

# Comparative Study of Boosting and Non-Boosting Training for Constructing Ensembles of Acoustic Models

Rong Zhang and Alexander I. Rudnicky

Language Technologies Institute, School of Computer Science  
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

{rongz, air}@cs.cmu.edu

## Abstract

This paper compares the performance of Boosting and non-Boosting training algorithms in large vocabulary continuous speech recognition (LVCSR) using ensembles of acoustic models. Both algorithms demonstrated significant word error rate reduction on the CMU Communicator corpus. However, both algorithms produced comparable improvements, even though one would expect that the Boosting algorithm, which has a solid theoretic foundation, should work much better than the non-Boosting algorithm. Several voting schemes for hypothesis combining were evaluated, including weighted voting, un-weighted voting and ROVER.

## 1. Introduction

Ensembles of classifiers have received increasing interest in recent years. An ensemble of classifiers is a collection of single classifiers, that is used to generate a hypothesis through majority voting (weighted or un-weighted) from its component classifiers. It has been observed that the combination of “weak” learners can result in a “strong” composite classifier whose accuracy is much better than that of any single classifier.

Boosting is the most extensively studied approach in the family of ensembles, due to its solid theoretic foundations [1]. In Boosting, single classifiers are iteratively trained in a fashion such that hard-to-classify examples are given increasing emphasis. More specifically, the algorithm maintains a probability distribution for the training data, and initially assigns every example equal weight. In each round, a new single classifier is learned from the current distribution. Meantime, a parameter that measures the classifier’s importance is determined in respect of its classification accuracy. The single classifier is then used to classify every training example. The probability distribution is updated in such a way that the weight of an example will be enhanced if it is misclassified, or reduced otherwise. As a result, those examples that are difficult to classify will receive more emphasis in the training of subsequent classifiers. In generalization, the single classifiers are composed to form the final classifier that outputs the hypothesis with the weighted majority vote.

Other approaches to ensemble construction exist. For example, Bagging is a commonly used method in machine learning field that randomly selects a number of examples from the original training set and produces a new single classifier on the selected subset [2]. In this paper we will call this a *non-Boosting* method.

Both Boosting and non-Boosting methods have recently been applied to speech recognition. Boosting has been applied to phoneme recognition [3], confidence annotation [4], speaker identification [5] and continuous speech recognition [6]. In

most cases significant reduction in word error rate was achieved. Non-Boosting methods have a long history in constructing ensembles of acoustic models. A well-known example is to train models based on the separation of gender (male or female), age (children or adult), and channel (microphone, telephone, or cell phone). More recently, multi-band models, which split the frequency range into a certain number of sub-bands and constructs models for each band [7] have been studied. In [8], we also proposed several non-Boosting training schemes that use word error rate as the basis to manipulate the distribution of data.

To date there have not been any direct comparisons of Boosting and non-Boosting methods for speech recognition under similar conditions. We believe that such comparison could benefit the understanding of ensemble techniques. This paper describes such a comparison between the performance of Boosting and non-Boosting training algorithms on a real-world continuous speech corpus.

Choice of classifier combination technique is also an important issue in ensembles. In speech recognition, Boosting and non-Boosting methods usually combine hypotheses at the string or utterance level using majority voting. This ignores the word confidence and timing information. Therefore, we also investigated ROVER (Recognizer Output Voting Error Reduction) [9] in our experiments, comparing its performance with the standard weighted and un-weighted majority voting.

## 2. Boosting and Non-Boosting Algorithms

Given a training set  $\mathbf{X} = \{\mathbf{x}\}$ , Boosting and non-Boosting algorithms aim to construct  $T$  acoustic models  $\{\mathbf{A}^t \mid 1 \leq t \leq T\}$  by manipulating the training data.  $\mathbf{x}$  could denote both the utterance and its feature sequence in this paper. Restricted to commonly available computers, the number of acoustic models that can be accommodated by a real-time LVCSR system would be in the single digits. As this paper focuses only on acoustic model training, the language model used is the same in all experiments.

