Acoustic Modeling with Bootstrap and Restructuring Based on Full Covariance

Xiaodong Cui[1]*, Xin Chen*[2]*, Jian Xue*[1]*, Peder A. Olsen*[1]*, John R. Hershey*[3] *and Bowen Zhou*[1]

[1]IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598, USA
[2]Department of Computer Science, University of Missouri, Columbia, MO, 65211 USA
[3]Mitsubishi Electric Research Laboratories, Cambridge, MA, 02139, USA
{cuix,jxue, pederao, bowen}@us.ibm.com[1], XinChen@mizzou.edu[2], hershey@merl.com[3]

Abstract

Bootstrap and restructuring (BSRS) has been shown in our previous work to be superior over the conventional acoustic modeling approach when dealing with low-resourced languages. This paper presents a full covariance based BSRS scheme, which is an extension of our previous work on diagonal covariance based BSRS acoustic modeling. Since full covariance provides richer structural information of acoustic model compared to its diagonal counterpart, it is advantageous for both model clustering and refinement. Therefore, in this work, full covariance is employed in BSRS to keep the structural information until the last step before being converted to diagonal covariance for practical applications. We show that using full covariance further improves the performance over diagonal covariance in the BSRS acoustic modeling framework under the same model size without increasing computational cost in decoding.

Index Terms: acoustic modeling, bootstrap, model restructuring, full covariance

1. Introduction

Acoustic modeling with bootstrap and restructuring (BSRS) was investigated in [1]. In BSRS, the training data was randomly sampled without replacement into multiple subsets and an acoustic model was trained from each bootstrapped subset. All the bootstrapped models shared the same decision tree. Those acoustic models were then aggregated to yield a more reliable acoustic model given the training data. The aggregated model obtained this way was large in size due to parametric redundancy. Gaussian clustering and model refinement were then applied to scale the model down to a desired size while keeping the performance close to the original aggregated model. BSRS has been shown to be effective when dealing with low-resourced languages. In [1], all covariance matrices were assumed to be diagonal since only diagonal covariances are affordable in virtually all practical applications. However, full covariance clearly contains richer structural information than diagonal covariance. In this paper, BSRS with full covariance is investigated where full covariance is employed for both Gaussian clustering and model refinement before being converted to diagonal covariance in the final acoustic model. With richer structural information it is reasonable to expect better performance out of the full covariance in terms of model restructuring.

The remainder of the paper is organized as follows. Section 2 gives the mathematical formulation of the bootstrap and model aggregation. Section 3 describes in detail the full covariance based model restructuring process including Gaussian clustering and model refinement in the full covariance space, conversion from full to diagonal covariance, and model refinement again in the diagonal covariance space. Experimental results are presented in Section 4 followed by a summary in Section 5.

2. Model Aggregation with Bootstrap

Suppose S is the training data set with R utterances, $|S| = R$, and $\mathcal{P}$ is the true underlying distribution. In bootstrap [2], $\mathcal{P}$ is approximated by the empirical sample distribution $\mathcal{F}$ which concentrates mass $\frac{1}{R}$ at each observed sample. Out of the original data set S, generate N subsets of data, $\{S_1, S_2, \cdots, S_N\}$, by resampling from S without replacement. Each subset covers a fraction, r, of the original data, namely, $|S_i| = r \cdot |S|$, $0 < r \leq 1$, $i = 1, \cdots, N$. An HMM, denoted $\lambda_{\mathsf{bs},i}$, is estimated from each individual subset S_i. For reliable estimation, an aggregated HMM is computed as

$$\lambda_b = \frac{1}{N}\sum_{i=1}^{N} \lambda_{\mathsf{bs},i} \approx \mathcal{E}_{\mathcal{F}}\left[\lambda_{\mathsf{bs}}\right] \tag{1}$$

which can be considered an approximation to the expectation of the estimated parameters with respect to the empirical sample distribution $\mathcal{F}$.

