IMPROVING THE PERFORMANCE OF AN LVCSR SYSTEM THROUGH ENSEMBLES OF ACOUSTIC MODELS

Rong Zhang and Alexander I. Rudnicky
School of Computer Science, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{rongz, air@cs.cmu.edu}

ABSTRACT

This paper describes our work on applying ensembles of acoustic models to the problem of large vocabulary continuous speech recognition (LVCSR). We propose three algorithms for constructing ensembles. The first two have their roots in bagging algorithms; however, instead of randomly sampling examples our algorithms construct training sets based on the word error rate. The third one is a boosting style algorithm. Different from other boosting methods which demand large resources for computation and storage, our method present a more efficient solution suitable for acoustic model training. We also investigate a method that seeks optimal combination for models. We report experimental results on a large real world corpus collected from the Carnegie Mellon Communicator dialog system. Significant improvements on system performance are observed in that up to 15.56% relative reduction on word error rate is achieved.

1. INTRODUCTION

Using ensembles of classifiers to improve the accuracy of supervised learning has received increasing attention in recent years [1]. An ensemble of classifiers is a collection of single classifiers which is used to select a hypothesis based on the majority vote from its components. Bagging [2] and boosting [3] are the two most successful algorithms for constructing ensembles. The application of such techniques to speech recognition is at an early stage but appears to be highly promising.

Bagging constructs ensembles in a straightforward way. In each round, bagging randomly selects a number of examples from the original training set, and produces a new single classifier based on the selected subset. The final classifier is built by choosing the hypothesis best agreed on by single classifiers.

In boosting, the single classifiers are iteratively trained in a fashion such that hard-to-classify examples are given increasing emphasis. More specifically, a probability distribution is maintained for the training data, and initially every example is assigned 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 probability distribution is then updated to increase the weight of incorrectly classified examples. As a result those examples that are difficult to classify will receive more attention from subsequent classifiers. In generalization, the final hypothesis is the weighted majority vote from the single classifiers.

Algorithms similar to bagging have been applied to acoustic model training. For supervised training, [4] proposed a method that emphasizes the "bad" utterance that has high word error rate or low confidence score. On the contrary, for unsupervised training, [5] suggested focusing on "good" data, measured by high confidence score.

Boosting has also been used to solve problems in speech recognition, such as phoneme recognition [6], confidence annotation [7] and speaker identification [8]. However, the complexity of these problems doesn't exceed the level of standard multi-class classification. [9] is probably the first approach that presents a feasible boosting solution for continuous speech recognition. In [9], the hypothesis space is compressed to the N-best lists, and "a posteriori" probability is calculated for each list. The AdaBoost.M2 algorithm is then used. Although its feasibility is verified by experiments on word recognition, this method raises the concern that the computation of word lattice and N-best lists may be an unbearable cost for a large continuous speech corpus. Moreover, the decoding score based on "a posteriori" probability isn't likely to be a good estimation since the decoding score could vary over a wide range.

The familiar gender-based speech recognition technique, that uses separate models for male and female speakers, can be regarded as a variant of ensemble of classifiers. Other successful examples include multi-band acoustic modeling [10].

In applying ensemble of models techniques to acoustic model training, one has to bear in mind the characteristics of large vocabulary continuous speech recognition (LVCSR) decoding. Specifically, LVCSR is more complicated than multi-class classification. First, it's difficult to create an accurate segmentation for phoneme and word from continuous speech. So the acoustic model has to be trained on the utterance level. Second, the number of decoding hypotheses for an utterance could be infinite. Assume that the decoder has 5,000 words in its vocabulary, and that the maximum length for utterance is confined to 20 words. Theoretically, the decoder could output up to 5000^{20} different hypotheses. Third, given present computer technology it's difficult to assimilate the cost of combining hundreds of acoustic models. According to our experience, the number of acoustic models that can be accommodated by a real-time LVCSR system would be in the single digits. Additionally, the training data for acoustic model usually contains a large amount of noise. For such noisy data, the standard boosting approach could perform very poorly.

