INTERSPEECH 2009 BRIGHTON

A Study of Bootstrapping with Multiple Acoustic Features for Improved Automatic Speech Recognition

Xiaodong Cui, Jian Xue, Bing Xiang and Bowen Zhou

IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598, USA

{cuix,jxue,bxiang,zhou}@us.ibm.com

10.21437/Interspeech.2009-85

Abstract

This paper investigates a scheme of bootstrapping with multiple acoustic features (MFCC, PLP and LPCC) to improve the overall performance of automatic speech recognition. In this scheme, a Gaussian mixture distribution is estimated for each type of feature resampled in each HMM state by single-pass re-training on a shared decision tree. Thus obtained acoustic models based on the multiple features are combined by likelihood averaging during decoding. Experiments on large vocabulary spontaneous speech recognition show its superior overall performance than the best of acoustic models from individual features. It also achieves comparable performance to Recognizer Output Voting Error Reduction (ROVER) with computational advantages.

Index Terms: feature bootstrap, multiple acoustic features, spontaneous speech recognition

1. Introduction

Statistical model combination aims to combine simple or unreliable models to approximate a complex system in a more reliable way. It has been extensively studied in the machine learning community [1]. In recent years, there has been a growing interest in introducing model combination techniques into large vocabulary continuous speech recognition (LVCSR) [2, 3, 4, 5]. Among them, random forest [6] on phonetic decision trees [2, 3, 4] and ROVER on the ASR outputs [5] are most notable. Construction of phonetic decision trees following the random forest theory seeks to randomize the questions for splitting the node when growing the decision tree and the states tied this way result in complementary acoustic models. The ROVER techniques combine decoding outputs from a variety of ASR systems and generate a single hypothesis via majority voting. In both cases, by combining complementary ASR systems, the approaches help to improve the overall performance across individual systems.

In this paper, we investigate a model combination scheme by bootstrapping a variety of acoustic features in each HMM state. The features chosen for bootstrapping are MFCC, PLP and LPCC, respectively. A GMM is estimated for each type of resampled feature in the state by single-pass re-training so that the state has more than one distribution to describe the acoustic characteristics under the multiple features. This differs from how multi-stream models are built [7, 8]. The GMMs are combined by averaging the corresponding likelihood scores in that state during decoding. It is hoped that the features can be complementary to give rise to a more accurate modeling of the acoustics in each state and therefore lead to better overall performance. The weights of distributions from each type of feature can be simply equal or distinct whose maximum likelihood estimation is conducted on the state level. Instead of building individual ASR systems with different features from scratch, the acoustic model combined this way has only one decision tree and the feature resampling is achieved by single-pass re-training. Hence, compared to random forest and ROVER, this feature bootstrapping scheme gives computational advantages since it does not require multiple decoding.

The remainder of the paper is organized as follows. In Section 2, we give the mathematical formulation of the feature bootstrapping approach and the maximum likelihood estimation of the state-dependent weights of each individual feature distribution. Experimental results on large vocabulary spontaneous speech are presented in Section 3 and a discussion and summary are provided in Section 4.

2. Mathematical Formulation

2.1. Feature Bootstrapping

Suppose there are L types of features available. Let f, $f \in \{1, \cdots, L\}$, stand for a particular feature, $\boldsymbol{o}_t^f$ for the feature observation at time t and $(\cdot)^f$ for the statistics related to that particular feature type.

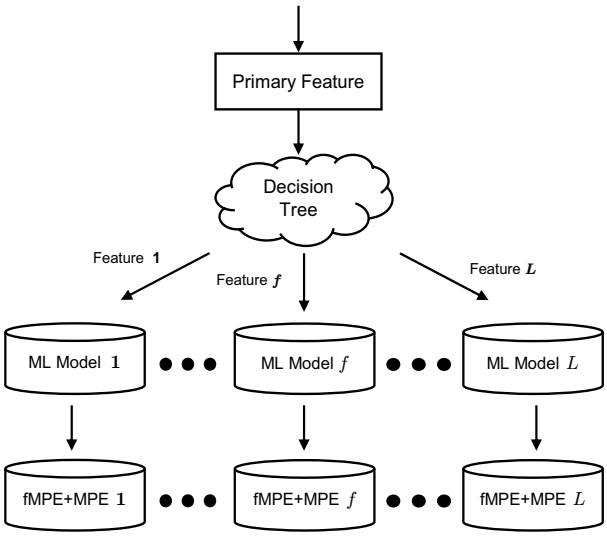

Figure 1: Diagram of generating acoustic models with bootstrapped features.

