The learning vector quantization (LVQ) algorithm [8] is then applied to fine-tune the decision boundaries given by the unsupervised learning. The LVQ algorithm moves the reference vectors away from the decision surface to separate the class borders more accurately. The real advantage of combining supervised and unsupervised learning in an architecture like this comes when the training set contains a large number of unlabeled patterns and a small number of labeled patterns. In such a situation, this model uses the unlabeled data set for the unsupervised learning, and uses the labeled data set to label the neurons in the network as well as to fine tune the labeled network in a supervised

1With respect to the execution speed of a sequential implementation.
manner.

Now we turn to the second level of the hierarchical structure. At this level a new NG network is created for each neuron in the base network. This is depicted in Figure 1 where networks $A'$, $B'$ and $C'$ are networks in the second level that are created for the base level neurons $A$, $B$ and $C$. Each network in the second level is initialized in a similar fashion to the initialization of the base network. The reference vectors of neurons of the second level network are initialized randomly but around the reference vector of their corresponding base neuron. Each network is trained on all training patterns for which the base neuron for that network is either the winner or one of the $p$ runners-up, $p$ being a prespecified number. We refer to the second level networks as overlapped NG networks because they are trained with overlapping sets of training patterns (i.e., individual training patterns are in general employed in the training of more than one second level NG network.) After training, the neurons in the overlapped NG (ONG) networks are labeled using the same procedure as is used in the base network.

Now we come to the test phase, which we commence by feeding a test pattern to the base network. We employ the winning neuron as well as the $q$ runner up neurons in the base network to identify those networks in the second level that will be used to vote on the class of the test pattern. The value of $q$ is set to a little less than $p$ (used in training) so that neurons at the base level, in which we have less confidence, are ignored. Experience has shown that this leads to improved performance. So, in the test phase, for a given test pattern we consider the winner and the $q$ runner-up ONG networks (i.e., $q + 1$ ONG networks) to implement the final classification step. Each ONG network participating in the classification for a given pattern outputs its decision as a confidence vector\(^2\) with the confidence scores for each class lying between 0 and 1. A confidence score of 1 for a certain class indicates absolute certainty that the pattern is in that class and a confidence score of 0 implies absolute certainty that the pattern is not in that class.

In order to combine the outputs of the $q + 1$ ONG networks, various classifier fusion techniques can be used [1, 2], but we have found that best results are obtained by interpreting the outputs as fuzzy membership values and using the fuzzy integral to implement the fusion.

\(^2\)These confidence scores do not generally sum to one.

3 ADABOOST ALGORITHM WITH HONG ARCHITECTURE

As explained in [6], the AdaBoost algorithm can be implemented by either weighted random sampling from

\[ \epsilon_t = \frac{1}{2} \sum_{(i,y) \in B} D_t(i,y)(1 - h_t(x_i, y_i) + h_t(x_i, y)) \]

where $D_t(\cdot, \cdot)$ is a “mislable” distribution, $h_t(\cdot, \cdot)$ is a
plausibility measure and \( t \) is a count of the number of rounds of boosting.

In [6] the plausibility measure was calculated based upon probability arguments, but in our approach we employ the fuzzy integral which, according to Yager [14] can, for finite sets, be calculated using the expression

\[
e = \max_{i=1,2,\ldots,n} \{ \min(q(y_i), g(A_i)) \} \quad (2)
\]

where

\[ A_i = \bigcup_{j=1}^{i} \{ y_j \} = \{ y_1, y_2, \ldots, y_i \} \]

The function \( q(\cdot) \) in (2) is referred to in the fuzzy literature as a *partial evaluation* and \( g(\cdot) \) is referred as a *degree of importance* and when they are combined by the fuzzy integral they give an overall evaluation of the object under study. We use this overall evaluation as the plausibility measure required by AdaBoost.M2. From the HONG network the partial evaluations \( q(\cdot) \) are obtained from the outputs of the ONG network which are probabilities that the data belong to different classes. And the degrees of importance \( g(\cdot) \) are given by the proportion of correct classifications obtained from each of the ONG networks. For more details on this see [3] where the technique is used for the basic network without boosting. Evaluation of the fuzzy integral using the \( q(\cdot) \) and \( g(\cdot) \) values obtained from the HONG network gives the plausibility measure required by AdaBoost.M2.

\section{Performance of the HONG Network}

As mentioned earlier, the HONG network was mainly developed for learning from a mixture of labeled and unlabeled data. Other published methods that have used boosting have employed labeled data only. So, in order to compare with these other methods, we are obliged to use fully-labeled data. In order for this comparison to be fair, we use the full set of labels to fine tune the HONG network after the clustering procedure has been implemented.