Our research aims to develop efficient and effective algorithms for constructing ensembles that are suitable for LVCSR systems. In this paper we propose three algorithms and implemented these on a large real-world corpus. The first two are inspired by bagging. We differ from [4] in that the goal of our algorithms is to construct ensembles rather than a single (optimal) model. The third algorithm attempts to present an efficient boosting solution for acoustic model training without the need for generating word lattice and N-best lists. We also investigate a method that provides optimal combination for models. Experimental results show significant improvement on system performance, up to 15.56% relative reduction of word error rate is achieved.

2. ALGORITHMS

Some notations used in this paper are introduced here.

- $\mathbf{X}$: The training set.
- $\mathbf{x}_i$: The *i*-th training utterance that $1 \le i \le |\mathbf{X}|$.
- $\mathbf{X}^t$: The re-sampled training set in round *t*.
- *T*: The number of models in ensemble.
- w_i^t : The weight assigned to $\mathbf{x}_i$ in round *t*.
- $\mathbf{A}^t$: The acoustic model trained in round *t*.
- c_t : The weight assigned to $\mathbf{A}^t$.
- ε_i^t : The word error rate of utterance $\mathbf{x}_i$ under model $\mathbf{A}^t$. Its maximum value is set to 1.
- $h^t(\mathbf{x})$: The hypothesis for utterance $\mathbf{x}$ generate by model $\mathbf{A}^t$.
- $s^t(\mathbf{x})$: The decoding score for utterance $\mathbf{x}$ generated by model $\mathbf{A}^t$.

2.1. Algorithm 1

The first algorithm that we investigated is based on the intuition that an incorrectly recognized utterance should receive more attention in training. We use word error rate to measure how difficult an utterance is to recognize, and associate the weight of each training utterance with this metric. Figure 1 shows the algorithm in detail.

Figure1: Algorithm 1

```
Initialize:
  • Let $\mathbf{X}^0 = \mathbf{X}$.
  • Assign equal weight to each utterance $\mathbf{x}_i$ that $w_i^0 = 1$.
For t = 1 to T:
  • Train new acoustic model $\mathbf{A}^t$ from data set $\mathbf{X}^{t-1}$.
  • Test model $\mathbf{A}^t$ on the initial training set $\mathbf{X}$, computing
    word error rate $\varepsilon_i^t$ for each utterance $\mathbf{x}_i$.
  • Update distribution $w_i^t = w_i^{t-1}(1 + \lambda \varepsilon_i^t)$.
  • Resample training data according to $w_i^t$, forming new
    training set $\mathbf{X}^t$.
```

The resampling of training data is executed as in the following example: if the weight of an utterance is 2.6, we first add two copies of the utterance to the new training set, and then add its third copy with probability 0.6.

Different from some bagging and boosting methods, Algorithm 1 does not get rid of the correctly recognized utterances, since their weights are at least 1. This choice is based on the observation that acoustic models are usually improved by incorporating more data. Nevertheless, to prevent the training set from being too large, parameter λ is used to soften the weight.

In generalization, the final hypothesis $h(\mathbf{x})$ for a new utterance $\mathbf{x}$ is determined in such a way that $h(\mathbf{x}) = h^t(\mathbf{x})$ if $s^t(\mathbf{x}) = \max\{s^1(\mathbf{x}), s^2(\mathbf{x}), \ldots, s^T(\mathbf{x})\}$.

2.2. Algorithm 2

The exponential increase in the size of training set is a severe problem for algorithm 1, especially when *T* is large or word error rate is high. Algorithm 2 is proposed to address this problem, and is shown in Figure 2.

