Apply N-Best List Re-Ranking to Acoustic Model Combinations of Boosting Training

Rong Zhang and Alexander I. Rudnicky

Language Technologies Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213, USA
{rongz,air}@cs.cmu.edu

Abstract

The object function for Boosting training method in acoustic modeling aims to reduce utterance level error rate. This is different from the most commonly used performance metric in speech recognition, word error rate. This paper proposes that the combination of N-best list re-ranking and ROVER can partly address this problem. In particular, model combination is applied to re-ranked hypotheses rather than to the original top-1 hypotheses and carried on word level. Improvement of system performance is observed in our experiments. In addition, we describe and evaluate a new confidence feature that measures the correctness of frame level decoding result.

1. Introduction

Boosting [1] has received increasing interests from speech recognition community. It differs from conventional training methods that aim to construct a single optimized classifier under some criteria e.g. MLE or MCE, by generating and exploiting multiple classifiers to improve recognition performance. A speech recognition system trained using the Boosting algorithm consists of a set of sub-systems, each of which has its own acoustic model and language model, and the final hypothesis is made by combining decoding results from all the sub-systems.

In Boosting, individual models or classifiers are trained in an iterative fashion such that hard-to-classify examples are given higher weight than others. Specifically, Boosting maintains a probability distribution for the training data and initially assigns equal weight to each example. In each training round, a new classifier is learned from the current distribution and then applied to classify every training example. After that, the probability distribution is updated in such a way that the weight of an example will be increased if it is misclassified, otherwise decreased. This enables the training of subsequent classifier to concentrate on the examples which are difficult to be correctly classified.

Boosting has been applied to various speech recognition problems, such as word and phone recognition [2], confidence annotation [3], speaker identification [4] and continuous speech recognition [5][6]. As one of the most challenging task in natural language processing and machine learning field, continuous speech recognition attracts more attention from researchers. The effectiveness of Boosting training methods is demonstrated by the observance of substantial improvements on recognition accuracy when use multiple acoustic models. However, our observation is that the object function for the Boosting algorithm used in acoustic model training focuses on reducing utterance level error rate rather than reducing word level error rate, the most commonly used performance metric in speech recognition. Although there is a strong correlation linking utterance and word errors, the mismatch between training criterion and target may not provide a satisfactory result. This weakness is mainly due to the unavailability of accurate word segmentation information within a continuous utterance. In other words, the characteristics of speech processing force us to accept and use a practical but suboptimal criterion.

To address this problem, one straightforward solution would be to modify the object function making it reflect word level errors. However, in this paper we approach this problem from a different perspective, by investigating post-processing techniques to improve system performance. Conventionally, the final hypothesis is formed by linearly combining top-1 hypotheses of each sub-system. In our experiments, we use confidence annotation techniques to first re-rank N-best lists, and thereby provide more reliable inputs for hypotheses combination. In addition, we replace conventional utterance level combination by ROVER [7], a word level combination method. Improvement is observed in our experiments, in which the word error rate on a real word corpus is reduced from 14.99% to 12.52%.

2. Boosting Algorithm for Acoustic Modeling

Generally speaking, the Boosting algorithm belongs to the discriminative training family whose goal is to increase the separability between the desired class and competing classes. Suppose we have a training set $\Psi = \{(\mathbf{x}_i, h_i) \,|\, 1 \le i \le N\}$, where $\mathbf{x}_i$ denotes a training example and h_i denotes its class label. Specifically, in continuous speech recognition, $\mathbf{x}_i$ is usually the sequence of feature vectors for the *i*-th training utterance, while h_i is the corresponding transcript. (Without confusion, the symbol h is also used to denote the hypothesis in this paper.) Same as other discriminative methods such as MCE, the Boosting algorithm is also based on an object function that is related to class confusion or classification error. One form is as follows (other variants exist).

$$L = \sum_{i=1}^{N} \sum_{h \neq h_i} \exp(f(\mathbf{x}_i; h) - f(\mathbf{x}_i; h_i)) \qquad (1)$$

where function $f(\mathbf{x}; h)$ could be interpreted as a classifier or speech recognizer that measures the likelihood that instance $\mathbf{x}$ is classified or recognized as class or hypothesis h. The essential of the object function is that minimizing L could lead to $f(\mathbf{x}_i, h_i) >> f(\mathbf{x}_i, h)$ for $h \neq h_i$ and consequently decrease the classification error. This is different from MCE in which $f(\mathbf{x}; h)$ is realized by a single model, in Boosting

$f(\mathbf{x};h)$ is the linear combination of predictions from multiple models $\{\lambda_k\}$:

$$f(\mathbf{x};h) = \sum_k c_k * f_{\lambda_k}(\mathbf{x};h) \quad (2)$$

where c_k is the weight for model λ_k.