Fig.1 demonstrates the process of generating acoustic models with bootstrapped features. Among the L features, a primary feature m is first selected to build a decision tree. Then an

HMM model with a GMM distribution in each state is trained with this feature under the ML criterion with the EM algorithm [10]. Given the decision tree and HMM, for state i and Gaussian component k, the mean and covariance of the Gaussian distribution for each feature f in $\{1, \cdots, L\}$, is estimated as

$$\boldsymbol{\mu}_{ik}^{f} = \frac{\sum_{t=1}^{T} \gamma_{ik}^{m}(t) \boldsymbol{o}_t^f}{\sum_{t=1}^{T} \gamma_{ik}^{m}(t)} \quad (1)$$

$$\boldsymbol{\Sigma}_{ik}^{f} = \frac{\sum_{t=1}^{T} \gamma_{ik}^{m}(t) (\boldsymbol{o}_t^f - \boldsymbol{\mu}_{ik}^{f})(\boldsymbol{o}_t^f - \boldsymbol{\mu}_{ik}^{f})^{\mathsf{T}}}{\sum_{t=1}^{T} \gamma_{ik}^{m}(t)} \quad (2)$$

where $\gamma_{ik}^{m}(t)$ is the posterior probability of the primary feature being at state i Gaussian component k at time t and T the total number of frames.

Suppose there are M_i Gaussians in the GMM distribution in state i. Given $\boldsymbol{\mu}_{ik}^{f}$ and $\boldsymbol{\Sigma}_{ik}^{f}$, the weight of Gaussian component k for feature f, $c_{ik}^{f},\ k \in \{1, \ldots, M_i\}$, is obtained as

$$c_{ik}^{f} = \frac{\sum_{t=1}^{T} \gamma_{ik}^{f}(t)}{\sum_{t=1}^{T} \sum_{l=1}^{M_i} \gamma_{il}^{f}(t)}$$

where

$$\begin{aligned} \gamma_{ik}^{f}(t) &= \gamma_{i}^{m}(t) \cdot \gamma_{k|i}^{f}(t) \\ &= \gamma_{i}^{m}(t) \cdot \frac{c_{ik}^{m} \mathcal{N}(\boldsymbol{o}_t^f; \boldsymbol{\mu}_{ik}^{f}, \boldsymbol{\Sigma}_{ik}^{f})}{\sum_{l=1}^{M_i} c_{il}^{m} \mathcal{N}(\boldsymbol{o}_t^f; \boldsymbol{\mu}_{il}^{f}, \boldsymbol{\Sigma}_{il}^{f})} \end{aligned} \quad (3)$$

From the above estimates of $\boldsymbol{\mu}_{ik}^{f}$ (Eq.1), $\boldsymbol{\Sigma}_{ik}^{f}$ (Eq.2) and c_{ik}^{f} (Eq.3), one can tell that the estimates of the model parameters of all the L features share the same posterior probability of a particular state i based on the primary feature, $\gamma_{i}^{m}(t)$. This is equivalent to indicating that, given time t, the feature observations $\boldsymbol{o}_t^f$ are all softly aligned to the same state for all f. In other words, the GMMs estimated this way can be considered as distributions by resampling from state i with different types of features. Fig.2 illustrates as an example the GMMs estimated in a particular HMM state by resampling MFCC, PLP and LPCC features.