### 2.1. Boosting Training

The Boosting algorithm was initially designed to solve the binary or multi-class classification problems, in which the number of classes isn’t very large. However, for continuous speech recognition, the number of possible hypothesis could be infinite. Assume the recognizer has 5,000 words in vocabulary, and the maximum length for an utterance is confined to 20 words. Theoretically, without concerning the time information associated to each word, the recognizer could output up to about  $5000^{20}$  different hypotheses. Apparently, such a huge number makes it impossible for the recognizer to traverse all of the classes. To address this problem, we have to compress the hypothesis space into a subset with limit size. In

our experiments, the subset of hypothesis only consists of the hypotheses in the N-best list.

Figure 1: Boosting Algorithm

<p>Initialize:</p> <ul style="list-style-type: none"> <li>Let <math>\mathbf{X}^0 = \mathbf{X}</math>.</li> </ul> <p>For <math>t = 1</math> to <math>T</math>:</p> <ul style="list-style-type: none"> <li>Train new acoustic model <math>\mathbf{A}^t</math> from data set <math>\mathbf{X}^{t-1}</math>.</li> <li>Test model <math>\mathbf{A}^t</math> on <math>\mathbf{X}^{t-1}</math>, generating N-best list for each utterance <math>\mathbf{x}</math>, and computing probability <math>P(\mathbf{u}   \mathbf{x}, \mathbf{A}^t, \mathbf{L})</math> for each hypothesis <math>\mathbf{u}</math> in the N-best list of <math>\mathbf{x}</math>.</li> <li>Compute pseudo loss           <math display="block">\mathcal{E}^t = \frac{\sum_{\mathbf{x}} \sum_{\mathbf{u} \neq \mathbf{u}^*} (1 - P(\mathbf{u}^*   \mathbf{x}, \mathbf{A}^t, \mathbf{L}) + P(\mathbf{u}   \mathbf{x}, \mathbf{A}^t, \mathbf{L}))}{2N   \mathbf{X}^{t-1}  }</math>           where <math>\mathbf{u}^*</math> denotes the correct hypothesis and <math>N</math> is the size of the N-best list.         </li> <li>Set <math>c_t = \mathcal{E}^t / (1 - \mathcal{E}^t)</math></li> <li>Calculate weight for each hypothesis <math>\mathbf{u}</math> in the N-best list of utterance <math>\mathbf{x}</math> <math display="block">w(\mathbf{x}, \mathbf{u}) = c_t \frac{1}{2^{(1+P(\mathbf{u}^*   \mathbf{x}, \mathbf{A}^t, \mathbf{L}) - P(\mathbf{u}   \mathbf{x}, \mathbf{A}^t, \mathbf{L}))}}</math> </li> <li>Calculate weight for each utterance <math>\mathbf{x}</math> <math display="block">w(\mathbf{x}) = \sum_{\mathbf{u} \in \mathbf{u}^*} w(\mathbf{x}, \mathbf{u})</math> </li> <li>Resample training data according to normalized <math>w(\mathbf{x})</math>, forming new training set <math>\mathbf{X}^t</math>.</li> </ul> <p>In generalization, the hypothesis to a new utterance is determined by <math>\mathbf{u}^* = \arg \max_{\mathbf{u}} \sum_{t=1}^T \log \frac{1}{c_t} P(\mathbf{u}   \mathbf{x}, \mathbf{A}^t, \mathbf{L})</math></p>
---

In implementation, Boosting requires a probability estimation for each class, while most speech recognizers only output the log-likelihood score of acoustic and language model:  $\log P(\mathbf{x} | \mathbf{u}, \mathbf{A})$  and  $\log P(\mathbf{u} | \mathbf{L})$ , where  $\mathbf{A}$  and  $\mathbf{L}$  denote the acoustic model and language model respectively, and  $\mathbf{u}$  represents a hypothesis. (For simplicity, the segmentation information isn't considered.) We use the following method converting the decoding score into probability.

$$P(\mathbf{u} | \mathbf{x}, \mathbf{A}, \mathbf{L}) = \frac{P(\mathbf{x} | \mathbf{u}, \mathbf{A})^\alpha P(\mathbf{u} | \mathbf{L})^\beta}{\sum_{\mathbf{u}' \in N\text{-best list}} P(\mathbf{x} | \mathbf{u}', \mathbf{A})^\alpha P(\mathbf{u}' | \mathbf{L})^\beta}$$

where  $\alpha$  and  $\beta$  are the weights associated with acoustic and language model. In the case that the correct hypothesis doesn't occur in the N-best list, one could use forced alignment to get the log-likelihood score, or just choose a small default value, e.g. assuming the correct hypothesis has the same score or probability as the last hypothesis in N-best list.