Note that to make the Boosting algorithm work for acoustic model training, a couple of implementation issues need be addressed. First, the number of possible hypothesis h for a given utterance $\mathbf{x}$ could be infinite. Supposing a recognizer has 5000 words in vocabulary and the length of an utterance is confined to no more than 20 words, theoretically, without concerning segmentation information the recognizer could output up to about 5000^{20} different hypotheses. Obviously, such a huge number make it impossible to enumerate every hypothesis. To solve this problem, we have to compress the hypothesis space into a subset with limit size, e.g. the N-best lists.

Another issue of concern is how to define and calculate $f_\lambda(\mathbf{x};h)$. A good candidate for $f_\lambda(\mathbf{x};h)$ in acoustic modeling is the posterior probability $P_\lambda(h|\mathbf{x})$. However, most speech recognizers only output the log-likelihood score or joint probability $P_\lambda(h,\mathbf{x})$ rather than the posterior probability. We use the following method converting the joint probability into posterior probability.

$$P_\lambda(h|\mathbf{x}) \approx \frac{P_\lambda(h,\mathbf{x})^\beta}{\sum_{h' \in \text{ n-best list of } \mathbf{x}} P_\lambda(h',\mathbf{x})^\beta} \quad (3)$$

where β is the smoothing parameter, whose value is empirically set. In the case that the correct hypothesis may not exist in the N-best list, one could run forced alignment for it to get the log-likelihood score, or just use a small default value. Figure 1 shows the pseudo code of a modified Boosting algorithm that is suitable for acoustic modeling.

Our analysis shows that the Boosting algorithm tries to increase utterance accuracy, but this is correct only when all the words in the utterance are recognized correctly. In other circumstances, the commonly used word accuracy would be more appropriate. First, the posterior probability $P_\lambda(h|\mathbf{x})$ appearing in object function and pseudo loss is a metric defined on the utterance level. This is not sufficient for determining how many words in the utterance are misrecognized. For example, in some cases the hypothesis with the highest posterior probability in N-best list may not be the one with lowest word errors. In other words, there is considerable mismatch between the training criterion, to minimize an utterance level object function, and training target, to minimize word error rate. Second, the combination method used by the Boosting method to generate final hypothesis also works on the utterance level. The essential of this method is to select the most likely result from the existing top-1 hypotheses through majority voting. This means the candidates are constrained within the set of top-1 hypothesis, and any hypothesis out of this set won't be able to be selected or generated.

To address the problem mentioned above, one could modify the object function by integrating word or phoneme information. However, in this paper we focus on post processing techniques which in the past have received less attention. In our experiments, we use confidence annotation techniques to re-rank N-best lists, selecting more reliable results for hypotheses combination. In addition, we use ROVER to achieve word level combination. We expect these techniques can at least partly fill the gap between training criterion and training target.

Figure 1: Ada-Boosting Algorithm

Initialize:

- Let $\mathbf{U}^0 = \mathbf{U}$.
- Assign equal weight to each utterance $\mathbf{x}_i$ that $w_i^0 = 1$.

For k = 1 to K:

- Train new acoustic model λ_k from data set $\mathbf{U}^{k-1}$.
- Test model λ_k on the training set $\mathbf{U}^{k-1}$, generating N-best list for each utterance $\mathbf{x}_i$, and computing probability $P_{\lambda_k}(h|\mathbf{x}_i)$ for each hypothesis h in the N-best list of $\mathbf{x}_i$.
- Compute pseudo loss

$$\varepsilon^k = \frac{1}{2|h||\mathbf{U}^{k-1}|} \sum_{\mathbf{x}_i \in \mathbf{U}^{k-1}} \sum_{h \neq h_i^*} (1 - P_{\lambda_k}(h_i|\mathbf{x}_i) + P_{\lambda_k}(h|\mathbf{x}_i))$$

- Set $c_k = \varepsilon^k / (1-\varepsilon^k)$
- Calculate weight for each hypothesis h ($h \neq h_i$) in the N-best list of utterance $\mathbf{x}_i$

$$w(\mathbf{x}_i, h) = c_k^{\frac{1}{2}(1+P(h_i|\mathbf{x}_i)-P(h|\mathbf{x}_i))}$$

- Calculate new weight for each utterance $\mathbf{x}_i$

$$w_i^k = \sum_{h \neq h_i^*} w(\mathbf{x}_i, h)$$

- Resample training data according to normalized w_i^k, forming new training set $\mathbf{U}^k$.