Figure 2: Algorithm 2

```
Initialize:
  • Let $\mathbf{X}^0 = \mathbf{X}$.
  • Assign equal weight to each utterance $\mathbf{x}_i$ that $w_i^0 = 1$.
For t = 1 to T:
  • Train new acoustic model $\mathbf{A}^t$ from data set $\mathbf{X}^{t-1}$.
  • Test model $\mathbf{A}^t$ on the initial training set $\mathbf{X}$, computing
    word error rate $\varepsilon_i^t$ for each utterance $\mathbf{x}_i$ and recoding the
    decoding score $s^t(\mathbf{x}_i)$.
  • Determine the hypothesis for $\mathbf{x}_i$ by selecting the "best"
    model from $\mathbf{A}^1$, $\mathbf{A}^2$,…, $\mathbf{A}^t$:
    Let $h(\mathbf{x}_i) = h^j(\mathbf{x}_i)$
    if $s^j(\mathbf{x}_i) = \max\{s^1(\mathbf{x}_i), s^2(\mathbf{x}_i), \ldots, s^t(\mathbf{x}_i)\}$.
  • Compute word error rate $\varepsilon_i$ for $h(\mathbf{x}_i)$.
  • Update distribution $w_i^t = 1 + \varepsilon_i$.
  • Resample training data according to $w_i^t$, forming new
    training set $\mathbf{X}^t$.
```

Apparently, the weight w_i^t is always within the range [1,2], that guarantees the size of train set is maintained on an acceptable level. In generalization, the hypothesis $h(\mathbf{x})$ for a new utterance $\mathbf{x}$ is determined in the same way as algorithm 1.

2.3. Algorithm 3

In Algorithms 1 and 2, there is no concern to measure how important a model is relative to others. All of the models have equal weight, while intuitively, good model should play more important role than bad one. Algorithm 3 presents a boosting-style solution that incorporate a parameter c_t to describe the difference in importance between models. The algorithm is based on the following cost function.

$$L=\sum_{i=1}^{|\mathbf{X}|}\exp\Big(-\sum_{t=1}^{T}c_t e_t(\mathbf{x}_i)\Big) \tag{1}$$

where

$$e_t(\mathbf{x}_i)=\begin{cases}\alpha & if\ \varepsilon_i^t=0\\ -\varepsilon_i^t & otherwise\end{cases} \tag{2}$$

$e_t(\mathbf{x}_i)$ can be understood as the outside feedback to model $\mathbf{A}^t$. The incorrect recognition is penalized by the word error rate ε_i^t, while correct recognition is prized with the parameter α which value is empirically set.

Figure 3: Algorithm 3

```
Initialize:
  • Let $\mathbf{X}^0=\mathbf{X}$.
  • Assign equal weight to each utterance $\mathbf{x}_i$ that $w_i^0=1$.
For t = 1 to T:
  • Train new acoustic model $\mathbf{A}^t$ from data set $\mathbf{X}^{t-1}$.
  • Test model $\mathbf{A}^t$ on the initial training set $\mathbf{X}$, computing word error rate $\varepsilon_i^t$ for each utterance $\mathbf{x}_i$.
  • Determine the weight $c_t$ for model $\mathbf{A}^t$ through linear search that minimizes the cost function
        $L=\sum_{i=1}^{|\mathbf{X}|} w_i^{t-1}\exp(-c_t e_t(\mathbf{x}_i))$
  • Update distribution
        $w_i^t=\begin{cases} w_i^{t-1} & if\ \varepsilon_i^t=0\\ w_i^{t-1}\exp(c_t(\varepsilon_i^t+\alpha)) & otherwise\end{cases}$
  • Resample training data according to $w_i^t$, forming new training set $\mathbf{X}^t$.
```

(1) can be rewritten as the following form.

$$\begin{aligned}L&=\sum_{i=1}^{|\mathbf{X}|}\exp\Big(-\sum_{t=1}^{T-1}c_t e_t(\mathbf{x}_i)\Big)\exp(-c_T e_T(\mathbf{x}_i))\\ &=\sum_{i=1}^{|\mathbf{X}|}w_i^{T-1}\exp(-c_T e_T(\mathbf{x}_i))\end{aligned} \tag{3}$$

(3) suggests that the cost function can be minimized in an iterative way, resulting in Algorithm 3 shown in Figure 3.