The Boosting algorithm for acoustic model training is given in Figure 1.

## 2.2. Non-Boosting Training

The non-Boosting algorithm is based on the intuition that the misrecognized utterance should receive more attention in the successive training. Word error rate is a natural metric that indicates how difficult an utterance is to recognize, and how

important it is in the training of the next model. For example, an utterance that has 100% word error with current acoustic model should have higher weight than one with 10% word error. Figure 2 shows the algorithm.

Figure2: Non-Boosting Algorithm

<p>Initialize:</p> <ul style="list-style-type: none"> <li>Let <math>\mathbf{X}^0 = \mathbf{X}</math>.</li> <li>Assign equal weight to each utterance <math>\mathbf{x}_i</math> that <math>w_i^0 = 1</math>.</li> </ul> <p>For <math>t = 1</math> to <math>T</math>:</p> <ul style="list-style-type: none"> <li>Train new acoustic model <math>\mathbf{A}^t</math> from data set <math>\mathbf{X}^{t-1}</math>.</li> <li>Test model <math>\mathbf{A}^t</math> on the initial training set <math>\mathbf{X}</math>, computing word error rate <math>\mathcal{E}_i^t</math> for each utterance <math>\mathbf{x}_i</math>.</li> <li>Update distribution <math>w_i^t = w_i^{t-1} (1 + \lambda \mathcal{E}_i^t)</math>.</li> <li>Resample training data according to <math>w_i^t</math>, forming new training set <math>\mathbf{X}^t</math>.</li> </ul> <p>In generalization, the hypothesis to a new utterance is determined by <math>\mathbf{u}^* = \arg \max_{\mathbf{u}} \sum_{t=1}^T P(\mathbf{u}   \mathbf{x}, \mathbf{A}^t, \mathbf{L})</math></p>
---

In Figure 2,  $\lambda$  is a parameter that prevents the size of the training set from being too large; in our experiments we set the value empirically. We also use an un-weighted majority voting scheme to combine the hypotheses of each model instead of the weighted voting adopted by Boosting.

## 3. Combination Methods

The combination methods used by the Boosting and non-Boosting algorithms described in the previous section represent two major techniques for combining hypotheses: weighted voting and un-weighted voting. We note that these two methods operate on the utterance level. That is, they ignore some important information associated with individual word in the hypothesis, such as confidence and segmentation (location). Such word level information has been shown to be able to improve recognition accuracy, e.g. through confidence annotation and N-best list re-ranking.

ROVER is a successful method for realizing word level hypothesis combination. First, the hypotheses from different acoustic models or recognizers are combined into a single word transition network by using dynamic programming alignment. Once the network is generated, a voting scheme respecting frequency, confidence and time information is used to search for the best scoring word sequence. This differs from the weighted and un-weighted voting adopted by Boosting and non-Boosting algorithms that select the most likely hypothesis from the existing set: ROVER can create a new hypothesis by merging existing word level hypotheses.

ROVER was initially intended to reduce word error rate by exploiting the difference between outputs from multiple speech recognition systems (representing different training and decoding approaches). However, we believe ROVER could also benefit ensembles even though all the models in the ensemble are trained using the same techniques, albeit with different views of the corpus. In the following experiments, ROVER is investigated and compared with the utterance level weighted and un-weighted voting.

## 4. Data Set and Experiment Configuration

The corpus used in our study was collected using the CMU Communicator system, a telephone based dialog system that supports planning in a travel domain [10]. The training set has 31,248 utterances, which were collected between April 1998 and November 2000. The test set consists of 1,689 utterances, which were collected during a NIST evaluation conducted in July 2000. There are 9,769 words in the vocabulary. The maximum number of models in each ensemble is set to 5.

All of our experiments, both training and test, are performed using the Carnegie Mellon Sphinx-2 system [11]. The decoding process consists of two passes: the first one is a Viterbi search, and the second one is an A\* search that generates N-best lists from the word lattice. We select the best path in the word lattice as the final hypothesis for each utterance. The baseline word error rate for our experiments is 33.31% and 28.18%, for test and training sets respectively.