In generalization, the hypothesis to a new utterance $\mathbf{x}$ is determined by $h^* = \arg\max_h \sum_{k=1}^{K} \log \frac{1}{c_k} P_{\lambda_k}(h|\mathbf{x})$, where h denoted the top-1 hypothesis of N-best list of each model.

3. N-Best List Re-Ranking

As we discussed in the previous section, it's impossible for a speech recognizer to output every possible hypothesis. In implementation, most systems only generate and output the most probable results ranked by their decoding scores or joint probabilities $P_\lambda(h,\mathbf{x})$, which is called the N-best list. Examination of the N-best list reveals that the best hypothesis, the one with the lowest word error rate, is not always in top-1 position. This phenomenon is due to many reasons, such as inaccurate acoustic and language models, unavailability of sufficient training data, and lack of good features.

N-best list re-ranking is a post-processing technique that attempts to locate the hypothesis with lowest word errors rather than accept the top-1 result blindly; its effectiveness has been shown in many independent experiments [8]. The

application of N-best re-ranking involves two aspects: the identification of useful features and the selection of an effective re-ranking technique. According to our experience in many classification tasks, good features usually play a more important role in creating a successful system.

Four features are investigated in our experiments, which represent information from four sources:

Language model feature

LM-Backoff-Mode. This is a language model related feature. For each word, the value of backoff mode is determined according to whether the 1, 2, or 3-gram is used to compute language model score. For a hypothesis, the feature value is set to the average value of every word.

Utterance level feature

Utterance level posterior probability $P_\lambda(h \mid \mathbf{x})$. This is a feature measuring the reliability of the whole hypothesis. One implementation scheme has been discussed in Section 2.

Word level feature

Word level posterior probability $P_\lambda(a \mid \mathbf{x})$ where a denotes a word in hypothesis [9]. This feature measures how likely a particular hypothesized word a is a correct recognition result. The value is computed from the word lattice or the N-best list by summing and normalizing the scores of paths passing through the word in question. The feature value for a hypothesis is set to the average value of words.

Frame level feature

Frame level posterior probability. Motivated by the utterance and word level features, we investigated a new family of features that estimates posterior probability for frame level hypothesis.

Let $\mathbf{x}$ be the sequence of cepstrum vectors for an utterance with *T* frames. ϕ_i is a variable to indicate possible word for frame *i*. Usually the value range for ϕ_i could be the entire vocabulary. In addition, we write a N-best list hypothesis in the format like $h = (a_1, a_2, ..., a_T)$ in which a_i is the hypothesized word at frame *i. Frame level posterior probability* is to estimate the probability that $P(\phi_i = a_i \mid \mathbf{x})$, whose value could be computed as follows.

$$\begin{aligned} & P(\phi_i = a_i \mid \mathbf{x}) \\ &= P(\phi_i = a_i, \mathbf{x}) / P(\mathbf{x}) \\ &= \sum_{\substack{h \in \text{n-best list of } \mathbf{x} \\ \text{and } \phi_i = a_i}} P(h, \mathbf{x})^\beta \Bigg/ \sum_{h \in \text{n-best list of } \mathbf{x}} P(h, \mathbf{x})^\beta \end{aligned} \tag{4}$$

where β is an empirically set smoothing parameter.

For a hypothesis in the N-best list, its feature value is computed by summing and normalizing frame probability, which is

$$f = \frac{1}{T} \sum_{i=1}^{T} P(\phi_i = a_i \mid \mathbf{x}) \tag{5}$$

This feature could be understood as the frame based word accuracy, and the correct hypothesis is expected to have higher value than others.

We use Neural Network as the re-ranking method. The inputs are the four features we discussed above, while the output is trained to approximate word accuracy of each hypothesis. After re-ranking, the hypothesis with the highest estimated word accuracy will be chosen as the best hypothesis for combination.

4. Rover Combination

The standard Boosting algorithm uses utterance level majority voting to combine hypotheses from each acoustic models. This method ignores some important information associated with individual words in the hypothesis, such as confidence and segmentation. Our previous research has show that the word level combination that integrates word information could improve recognition accuracy.

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 seek the best scoring word sequence. This is different from the majority voting adopted by the Boosting algorithm that only selects the most probable result from the existing top-1 hypotheses: ROVER can create a new hypothesis by merging two or more 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 Boosting algorithm even though all the acoustic models are trained using the same technique, albeit with different views of the corpus. Experimental results will be provided in the next section.

5. Experiments

5.1. 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. In our experiments, the number of acoustic models trained by Boosting algorithm is set to 4.

In a difference from our previously reported experiments which used semi-continuous HMM, we used the Carnegie Mellon Sphinx-3 decoder, which is a continuous HMM system, for training and test. 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 list from the word lattice. After that, we use Neural Network to re-rank N-best list to determine the most probable hypothesis. Furthermore, ROVER system is used to combine hypotheses selected from each model.