Theoretically, Algorithm 3 determines the final hypothesis in such a way that

$$h(\mathbf{x})=\arg\max_y\Big\{\sum_{t=1}^{T}c_t\delta(h^t(\mathbf{x})=y)\Big\} \tag{4}$$

where $\delta(Q)$ is 1 if Q is true and 0 otherwise. However, this method did not work very well in our experiments, so a more effective solution is discussed in section 3.

3. METHOD FOR COMBINING MODELS

In Algorithm 1 and 2, hypothesis is selected by comparing their decoding score $s^t(\mathbf{x})$. We further associate $s^t(\mathbf{x})$ with c_t, the parameter describing the importance of model $\mathbf{A}^t$, forming weighted decoding score $c_t s^t(\mathbf{x})$. $c_t s^t(\mathbf{x})$ can be regarded as the approximation for the weighted probability $P(h^t(\mathbf{x})\,|\,\mathbf{x})^{c_t}$ since $s^t(\mathbf{x})$ is the log-likelihood for $P(h^t(\mathbf{x}),\mathbf{x})$. Now the hypothesis is selected by choosing the "best" model with the highest weighted decoding score that $c_t s^t(\mathbf{x})=\max\{c_1 s^1(\mathbf{x}),c_2 s^2(\mathbf{x}),\ldots\ldots,c_T s^T(\mathbf{x})\}$. The optimal value of c_t is determined by minimizing the recognition error on the whole training corpus. Linear search or "hill climbing" are two methods that can be used to find the optimal value.

4. EXPERIMENTS

4.1. Data Set and Configuration

The data set used in our experiment was collected using the CMU Communicator system, a telephone based dialog system that supports planning in a travel domain [11]. The training set has 89,735 utterances, which were collected from April 1998 to November 2000. The test set consists of 1,689 utterances, which were collected from a NIST evaluation during July 2000. All of our experiments, both training and decoding, are performed using the Carnegie Mellon Sphinx-2 system [12]. There are 9,769 words in the vocabulary.

In our experiments, the number of models (*T* in each algorithm) is set to 6. The parameter λ in Algorithm 1 is set to 0.6 and the parameter α in Algorithm 3 is set to 0.65, both of which are empirical selected to impose restriction on training size.

The baseline recognition word error rate in our experiments is 27.06% and 20.42%, for test and training sets respectively.

4.2. Experimental Results

In our experiments, all three algorithms demonstrated significant improvements over baseline. Table 1 shows the word error rates that are achieved by the three algorithms. For the test set, all of them relatively reduced the word error rates by at least 10%; the best one, Algorithm 1, realized a relative reduction of 15.56%.

Algorithm	Training Error	Relative Reduction	Test Error	Relative Reduction
A. 1	18.19%	10.92%	22.85%	15.56%
A. 2	18.51%	9.35%	23.89%	11.71%
A. 3	18.09%	11.41%	23.17%	14.38%

Table 1 Final Performance of Ensembles

More details can be found in Figure 4 and 5, which show algorithms performance as a function of *T* (the number of acoustic models in the ensemble). Baseline is at *T* = 1.

Algorithm 1 and 3 exhibit the most consistent downtrends for word error rate. Moreover, from their curves, we can expect additional reduction with increasing *T*. However, the decline tendencies are deteriorated after *T* = 4 for all the three algorithms, especially on training set. Another notable phenomenon is that the best algorithm on test set is not the best one on training set. This may suggest that Algorithm 3 is susceptible to overfitting.