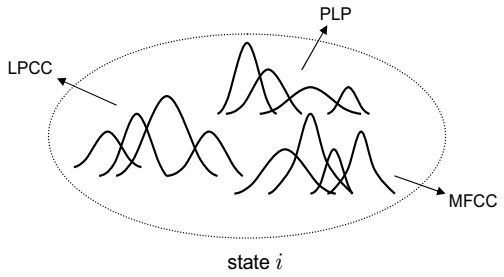

Figure 2: An illustration of resampling of a state with multiple features.

Once the ML models are trained for the L types of features, discriminative training in both feature space (fMPE) [11] and model space (MPE) [12] are performed to obtain the final models for decoding.

2.2. Model Combination

Assume all of the L features have been sampled for each individual HMM state and the corresponding GMM distribution is estimated according to Eqs.1, 2 and 3. In the decoding stage, they contribute to the final likelihood of the state i by averaging across the L respective GMM distributions as shown in Eq.4,

$$b_i(\boldsymbol{o}_t) = \exp\left[\sum_{f=1}^{L} \alpha_{if} \log b_{if}(\boldsymbol{o}_t^f | \boldsymbol{\theta}^f)\right] \quad (4)$$

where $b_{if}(\boldsymbol{o}_t^f | \boldsymbol{\theta}^f)$ is the GMM distribution for feature f

$$b_{if}(\boldsymbol{o}_t^f | \boldsymbol{\theta}^f) = \sum_{k=1}^{M_i} c_{ik}^{f} \mathcal{N}(\boldsymbol{o}_t^f; \boldsymbol{\mu}_{ik}^{f}, \boldsymbol{\Sigma}_{ik}^{f}) \quad (5)$$

and

$$\sum_{f=1}^{L} \alpha_{if} = 1 \quad (6)$$

$$0 \leq \alpha_{if} \leq 1 \quad (7)$$

are the weights of the distributions of the L features in state i.

A straightforward way of choosing α_{if} is to globally tie and equally weight the log-likelihoods $\log b_{if}(\boldsymbol{o}_t^f | \boldsymbol{\theta}^f)$ from each feature f in which case Eq.4 turns to

$$b_i(\boldsymbol{o}_t) = \exp\left[\frac{1}{L}\sum_{f=1}^{L} \log b_{if}(\boldsymbol{o}_t^f | \boldsymbol{\theta}^f)\right] \quad (8)$$

where the final log-likelihood $\log b_i(\boldsymbol{o}_t)$ is simply the arithmetic mean of the log-likelihoods $\log b_{if}(\boldsymbol{o}_t^f | \boldsymbol{\theta}^f)$ from the L features.

In a general case, α_{if} is state dependent and shall be estimated. However, direct ML estimation of α_{if} under the constraints Eq.6 and Eq.7 may pose problem since in this case the final log-likelihood $\log b_i(\boldsymbol{o}_t)$ with the Lagrange multiplier is a linear function of α_{if} and it only yields its maximum at the constraint boundary. It amounts to setting a particular α_{if} to 1 and the others to 0, equivalent to selecting only one type feature for the state. Therefore, we resort to the L_p ($p \geq 2$) norm constraint on the weights as shown in Eq.9 which has been used in previous works such as [8] and [9].

$$\sum_{f=1}^{L} \alpha_{if}^{p} = K, \quad p \geq 2 \quad (9)$$

Following the EM algorithm [10] with the constraint imposed by Eq.9, the Q-function of the E-step is given by

$$Q(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}) = \sum_{t=1}^{T} \sum_{i} \gamma_i(t) \log b_i(\boldsymbol{o}_t) + \sum_{i} \lambda_i \left(\sum_{f=1}^{L} \alpha_{if}^{p} - K\right)$$

Substituting Eq.4 into $\log b_i(\boldsymbol{o}_t)$ and taking the derivative with respect to α_{if}, we have

$$\frac{\partial Q(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})}{\partial \alpha_{if}} = \sum_{t=1}^{T} \gamma_i(t) \log b_{if}(\boldsymbol{o}_t^f | \boldsymbol{\theta}^f) + p\lambda_i \alpha_{if}^{p-1}$$