Analogous to [1], all HMMs $\lambda_{\mathsf{bs},i}$ share the same LDA (Linear Discriminant Analysis) matrix, global STC (Semi-Tied Covariance) and decision tree which are built on the whole ensemble of resampled subsets. In this case the model aggregation amounts to averaging on observation distributions $f_{\mathsf{bs},i,s}(x)$ in each state s with GMM distribution of K_{is} Gaussian components across all the bootstrapped HMMs $\lambda_{\mathsf{bs},i}$.

$$\begin{aligned} f_s(x) &= \frac{1}{N}\sum_{i=1}^{N} f_{\mathsf{bs},i,s}(x) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k_{is}=1}^{K_{is}} c_{k_{is}}\mathcal{N}(x;\mu_{k_{is}},\Sigma_{k_{is}}) \\ &\triangleq \sum_{i=1}^{N}\sum_{k_{is}=1}^{K_{is}} w_{k_{is}}\mathcal{N}(x;\mu_{k_{is}},\Sigma_{k_{is}}) \end{aligned} \tag{2}$$

with $w_{k_{is}} = c_{k_{is}}/N$. Eq.2 shows that the model aggregation results in an HMM with a larger GMM in each state. For state s, there are $M_s = \sum_{i=1}^{N} K_{is}$ Gaussian components and the covariance matrices $\Sigma_{k_{is}}$ under discussion in Eq.2 are full matrices. Note that built on the ensemble of resampled subsets with significantly more samples, the decision tree associated with the aggregated HMM λ_b can grow deeper with resampling occurring at node split. Since the sampling is carried out without replacement, which is different from the conventional bootstrap approaches that sample with replacement, the approach investigated here can be categorized as "subbagging (subset bagging)"[3][4].

The aggregated model will be shown to give significantly better performance over single systems built from the original training data under the conventional training recipe. However, as it is obvious from Eq.2, this improved performance comes at a cost of substantially larger model size. Therefore, model restructuring is applied to scale down the model size for practical usage while maintaining decent performance.

3. Model Restructuring

Given the aggregated model λ_b with full covariance, Fig.1 illustrates the process of model restructuring. Gaussian clustering is first performed to reduce Gaussian components to a reasonable number which is followed by a model refinement to minimize certain "distance" between the down-scaled GMM and the original aggregated large GMM in each state. These two steps are carried out in the full covariance space. After the model refinement, the down-scaled GMM in full covariance is converted to one in diagonal covariance. Lastly, final model λ_r is obtained by another model refinement conducted in the diagonal covariance space. In what follows, each step in Fig.1 will be elaborated.

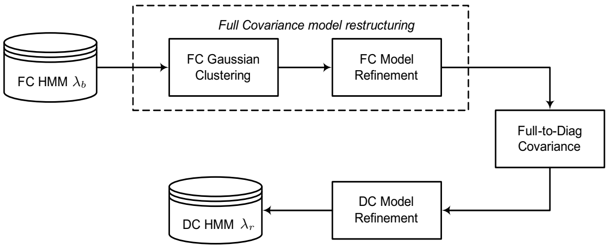

Figure 1: Model restructuring of the aggregated model with full covariance (FC) and conversion to diagonal covariance (DC).

3.1. Full Covariance Gaussian Clustering

In order to perform Gaussian clustering, a "distance" between two Gaussians needs to be defined. Similar to [1] in the diagonal space , a variety of "distances" including KL divergence, entropy and Bayes error are investigated in the full covariance space in this paper and their performance is compared in the experimental section.

3.1.1. KL divergence

Assume $f_1(x)$ and $f_2(x)$ are two Gaussian distributions, their KL divergence is computed as

$$D_{\text{kl}}(f_1||f_2) = \frac{1}{2}[\log\frac{|\Sigma_2|}{|\Sigma_1|} + \text{Tr}(\Sigma_2^{-1}\Sigma_1 - I_n) + (\mu_1 - \mu_2)^{\mathsf{T}}\Sigma_2^{-1}(\mu_1 - \mu_2)] \quad (3)$$

where n is the dimension of features.