5.2. Experimental results

Three experiments are designed to test the effectiveness of N-best list re-ranking and ROVER to Boosting style acoustic modeling. The first one is the training and test of multiple acoustic models using conventional Boosting algorithm and utterance level combination method. The result obtained from this experiment is regarded as the baseline for following

experiments. Neural Network based N-best list re-ranking was tested in the second experiment, in which utterance combination is applied to reordered hypotheses. The last experiment is designed to test ROVER-based combination which is applied to the reordered hypotheses.

Table 1 shows the performance of Boosting training as a function of the number of acoustic models. Please note that the result for K=n means the word error rate obtained from hypotheses combination of n models. As we reported before, cooperation of multiple models outperforms single model on recognition accuracy. When use 4 acoustic models, the word error rate is down to 13.27% from 14.99%, the number for one model, which represents 11.47% relative reduction. Special attention should be paid to the last two columns which shows that there is no improvement when increase the model numbers from 3 to 4. This indicates that a small set of acoustic models may be sufficient for some speech recognition task, and adding too many models to the system may lead to overfitting.

# models	K=1	K=2	K=3	K=4
W.E.R	14.99%	13.54%	13.31%	13.27%

Table 1 Boosting training

Table 2 shows experimental results using N-best list re-ranking. Comparing Table 2 with Table 1, we can find that the re-ranking produced a consistent improvement in system performance. The results demonstrate the effectiveness of the Neural Network based re-ranking method, as well as the utility of the features we adopted in the experiments. As in experiment 1, we also notice that there is no great difference between K=3 and K=4.

# models	K=1	K=2	K=3	K=4
W.E.R	14.62%	13.14%	13.03%	12.98%

Table 2 Boosting training + N-best list re-ranking

Building on N-best list re-ranking, we further investigate the performance of ROVER combination. Table 3 shows the result. The performance demonstrates that ROVER outperforms conventional method in combining multiple hypotheses. This strongly supports the view that word level combination is more suitable for speech recognition than utterance level combination. The final result is very encouraging. When we use 4 acoustic models, along with the N-best list re-ranking and ROVER, the word error rate is reduced to 12.52% which represents 16.5% relative reduction compared to the performance of single model. One phenomenon deserving more attention is that when K=2 the word error rate achieved by ROVER is higher than that given by utterance combination. (See Table 2.) We think this could be explained by a characteristic of ROVER which essentially disposes it to select words with a high confidence score and frequency. In the case of two hypothesis combination, the frequency information isn't very helpful. Therefore, if the confidence measure isn't accurate, ROVER may be unable to generate correct result.

# models	K=1	K=2	K=3	K=4
W.E.R	14.62%	13.32%	12.70%	12.52%

Table 3 Boosting training + N-best list re-ranking + ROVER

6. Conclusions

We investigated the use of Boosting for acoustic model training, along with N-best list re-ranking and ROVER hypothesis combination. As we proposed, there is considerable mismatch between the training criterion of the Boosting algorithm and commonly used performance metrics. The goal of the work we report is to fill the gap using post-processing techniques. Encouraging improvement is observed in our experiments. When the three methods, Boosting, re-ranking and ROVER are used together, our system achieves 16.5% relative reduction in word error rate.

Acknowledgement

This work was supported under DARPA grant NBCH-D-03-0010. 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.

References

[1] R. E. Schapire, "A brief Introduction to Boosting", Proc. of the 16th International Joint Conference on Artificial Intelligence, 1999.

[2] H. Schwenk, "Using Boosting to Improve A Hybrid HMM/Neural Network Speech Recognizer", Proc. of ICASSP 1999.

[3] P. Moreno, B. Logan and B. Raj, "A Boosting Approach for Confidence Scoring", Proc. of EuroSpeech 2001.

[4] S-W Foo and E-G Lim, "Speaker Recognition Using Adaptively Boosted Decision Tree Classifier", Proc. of ICASSP 2002.

[5] C. Meyer, "Utterance-Level Boosting of HMM Speech Recognizers", Proc. of ICASSP 2002.

[6] R. Zhang and A. I. Rudnicky, "Comparative Study of Boosting and Non-Boosting Training for Constructing Ensembles of Acoustic Models", Proc. of Eurospeech 2003.

[7] J. G. Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)", Proc. of ASRU 1997.

[8] A. Chotimongcol and A. I. Rudnicky, "N-best Speech Hypotheses Reordering Using Linear Regression", Proc. of Eurospeech 2001.

[9] F. Wessel, K. Macherey and R. Schluter, "Using Word Probabilities as Confidence Measures", Proc. of ICASSP 1998.

[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.