Setting the derivative to zero and solving the equation, we get

$$\alpha_{if}^{p} = K \cdot \frac{\left[-\sum_{t=1}^{T} \gamma_i(t) \log b_{if}(\boldsymbol{o}_t^f)\right]^{\frac{p}{p-1}}}{\sum_{j=1}^{L} \left[-\sum_{t=1}^{T} \gamma_i(t) \log b_{ij}(\boldsymbol{o}_t^j)\right]^{\frac{p}{p-1}}} \quad (10)$$

Note that, $\gamma_i(t)$, the posterior probability of being state i at time t, can no longer depend on the primary feature. A reasonable way of starting the EM estimation in Eq.10 is to seed with the averaging strategy in Eq.8 and $\gamma_i(t)$ can be computed accordingly.

3. Experimental Results

3.1. Feature Space

Three acoustic features are chosen in this paper, namely, MFCC, PLP and LPCC. They are among the most widely used features in the speech recognition community. In spite of certain degree of similarity in the extraction process (e.g. both MFCC and PLP exploit Mel-frequency warping while both PLP and LPCC are based on linear prediction), they can roughly be considered "independent" with each other and therefore have complementary acoustic characteristics. The choice of the features are certainly not limited to those three types of features. In general, these features should be complementary and have comparable performance.

The construction of the feature space is shown in Fig.3. As indicated by the pipeline in the figure, the fundamental features are extracted with 24 dimensional MFCC, 19 dimensional PLP and 19 dimensional LPCC. Cepstral mean is removed at the utterance level. The LDA projection maps a splice of 9 frames of the cepstral mean normalized fundamental features to a 40 dimensional space in which feature space discriminative training fMPE [11] is performed. In the fMPE training, an improved training strategy discussed in [13] is employed. In this improved training configuration, the Gaussian cluster for the computation of posteriors is composed of 1024 Gaussians. The posterior from the nth Gaussian in the cluster is supplemented for project with the scaled offset of the LDA feature from the mean of the nth Gaussian and normalized by its standard deviation. Furthermore, a two-layered hierarchical projection is also employed for computational efficiency.

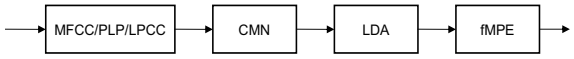

Figure 3: Pipeline of the construction of the feature space.

3.2. Training and Decoding Setup

There are in total of 240 hours spontaneous English speech in the training set. MFCC is chosen as the primary feature to grow the decision tree and build a well-trained ML model. The ML model consists of 7K quinphone states as the leaves of the decision tree and 100K Gaussians with diagonal covariance in the acoustic model. After that, all the three features run the single-pass re-training to estimate their ML models according to Eqs.1, 2 and 3 in Section 2. Therefore, all the three ML models share the same decision tree (therefore the same quinphone states) and have the same number of Gaussians in each state but with distinct sets of weights. Based on the ML models, 4 iterations of fMPE training followed by another 4 iterations of MPE training are applied to each type of feature.

Decoding is carried out by a Viterbi decoder on a finite state graph in which language model, dictionary and decision tree are hierarchically compiled. This graph configuration is able to provide a very fast decoding speed. The language model is trigram with a dictionary of 43K words. Since all the features share the same decision tree, they can be decoded with the same graph.

The performance is evaluated on three test sets which are referred to as Sets A, B and C, respectively. Set A has 651 utterances (0.6 hour) from 15 speakers. Set B has 495 utterances (0.5 hour) from 16 speakers. Set C has 2422 utterances (2.3 hours) from 29 speakers. All the three test sets are spontaneous speech collected by DARPA Transtac project for speech-to-speech translation.

3.3. Performance

As mentioned earlier, distinct from conventional model combination on independent systems, the feature bootstrapping investigated in this paper is realized via single-pass re-training on a decision tree followed by ML and discriminative training.