3.1.2. Entropy

The change of entropy after the merge of two Gaussians with counts w_1 and w_2 is computed as

$$D_{\text{ent}}(f_1||f_2) = (w_1 + w_2)\log|\Sigma| - w_1 \log|\Sigma_1| - w_2 \log|\Sigma_2|$$

where

$$\Sigma = \frac{w_1}{w_1 + w_2}\Sigma_1 + \frac{w_2}{w_1 + w_2}\Sigma_2 + \frac{w_1 w_2}{(w_1 + w_2)^2}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^{\mathsf{T}}.$$

3.1.3. Bayes Error

The Bayes error [5], which measures the overlap between two distributions, is defined as

$$D_{\text{bayes}}(f_1||f_2) = \int \min(f_1(x), f_2(x))dx.$$

A Chernoff distance based variational approach is used to compute the above Bayes error when $f_1(x)$ and $f_2(x)$ are multivariate full covariance Gaussians. Define the Chernoff function

$$C(s) = C_s(f_1||f_2) = \int f_1(x)^s f_2(x)^{1-s}dx, \quad 0 \le s \le 1$$

from which the Chernoff distance is defined as

$$D_{\text{chern}}(f_1||f_2) = \min_{0 \le s \le 1} \int f_1(x)^s f_2(x)^{1-s}dx.$$

It can be shown that

$$D_{\text{bayes}}(f_1||f_2) \le D_{\text{chern}}(f_1||f_2).$$

Therefore the Chernoff distance $D_{\text{chern}}(f_1||f_2)$ is an upper bound of the Bayes error $D_{\text{bayes}}(f_1||f_2)$ and the latter can be approximated by the former.

Since the Chernoff function can be computed analytically [5] for the case of two Gaussians, some derivative-free approaches can be applied to search for the minimum of the convex Chernoff function. In this paper, a Newtown approach is used to search the minimum. Compared to derivative-free methods, it converges faster and is more efficient for clustering.

In particular, when Gaussian distribution (with mean μ and covariance Σ) is taken into consideration, it can be written in a general exponential family form

$$f(x) = \frac{1}{Z(\theta)}e^{\theta^{\mathsf{T}}\Phi(x)}, \quad Z(\theta) = \int e^{\theta^{\mathsf{T}}\Phi(x)}dx$$

with θ being the parameter, $\Phi(x)$ the sufficient statistics and $Z(\theta)$ the normalization term.

Let $P \triangleq \Sigma^{-1}$ and $\psi \triangleq P\mu$, then one has

$$\theta = \left[\text{vec}(P)^{\mathsf{T}}, \psi^{\mathsf{T}}\right]^{\mathsf{T}} \quad (4)$$

$$\Phi(x) = \left[-\frac{1}{2}\text{vec}(xx^{\mathsf{T}})^{\mathsf{T}}, x^{\mathsf{T}}\right]^{\mathsf{T}} \quad (5)$$

$$\log Z(\theta) = \frac{1}{2}\left[n\log(2\pi) - \log|P| + \psi^{\mathsf{T}}P^{-1}\psi\right] \quad (6)$$

where vectorization operator vec(A) creates a column vector from matrix A by stacking its column vectors.

Define $c(s) = \log C(s)$, one has

$$c(s) = \log Z(s\theta_1 + (1-s)\theta_2) - s\log Z(\theta_1) - (1-s)\log Z(\theta_2)$$

$c(s)$ is convex as a function of s and the Newton algorithm in Eq.7 can be applied to search for its minimum starting from any point, e.g. $s = \frac{1}{2}$, which corresponds to the Bhattacharyya distance [5].

$$s_{k+1} = s_k - \frac{c'(s)}{c''(s)} \quad (7)$$

It can be shown after derivation that

$$c'(s) = \log\frac{Z(\theta_2)}{Z(\theta_1)} + \sum_{i=1}^{n}\left[\frac{u_i v_i + s u_i^2 - \frac{1}{2}\xi_i}{1 + s\xi_i} - \frac{\frac{1}{2}\xi_i(v_i + s u_i)^2}{(1 + s\xi_i)^2}\right]$$

$$c''(s) = \sum_{i=1}^{n}\left[\frac{u_i^2}{1 + s\xi_i} - \frac{2\xi_i u_i v_i + 2s\xi_i u_i^2 - \frac{1}{2}\xi_i^2}{(1 + s\xi_i)^2} + \frac{\xi_i^2(v_i + s u_i)^2}{(1 + s\xi_i)^3}\right]$$

where

$$u = QP_2^{-\frac{1}{2}}\Delta_\psi \quad (8)$$

$$v = QP_2^{-\frac{1}{2}}\psi_2 \quad (9)$$

$$\Delta_p = P_1 - P_2 = \Sigma_1^{-1} - \Sigma_2^{-1} \quad (10)$$

$$\Delta_\psi = \psi_1 - \psi_2 = \Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2 \quad (11)$$

where Q and Ξ are the eigenvectors and eigenvalues of $P_2^{-\frac{1}{2}}\Delta_p P_2^{-\frac{1}{2}}$ and Ξ has ξ_i on its diagonal.