## 5. Experiments

Our first experiment compares the performance of Boosting and non-Boosting algorithms using utterance level weighted or un-weighted voting schemes. Table 1 shows the final word error rates (when  $T = 5$ ) for these two algorithms on training and test set.

Alg.	Training error	Relative reduction	Test error	Relative reduction
Boosting	25.93%	7.98%	28.77%	13.63%
Non-B.	26.41%	6.28%	29.58%	11.20%

Table 1 Final Performance of Ensembles

More details can be found in Table 2 and 3, which show the algorithms' performance as a function of  $T$ . The baseline is equivalent to  $T = 1$  performance.

Alg.	T=1	T=2	T=3	T=4	T=5
Boosting	28.18%	27.08%	26.54%	26.15%	25.93%
Non-B.	28.18%	27.78%	27.51%	27.03%	26.41%

Table 2 Performance on Training Set

Alg.	T=1	T=2	T=3	T=4	T=5
Boosting	33.31%	30.89%	29.97%	29.46%	28.77%
Non-B.	33.31%	31.57%	31.38%	30.01%	29.58%

Table 3 Performance on Test Set

In this experiment, both the Boosting and non-Boosting algorithms show significant improvements over the baseline. On the training set, the Boosting algorithm produced 7.98% relative reduction in word error rate while non-Boosting produced 6.28% relative reduction. The number for the test set is more encouraging in that both algorithms realized better than 10% relative reduction in word error rate.

From Table 2 and 3, we find that the Boosting algorithm works consistently better than the non-Boosting algorithm. However, the difference between them isn't as great as we expected. In other word, the Boosting algorithm which has a solid theoretical background is not significantly better than the non-Boosting algorithm, which is only based on an intuitive idea. We think this phenomenon is very important to the application of ensembles of acoustic models, which indicates the direction of further research.

Our next experiment investigates the performance of ROVER-based combination. ROVER supports a trade-off between the frequency of word and confidence score. In our experiment, confidence score is calculated for each hypothesis word using the method given by [12]. The value of the trade-off factor is

determined by using a linear search procedure, minimizing the word error rate of the training set. The final performance of Boosting and non-Boosting algorithms using ROVER as the combination method is presented in Table 4.

Alg.	Training error	Relative reduction	Test error	Relative reduction
Boosting	22.23%	21.11%	26.78%	19.60%
Non-B.	22.79%	19.13%	27.34%	17.92%

Table 4 Final Performance of Ensembles with ROVER

More details of the performance of these two algorithms varying with  $T$  can be found in Table 5 and 6.

Alg.	T=1	T=2	T=3	T=4	T=5
Boosting	28.18%	25.30%	24.32%	22.31%	22.23%
Non-B.	28.18%	26.53%	24.98%	22.88%	22.79%

Table 5 Performance with ROVER on Training Set

Alg.	T=1	T=2	T=3	T=4	T=5
Boosting	33.31%	32.24%	29.20%	27.11%	26.78%
Non-B.	33.31%	32.75%	29.22%	27.08%	27.34%

Table 6 Performance with ROVER on Test Set

The numbers shown in Table 1 and Table 4 suggest that ROVER generally outperforms weighted and un-weighted voting in combining multiple utterance hypotheses. This strongly supports the view that word level combination is more suitable for speech recognition than the utterance level combination. On the other hand, no significant difference between Boosting and non-Boosting algorithms is observed. Moreover, there are two anomalous phenomena in Table 6 that surprised us. One is that when  $T = 2$ , the word error rate achieved by ROVER is higher than that given by weighted and un-weighted voting showed in Table 3. Another one is, for the non-Boosting algorithm, the word error rate increases to 27.34% when  $T = 5$  from 27.08% when  $T = 4$ . These results seem anomalous and suggest that a better understanding of ROVER is necessary.

Figure 3 and Figure 4 show comparisons for Boosting and non-Boosting algorithms using different combination methods.

## 6. Discussion

We experimentally investigated Boosting and non-Boosting algorithms for acoustic model training on a real world continuous speech corpus, both of which demonstrated significant improvements over the baseline. We also compared several voting schemes for combining hypotheses including weighted voting, un-weighted voting and ROVER. The experimental results suggest that word level combination is more suitable for speech recognition than utterance level combination.