The two datasets we consider for performance comparison are the UCI Satellite data set (SatImage) generated from Landsat multi-spectral scanner image data and the UCI Letter-recognition data set (Letters) of off-line machine printed alphabetical characters [4]. For these data sets, performance of the boosted HONG network is compared with that of recently published best results.

The SatImage dataset is a sub-area of a scene, consisting of \( 80 \times 100 \) pixels, each pixel covering an area on the ground of approximately \( 80 \times 80 \) meters. The information given for each pixel consists of the class value and the intensities in four spectral bands. This dataset contains patterns with 36 attributes each of which belongs to one of 6 classes. The dataset was divided into a training and a test set with 4,435 patterns in the training set and 2,000 patterns in the test set for all the algorithms used in Table 1 (see [10] for more details).

The results on the test set, averaged over 10 random runs, are compared in Table 1. For the HONG network, the results shown in the table are after 75 rounds of boosting. Improvements in the generalization error beyond 75 rounds of boosting are negligible. It took about 52 minutes (on a Pentium III PC) to carry out the boosting process on the HONG network for this data set, and took just a few seconds to test the full data set once it was trained.

On the Letters dataset, the objective is to classify each of a large number of black and white rectangular pixel displays as one of the 26 capital letters of the English alphabet. The character images produced are based on 20 different fonts and each letter within these fonts has been randomly distorted to produce a file of 20,000 unique images. The parent fonts represented a full range of character types including script, italic, serif and Gothic. Each image was represented by 16 numerical attributes. Among the 20,000 patterns, the first 16,000 patterns were used to train, and the remaining 4,000 patterns were used to test for all the algorithms used in Table 2.

Results for the test set, averaged over 10 random runs, are compared in Table 2. For this data set, the results shown in the table are after 100 rounds of boosting beyond which improvements were insignificant. It took about 6 hours (on a Pentium III PC) to train the boosted HONG network for this data set, and just a few seconds to test the full data set after training (about 4 and a half minutes for a single round of boosting). The best results published for this data set are by Schwenk et. al. [12] but their boosted MLP system took more than a week on a fast processor (SGI Origin-2000) to train.

\begin{table}[h]
\centering
\begin{tabular}{llll}
\hline
Author & Algo. & Error (%) Without & Error (%) With \\
\hline
Freund et. al. [6] & C4.5 & 14.8 & 8.9 -M1 \\
Schwenk et. al. [12] & MLP & 12.8 & 8.1 -M2 \\
HONG & 8.9 & 8.8 -M1 & 7.7 -M2 \\
\hline
\end{tabular}
\caption{Test error rates on the UCI SatImage dataset without/with boosting by the algorithms compared. The first 4,435 patterns of the data set are used for training and the last 2,000 patterns are used for testing. Note: The arc-fs algorithm by Breiman is quite similar to the AdaBoost.M1 algorithm. Here M1 and M2 refers to the AdaBoost.M1 and AdaBoost.M2 algorithms respectively.}
\end{table}
Table 2: Test error rates on the UCI Letters dataset without/with boosting by the algorithms compared. The first 16,000 patterns of the data set are used for training and the remaining 4,000 patterns are used for testing.

<table>
<thead>
<tr>
<th>Author</th>
<th>Algo.</th>
<th>Error (%) Without</th>
<th>Error (%) With</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breiman [5]</td>
<td>CART</td>
<td>12.4</td>
<td>3.4 -arc-fs</td>
</tr>
<tr>
<td>Freund et. al. [6]</td>
<td>C4.5</td>
<td>13.8</td>
<td>3.3 -M1</td>
</tr>
<tr>
<td>Schwenk et. al. [12]</td>
<td>MLP</td>
<td>6.1</td>
<td>1.5 -M2</td>
</tr>
<tr>
<td></td>
<td>HONG</td>
<td>4.2</td>
<td>3.6 -M1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.9 -M2</td>
</tr>
</tbody>
</table>

The results in the two tables show that, as one would expect, AdaBoost.M2 performs better than AdaBoost.M1 on these two multi-class classification problems.

5 CONCLUSIONS

For pattern classification we have developed a classifier based on the neural gas algorithm with a hierarchical overlapped architecture called HONG, whose individual classifiers are combined to obtain the final classification. One of the important advantages of this system is that it allows the incorporation of a degree of supervised learning into a largely unsupervised scheme. This paper has shown that the performance of the HONG network can be further improved by use of the boosting algorithm. Excellent recognition rates on two well-known data sets were obtained.

References