Based on the above-mentioned "distances", a variety of clustering strategies has been investigated which includes N-best distance refinement, K-step look-ahead and breadth-first searched best path in the greedy bottom-up scheme as well as two-pass structural non-local optimization. Their implementations are elaborated on in [6] with performance and speed extensively discussed.

3.2. Full Covariance Model Refinement

Model refinement on the down-scaled model after Gaussian clustering is accomplished by Monte Carlo based KL Minimization on GMM (MCKLGMM) which minimizes the KL divergence between two GMMs $f_1(x)$ and $f_2(x)$. Assume $f_1(x)$ is the reference distribution and $f_2(x)$ is the distribution to be optimized

$$f_2(x) = \underset{f_2(x)}{\text{argmin}}\, D_{\text{kl}}(f_1||f_2) = \underset{f_2(x)}{\text{argmin}} \int f_1(x)\log\frac{f_1(x)}{f_2(x)}dx$$

which amounts to the following maximization problem

$$f_2(x) = \underset{f_2(x)}{\text{argmax}} \int f_1(x)\log f_2(x)dx = \underset{f_2(x)}{\text{argmax}}\, \mathbf{E}_{f_1}[\log f_2(x)]$$

$$\approx \underset{f_2(x)}{\text{argmax}}\, \frac{1}{N}\sum_{i=1}^{N}\log f_2(x_i) \quad (12)$$

Eq.12 reveals that MCKLGMM can be carried out by first sampling from the reference distribution $f_1(x)$ and then fitting the samples with the GMM model to be optimized under maximum likelihood. The EM algorithm can be readily applied seeding from any reasonable down-scaled GMMs.

3.3. Full to Diagonal Covariance Conversion

After restructuring, the down-scaled and refined model is converted from full covariance to diagonal covariance. This comes with a loss of certain structural information. The conversion aims at keeping as much structural information as possible by minimizing against some "distance" between the full covariance GMM and diagonal GMM.

A straightforward conversion is to keep only the diagonal components in the Gaussians of the full covariance GMM while setting off-diagonal components to zeros. It can be shown that this way of conversion is equivalent to a Gaussian-component-wise minimization of the KL divergence between full and diagonal covariance Gaussians in the GMM models. In fact, when $f_1(x)$ has full covariance and $f_2(x)$ has diagonal covariance and they are both single Gaussians, the KL divergence in Eq.3 can be written into

$$2D_{\text{kl}}(f_1||f_2) = \sum_{i=1}^{n}\log d_i + \left[\frac{a_{11}}{d_1} + \cdots + \frac{a_{nn}}{d_n}\right]$$

$$- n + \sum_{i=1}^{n}\frac{(\mu_{1i}-\mu_{2i})^2}{d_i} - \log|\Sigma_1| \quad (13)$$

where a_{ij} are the components of the full covariance of $f_1(x)$, d_i the diagonal components of the diagonal covariance $f_2(x)$ and n the dimension of the multivariate Gaussians. Setting the differential to zero, one has $d_i = a_{ii}$.

Instead of considering only Gaussian components, the KL divergence minimization can also be carried out between two GMMs in which case $f_1(x)$ is a GMM with full covariance and $f_2(x)$ a GMM with diagonal covariance. Monte Carlo method is used again to perform the optimization. Following the same mathematical treatment in Eq.12, one generates samples according to the full covariance GMM $f_1(x)$ from which obtains the maximum likelihood estimate of the diagonal covariance GMM $f_2(x)$ via the EM algorithm. Gaussian-component-wise diagonalization can be used as a starting estimate of $f_2(x)$ in the EM algorithm.

3.4. Model Refinement with Diagonal Covariance

When the acoustic model is converted from full covariance to diagonal covariance, another model refinement is conducted in the diagonal covariance space before outputting the final model. In this step, Monte Carlo HMM based KL minimization (MCKLHMM) is employed.