Table 2 shows the size of training set used in each round. Obviously, Algorithm 1 suffers the problem of exponential increase of training size. The number of training utterances is more than doubled when T = 6. Meantime, the training sizes for Algorithm 2 and 3 remain at affordable level. The decline of the number of training utterances for Algorithm 2 is due to its overall decrease on training error. It's may be surprising to see the size of training set for Algorithm 3, whose cost function has the exponential form, doesn't increase too much. This can be explained by the small value of the parameter c_t. We observed in the experiments that c_t is within [0.01, 0.1] when $t > 2$.

Alg.	T=1	T=2	T=3	T=4	T=5	T=6
A. 1	89735	98280	110287	130314	160945	201962
A. 2	89735	101050	99792	99369	99213	99146
A. 3	89735	112777	118921	121236	123748	124884

Table 2 Training Size in each round

5. DISCUSSION

We described three bagging and boosting style algorithms for constructing ensembles of acoustic models for continuous speech recognition. All three of these algorithms achieve significant improvements on system performance. The relative reductions of word error rate on the test set are 15.56%, 11.71% and 14.38% respectively. The results illustrate the potential of this approach as a meaningful method for improving the quality of acoustic models trained from a fixed corpus.

Several issues are currently under investigation which are expected to lead to further improvement. First, even though the three algorithms that we studied have shown good performance in our experiments, it's still hard to say they are the best approaches to construct ensembles. So the main task of our research is to seek more suitable approach for acoustic model training. For example, we'd like to improve the cost function defined in Algorithm 3 to incorporate more information. Second, we think the combination methods also deserve further study, especially the post-optimization techniques. Additionally, current computer technology doesn't allow us to combine large number of acoustic models in a real time system, so a possible way that could benefit from more models is to select a small number of "strong" models from a larger pool. Obviously, this brings us the issues such as how to build the larger pool and how to select and combine the strongest model.

6. 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.

7. REFERENCES

[1] T. G. Dietterich, "Machine Learning Research: Four Current Directions", AI Magazine, 18(4): 97-136, 1998.

[2] L. Breiman, "Bagging Predictors", Machine Learning, 24(2): 123-140, 1996.

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

[4] T. Kamm and G. Meyer, "Automatic Selection of Transcribed Training Material", Proc. of 7th IEEE Workshop on Automatic Speech Recognition and Understanding, 2001.

[5] T. Kemp and A. Waibel, "Unsupervised Training of A Speech Recognizer: Recent Experiments", Proc. of EuroSpeech 1999.

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

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

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

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

[10] A. Hagen, H. Bourlard, and A. Morris, "Adaptive ML-Weighting in Multi-Band Recombination of Gaussian Mixture ASR," Proc. of ICASSP 2001.

[11] A. 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.

[12] 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.

Figure 4 Performance on Test Set

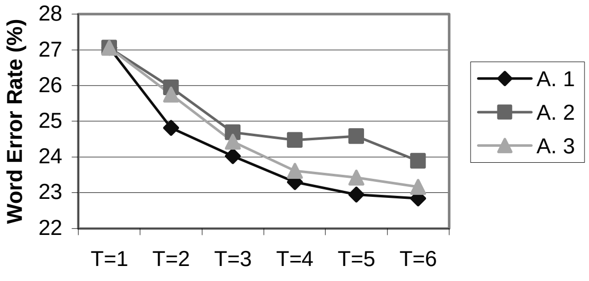
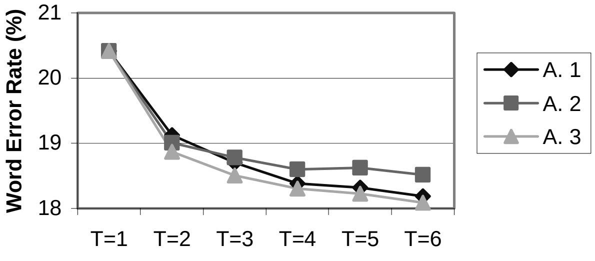


Figure 5 Performance on Training Set


Word Error Rate (%)
T=1
T=2
T=3
T=4
T=5
T=6
A. 1
A. 2
A. 3