First of all, Table 1 compares the word error rates (WERs) on the three test sets by the discriminatively trained models (fMPE and MPE) built from the three features, denoted by λ_{MFCC}, λ_{PLP} and λ_{LPCC}, with independent as well as shared decision trees. The feature used for growing the shared decision tree is MFCC. The independent models from the independent trees have approximately the same number of quinphone states (7K) and Gaussians (100K). From this table, it can be observed that models with shared decision tree have no significant difference from those independent models in terms of performance. However, the decision tree sharing is able to provide a way of avoiding multiple decoding.

independent tree	Set A	Set B	Set C
λ_{MFCC}	14.9	14.5	16.2
λ_{PLP}	15.1	15.1	15.9
λ_{LPCC}	18.8	17.5	16.6
shared tree (MFCC)	Set A	Set B	Set C
λ_{MFCC}	14.9	14.5	16.2
λ_{PLP}	15.1	15.0	16.2
λ_{LPCC}	18.3	17.2	16.8

Table 1: Word error rate (WER) of discriminatively trained acoustic models (fMPE and MPE) on Sets A, B and C with independent and shared decision trees.

Table 2 shows the WERs of discriminatively trained acoustic models based on the scheme of shared decision tree in accordance with Fig.1. The first three rows are from fMPE and MPE models with MFCC, PLP and LPCC features, respectively. They are simply the repetition of the second half of Table 1 for the purpose of comparison in this table with other feature combination algorithms. The MFCC and PLP yield roughly comparable performance on the test sets with MFCC slightly better and both are superior than LPCC across all the test sets.

We first conduct a system combination at the hypotheses level with ROVER [5] for a comparison. The decoding output from each of the three individual models are combined into a single word transition network. The network is created with NIST's SCLITE tool based on dynamic programming alignment. We use the hypotheses from the model trained with MFCC as the reference or the base on which the composite word transition network is developed. Once the network is created, the combination tool evaluates each branching point using a voting scheme to select the best scoring words for the final transcription. The scoring is conducted by voting on the occurrence frequency of the words that share the same starting and ending states in the word transition network. As shown in Table

2, ROVER provides consistent gain compared to the best single output. However, each test set has to be decoded three times before ROVER is performed. This brings additional significant computational cost compared to the feature bootstrapping approach proposed in this work.

The last three rows in the table are WERs by feature bootstrapping and model combination discussed in Section 2 with discriminative training, denoted by λ_{BS}. The way of combing the models is shown in the parentheses where (global, equal) stands for globally tied and equal weights for each type of feature as described by Eq.8 and (state,p,K) for state dependent weights with L_p constraints defined by Eq.9. The state dependent weights are iteratively estimated with 2 iterations starting with the globally tied and equal weights which are used to compute the posterior probability $\gamma_i(t)$ in the first iteration. The constant K in the constraint in Eq.9 is experimentally determined. It is approximately in the neighborhood of $L \cdot (\frac{1}{L})^p$.

	Set A	Set B	Set C
λ_{MFCC}	14.9	14.5	16.2
λ_{PLP}	15.1	15.0	16.2
λ_{LPCC}	18.3	17.2	16.8
ROVER ($\lambda_{\text{MFCC}} + \lambda_{\text{PLP}} + \lambda_{\text{LPCC}}$)	14.3	14.0	15.5
λ_{BS} (global, equal)	14.4	14.1	15.6
λ_{BS} (state,p=2,K=0.35)	14.3	13.9	15.6
λ_{BS} (state,p=3,K=0.15)	14.4	13.6	15.7

Table 2: Word error rate (WER) of discriminatively trained acoustic models (fMPE and MPE) on Sets A, B and C.

From the WERs in the table, it can be observed that λ_{BS} with simply equal weights yields comparable performance with ROVER. It improves the overall performance by relative 4% compared to the best individual models. Although state dependent weights give as much as 0.5% absolute improvement in Set B, overall they only slightly further reduce the WER from equal weights. Similar relative improvements are also observed on ML models but with discriminative training it yields the best overall performance.