Following the definitions in [7], let $x_{1:n} \triangleq (x_1, \cdots, x_n)$ be a sequence of observations and $f_1(x_{1:n})$ and $f_2(x_{1:n})$ are the reference HMM and HMM to be refined respectively.

$$f_2(x) = \underset{f_2(x_{1:n})}{\text{argmin}}\, D_{\text{kl}}(f_1||f_2) = \underset{f_2(x_{1:n})}{\text{argmax}}\, \mathbf{E}_{f_1}[\log f_2(x_{1:n})]$$

$$\approx \underset{f_2(x_{1:n})}{\text{argmax}}\, \frac{1}{N}\sum_{i=1}^{N}\log f_{2i}(x_{1:n}) \quad (14)$$

where N utterances are sampled according to HMM $f_1(x_{1:n})$. Same as [1], the ensemble of all bootstrapped utterances is treated as the N sequence samples according to the "ground truth" generative HMM $f_1(x_{1:n})$. The KL minimization on HMM in Eq.14 is equivalent to maximum likelihood estimation using the ensemble of all bootstrapped utterances starting from the diagonal covariance HMM obtained from Section 3.3.

4. Experimental Results

Pashto, one of the two major languages spoken in Afghanistan, is used for experiments. The data was collected and transcribed by DARPA under the Transtac project. There are 135 hours of training data from 116 speakers and 10 hours of test data from 22 speakers. The feature space is constructed by splicing 9 frames of 24 dimensional PLP features and projecting down to a 40 dimensional space via LDA followed by a global STC. Context-dependent quinphone states are tied by a decision tree. Discriminative training is applied to both the feature space (fMMI [8]) and model space (Boosted MMI [8]). The original training data is bootstrapped into 15 subsets with each subset covering 70% of the original training set. A trigram language model with 1.2M n-grams is used for test, with a dictionary of 30K words. A static graph with a compilation of LM, dictionary and decision tree is used for fast decoding [9].

Table 1 shows the performance of the ML baseline after the conventional training recipe, the performance of aggregated HMM model with full covariance (bs_full(ma)) and the performance of down-scaled model obtained by Gaussian clustering without further MCKLGMM refinement (bs_full_cluster) under different Gaussian distances, namely, KL divergence (KL), entropy (ENT) and Bayes error (Bayes). The sizes of the models are also presented in terms of the number of states and Gaussians (# of states/# of Gaussians). From the table, there is no significant difference in performance between the three clustering criteria.

model	size	WER
baseline	3.5K/100K	39.6%
bs_full(ma)	6K/1.8M	35.2%
bs_full_cluster(KL)	6K/100K	35.8%
bs_full_cluster(ENT)	6K/100K	36.0%
bs_full_cluster(Bayes Error)	6K/100K	36.0%

Table 1: WERs of three Gaussian clustering criteria (KL, Entropy and Bayes Error) of in the full covariance space.

Table 2 shows the comparative performance of ML models between baseline without bootstrap, BSRS with diagonal covariance (DC) and BSRS with full covariance (FC). The entropy criterion is used for Gaussian clustering in both FC and DC cases. First of all, since the decision tree embedded in the aggregated HMM is trained on the ensemble of all the bootstrapped data, it grows deeper resulting in more context-dependent states. To make a fair comparison, two more ML baselines (ML2 and ML3) are run in the experiments other than the conventional one (ML1) with the typical threshold of counts for node split at 5000 in the IBM attila toolkit. ML2 lowers the threshold to 1500 to reach 6K states while ML3 uses the whole ensemble of bootstrapped data with the threshold remaining 5000. From the table, both ML2 and ML3 yield almost the same WER as ML1 (39.7% vs. 39.6%; 39.5% vs. 39.6%). It indicates that simply lowering the threshold to grow the tree may lead to unreliable estimate of the parameters which may not improve the performance. On the other hand, simply pooling data together without introducing appropriate structure may not improve the performance either.

From Table 2, BSRS with DC yields 38.0% WER after model aggregation (bs_diag(ma)) and 38.6% after model restructuring (bs_diag(rs)). The overall improvement is 1.0% absolute over the baseline ML1. BSRS with FC yields 35.2% after model aggregation (bs_full(ma)), 35.7% after full covariance model restructuring (bs_full(rs)) and 38.1% after conversion to diagonal covariance and model refinement in the diagonal covariance space (bs_full2diag(rf)). The overall improvement is 1.5% absolute from the baseline and 0.5% absolute from its diagonal covariance counterpart.