In the experiments, we observed that the difference in performance between Boosting and non-Boosting algorithms is small. That is, Boosting doesn't manifest much advantage compared with non-Boosting methods. We think the following reasons may be responsible for its weakness, and believe that Boosting could achieve greater improvement by addressing these obstacles.

First, the training data for speech recognition, especially large corpus, unavoidably contains a certain number of mislabels. For such noisy data, the standard Boosting algorithm may perform very poorly (since the point of the procedure is to focus on poorly recognized data). Second, the training of acoustic models involves many stages, e.g. feature space clustering, context independent training and context dependent

training, each of which involves its own techniques and characteristics. It might be better to focus on the characteristics of each stage rather than simply use a unitary algorithm for the whole training process. Third, the training of acoustic models for continuous speech recognition is usually carried on the utterance level since it is difficult to get accurate time segment information at the word level. At the same time performance is evaluated on the word level. In Boosting, such mismatch appears in the cost function which is designed to simulate the string or utterance error, so minimizing this cost function doesn't always result in the reduction of word error. Integration of word level information to the cost function would be helpful to fill the gap between training and test metrics. Fourth, in the implementation of Boosting, the hypothesis space is confined to the N-best list. As a consequence, the information of the hypothesis out of N-best list is lost. The extreme case is that the correct hypothesis itself may not occur in the N-best list. In addition, the estimation for "a posteriori" probability of each hypothesis is based on the decoding score which could vary over a wide range, making the estimation far from reliable. Boosting algorithm is a discriminative training method that attempts to enlarge the distance or margin between correct hypothesis and incorrect hypothesis. Therefore, the experience and techniques gathered from discriminative training could benefit the application of Boosting algorithms to construct ensembles of acoustic models for speech recognition.

### 7. Acknowledgements

This research was sponsored in part by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

### 8. References

- [1] R. E. Schapire, "A brief Introduction to Boosting", Proc. of the 16<sup>th</sup> International Joint Conference on Artificial Intelligence, 1999.
- [2] L. Breiman, "Bagging Predictors", Machine Learning, 24(2): 123-140, 1996.
- [3] H. Schwenk, "Using Boosting to Improve A Hybrid HMM/Neural Network Speech Recognizer", Proc. of ICASSP 1999.
- [4] P. Moreno, B. Logan and B. Raj, "A Boosting Approach for Confidence Scoring", Proc. of EuroSpeech 2001.
- [5] S-W Foo and E-G Lim, "Speaker Recognition Using Adaptively Boosted Decision Tree Classifier", Proc. of ICASSP 2002.
- [6] C. Meyer, "Utterance-Level Boosting of HMM Speech Recognizers", Proc. of ICASSP 2002.
- [7] A. Hagen, H. Bourlard, and A. Morris, "Adaptive ML-Weighting in Multi-Band Recombination of Gaussian Mixture ASR," Proc. of ICASSP 2001.
- [8] R. Zhang and A. I. Rudnicky, "Improve the Performance of a LVCSR System through Ensembles of Acoustic Models", Proc. of ICASSP 2003.
- [9] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)", Proc. of ASRU 1997.
- [10] A. I. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Shern, K. Lenzo, W. Xu, A. Oh, "Creating Natural

Dialogs in the Carnegie Mellon Communicator System", Proc. of EuroSpeech 1999.

- [11] X. Huang, F. Alleva, H-W Hon, M-Y Hwang and R. Rosenfeld, "The SPHINX-II Speech Recognition System: An Overview", Technique Report, 510.7808 C28R 92-112, Carnegie Mellon University, 1992.
- [12] F. Wessel, K. Macherey and R. Schluter, "Using Word Probabilities as Confidence Measures", Proc. of ICASSP 1998.

Figure 3 Performance on Training Set

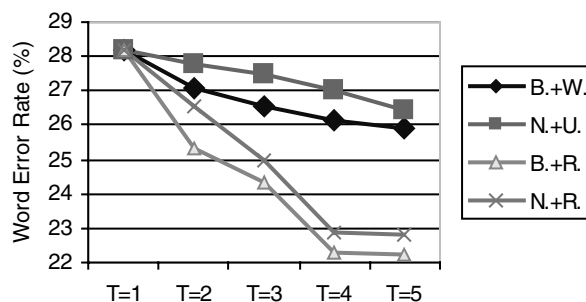


Figure 4 Performance on Test Set