4. Discussion and Summary

This paper studies a feature bootstrapping scheme with multiple acoustic features. A primary feature is first chosen to build the decision tree and initial ML model. Afterwards, ML models for all the features are estimated by single-pass re-training which allows different features from the same time stamp softly aligned to the same state. It is equivalent to resampling the state with multiple types of features. Feature space and model space discriminative training are also performed before the final acoustic models are combined by averaging the likelihood scores from distributions of different features in each state. The weights could be simply equal across the features or state dependent. If the features are complementary, the bootstrapped models are able to combine to improve the overall performance.

Compared to conventional ASR model combination techniques, such as randomized decision trees [2, 4] or ROVER [5], the proposed feature bootstrapping has only one decision tree and decoding graph which avoids multiple decoding. Therefore, it is computationally less demanding and requires smaller footprint.

The proposed feature bootstrapping also has subtle difference from multi-stream HMMs. Given the acoustic model and feature streams, multi-stream HMMs have a similar manner of computing the state likelihood of the streams (or sub-features) as Eq.4. But in published literature, multiple streams in most cases consist of static speech feature and its dynamics, such as [7, 8, 15], where the stream weights can either be fixed or estimated under ML or some other discriminative criterion. The multi-stream HMMs in the published works are often trained on augmented feature vectors composed of concatenation of the streams instead of exploiting the state posteriors as the feature bootstrapping. Besides, they usually do not require the streams to be complementary. Finding independent and complementary features is still an interesting and important topic for model combination. Audio-visual multi-stream models (e.g. [14]) do have complementary steams (audio and visual) but they are not strictly aligned to the same state. It would be interesting to include visual feature as complementary to the existing acoustic features in the future work.

5. Acknowledgements

This material is based upon work partially supported by the DARPA Transtac project.

6. References

[1] T. Dietterich, "Ensemble methods in machine learning", *Lecture Notes in Computer Science*, Vol. 1857, pp. 1-15, 2000.

[2] O. Siohan, B. Ramabhadran and B. Kingsbury, "Contructing ensembles of ASR systems using randomized decision trees", *ICASSP*, pp. 197-200, 2005.

[3] C. Breslin and M. Gales, "Complementary system generation using directed decision trees," *ICASSP*, pp. 337-340, 2007,

[4] J. Xue and Y. Zhao, "Random forests of phonetic decision trees for acoustic modeling in conversational speech recognition", *IEEE Trans. on Audio, Speech and Language Processing*, Vol.16, No.3, pp. 519-528, 2008.

[5] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)", *ASRU*, pp. 347-354, 1997.

[6] L. Breiman, "Random forests", *Machine Learning*, vol. 45, pp.5-32, 2001.

[7] C. Yang, F. Soong and T. Lee, "Static and dynamic spectral features: their noise robustness and optimal weights for ASR", *IEEE Trans. on Audio, Speech and Language Processing*, Vol.15, No.3, pp. 1087-1097, 2007.

[8] J. Hernando, "Maximum likelihood weighting of dynamic speech features for CDHMM speech recognition", *ICASSP*, pp. 1267-1270, 1997.

[9] O. Missaoui and H. Frigui, "Optimal feature weighting for the continuous HMM", *ICPR*, pp. 1-4, 2008.

[10] A. Dempster, N. Laird and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society*, Vol. 39, No. 1, pp. 1-38, 1977.

[11] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau and G. Zweig, "fMPE: discriminatively trained features for speech recognition", *ICASSP*, pp. 961-964, 2005.

[12] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training", *ICASSP*, pp. 105-108, 2002.

[13] D. Povey, "Improvements to fMPE for discriminative training of features", *Interspeech*, pp. 2977-2980, 2005.

[14] J. Huang and D. Povey, "Discriminatively trained features using fMPE for multi-stream audio-visual speech recognition", *Interspeech*, pp. 777-780, 2005.

[15] S. Young et. al., *The HTK Book 3.4*, Cambridge University.