After the BSRS is performed in the ML stage, the restructured diagonal acoustic model is used for discriminative training which is conducted in the conventional way. The performance of the acoustic models after feature and model space discriminative training is shown in Table 3. The baseline is obtained by FMMI and BMMI training on top of the ML1 from Table 2. From Table 3, it reveals that the improvement after discriminative training is smaller than that of ML training. Given the 135 hours of training data, BSRS with DC (bs_diag) gives 0.3% absolute gain over the baseline while BSRS with FC (bs_full2diag) gives 0.9% absolute gain over the baseline.

Note that the models of bs_diag and bs_full2diag have the same number of Gaussian as the baseline model. So there is almost no increase of computational cost when decoding. Since the size of static graph is not very sensitive to the number of states, there is also no significant change in the size of the graph with 6K states compared to the original 3.5K states. The proposed approach has proven to be useful for low-resourced languages. Compared to other ensemble acoustic models, e.g. [10] where tree array are used for decoding, the single decoding graph as a result of the shared decision tree makes it attractive for memory and speed constrained platforms, such as mobile phone.

model		size	WER
baseline	ML1	3.5K/100K	39.6%
	ML2	6K/100K	39.7%
	ML3	6K/100K	39.5%
DC space	bs_diag(ma)	6K/1.8M	38.0%
	bs_diag(rs)	6K/100K	38.6%
FC space	bs_full(ma)	6K/1.8M	35.2%
	bs_full(rs)	6K/100K	35.7%
	bs_full2diag(rf)	6K/100K	38.1%

Table 2: Sizes and WERs of ML models for baseline, BSRS with diagonal (DC) and full covariances (FC).

model	size	WER
baseline	3.5K/100K	34.1%
bs_diag	6K/100K	33.8%
bs_full2diag	6K/100K	33.2%

Table 3: Sizes and performance of FMMI+BMMI models for baseline, BSRS with diagonal and full covariances.

5. Summary

In this paper, we extend our previous BSRS acoustic modeling from diagonal covariance space to the full covariance space to make use of the rich structural information in the latter for better performance. Full covariance structure is employed in model restructuring which includes Gaussian clustering and model refinement. The full covariance structure is kept as long as possible in the modeling process until the last step before being converted back to the diagonal covariance space for acoustic models to be used in practical applications. Experiments show that this full covariance based BSRS approach can further improve the performance from the diagonal covariance based BSRS after both maximum likelihood and discriminative training.

6. References

[1] Cui, X., Xue, J., Dognin, P. L., Chaudhari, U. V., and Zhou, B., “Acoustic modeling wiht bootstrap and restructuring for low-resourced languages,” Interspeech, pp. 2794-2797, 2010.

[2] Efron, B., “Bootstrap methods: Another look at the jackknife, The Annals of Statistics, vol.7, no. 1, pp. 1-26, 1979.

[3] L. Breiman, “Baggging predictors,” Machine Learning, vol. 24, no. 2, pp. 123-140, 1996.

[4] A. Elisseeff, T. Evgeniou and M. Pontil, “Stability of randomized learning algorithms,” Journal of Machine Learning Research, vol. 6, pp. 55-79, 2005.

[5] R. O. Duda, P. E. Hart and D. G. Stork, “Pattern classification (2nd Edition),” John Wiley and Sons, 2001.

[6] Chen, X., Cui, X., Xue, J., Olsen, P. A., Hersey, J. R., and Zhou, B., “Full covariance bootstrapped acoustic model clustering,” ICASSP, pp. 4496-4499, 2011.

[7] Hershey J. R., and Olsen, P. A., “Variational Bhattacharyya divergence for hidden Markov models,” ICASSP, pp. 4557-4560, 2008.

[8] Povey, D., Kanevsky, D., Kingsbury, B., Ramabhadran, B., Saon, G., and Visweswariah, K., “Boosted MMI for model and feature-space discriminative training,” ICASSP, pp. 4057-4060, 2008.

[9] Saon, G., Povey, D., and Zweig, G., “Anatomy of an extremely fast LVCSR decoder,” Interspeech, pp. 549-552, 2005.

[10] X. Chen and Y. Zhao, “Integrating MLP features and discriminative training in data sampling based ensemble acoustic modeling,” Interspeech, pp. 1349-1352, 2010.