Hidden Markov Acoustic Modeling With Bootstrap and Restructuring for Low-Resourced Languages

Xiaodong Cui, Jian Xue, Xin Chen, Peder A. Olsen, Pierre L. Dognin, Upendra V. Chaudhari, John R. Hershey, and Bowen Zhou, *Member, IEEE*

***Abstract*—This paper proposes an acoustic modeling approach based on bootstrap and restructuring to dealing with data sparsity for low-resourced languages. The goal of the approach is to improve the statistical reliability of acoustic modeling for automatic speech recognition (ASR) in the context of speed, memory and response latency requirements for real-world applications. In this approach, randomized hidden Markov models (HMMs) estimated from the bootstrapped training data are aggregated for reliable sequence prediction. The aggregation leads to an HMM with superior prediction capability at cost of a substantially larger size. For practical usage the aggregated HMM is restructured by Gaussian clustering followed by model refinement. The restructuring aims at reducing the aggregated HMM to a desirable model size while maintaining its performance close to the original aggregated HMM. To that end, various Gaussian clustering criteria and model refinement algorithms have been investigated in the full covariance model space before the conversion to the diagonal covariance model space in the last stage of the restructuring. Large vocabulary continuous speech recognition (LVCSR) experiments on Pashto and Dari have shown that acoustic models obtained by the proposed approach can yield superior performance over the conventional training procedure with almost the same run-time memory consumption and decoding speed.**

***Index Terms*—Bagging, bootstrap and restructuring, hidden Markov model (HMM), low-resourced language, large vocabulary continuous speech recognition (LVCSR).**

I. Introduction

AUTOMATIC speech recognition (ASR) as an enabling technique has been widely used in human language technologies nowadays such as voice search and navigation, natural language understanding, dialog systems, and speech-to-speech translation. With such technologies prevailing worldwide, it is crucial and necessary for them to deploy rapidly in new languages. One of the barriers in porting to new languages is data sparsity, especially for those low-resourced languages with limited availability of transcribed speech data. The collection and annotation of speech signals from such languages are both expensive and time-consuming. As a result, acoustic modeling suffers from the data sparsity under this condition, which often leads to unsatisfactory performance of ASR. Therefore, it becomes doubly important to make good use of the data.

Various methods have been proposed for acoustic modeling on low-resourced languages to deal with the data sparsity, e.g., [1]–[3]. Among them, besides the unsupervised approaches, one interesting way to cope with the limited transcribed data is using multilingual information to help the acoustic modeling of the sparse new language with data and knowledge from other related languages [3]–[7]. For instance, in [3], decent performance has been shown by investigating cross-lingual independent and dependent acoustic modeling for bootstrapping the ASR systems in Vietnamese. In [6], it was found in the Southern Bantu language family that by pooling speech data from closely related languages improvements can be observed in recognition accuracy when the training data is scarce. Multilingual acoustic model estimation was extensively studied in [4] where speech data from various source languages is combined with the sparse data from the target new language for a fast ASR deployment in the new language.

In this paper, an approach based on bootstrap and restructuring (BSRS) is proposed for hidden Markov acoustic modeling with sparse training data. Bootstrap [8]–[10], as one of the resampling statistical methods, finds its success in a broad spectrum of applications [11], [12]. It is widely used for estimating the properties of an estimator by resampling from the empirical distribution of the observed data, especially when the observed data is insufficient. Bootstrap related methods are also actively investigated in machine learning. Bagging (bootstrap aggregating) [13], [14], which belongs to a family of ensemble-based randomized learning algorithms, can be considered an extension of bootstrap to improve the accuracy and stability of classifiers by reducing the variance of estimates through majority voting or averaging across a set of weak learners. Therefore, it provides an effective way to create robust decisions by aggregating randomized models from the original training data. This is particularly intriguing when data sparsity is an issue for reliable model estimation.

The approach proposed in this paper aims at robust acoustic modeling based on bootstrap and restructuring to deal with the data sparsity posed by low-resourced languages. While we focus on obtaining reliable acoustic models with better recognition accuracy, we also consider run-time speed and memory constraints. In this approach, the original training data is randomly resampled into multiple subsets and hidden Markov models (HMMs) are estimated from each individual

Manuscript received September 28, 2011; revised March 05, 2012; accepted April 24, 2012. Date of publication May 17, 2012; date of current version August 09, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Brian Mak.

X. Cui, J. Xue, P. A. Olsen, P. L. Dognin, U. V. Chaudhari, and B. Zhou are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: cuix@us.ibm.com, jxue@us.ibm.com; pederao@us.ibm.com; pdognin@us.ibm.com; uvc@us.ibm.com; zhou@us.ibm.com).

X. Chen is with the Pearson, Knowledge Technologies Group, Menlo Park, CA, 94025 USA (e-mail: xinchen@mail.mizzou.edu).

J. R. Hershey is with the Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: hershey@merl.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2012.2199982

resampled subset. The discriminant functions from the randomized HMMs are aggregated at the state level, which gives rise to one aggregated HMM with a substantially larger number of parameters. Although the aggregated HMM classifier obtained this way gives significantly better recognition accuracy, it is computationally prohibitive for practical usage. Therefore, the aggregated HMM is restructured to a smaller model size while the performance loss due to the downsizing is minimized. Preliminary results have been reported in [15] by model restructuring based on diagonal covariance models. In this paper, the model restructuring is performed in the full covariance space to keep the rich structural information of the data until the final step when the models are converted to the diagonal covariance space.

The work reported in this paper was conducted under the DARPA TRANSTAC (Spoken Language Communication and TRanslation System for TACtical Use) program for the ASR component in automatic speech-to-speech (S2S) translation. The language pairs were English–Pashto and English–Dari. Both Pashto and Dari were low resource languages on the order of 150 hours of training data. The data was collected, transcribed and distributed by DARPA TRANSTAC sequentially in a two-year span. The target deployment platforms of the S2S translation systems were initially laptop computers and later on smartphones. Due to low latency requirements and the platform capability, the computational power of the CPU and availability of run-time memory were also important factors to take into account for acoustic modeling. Such requirements are typical for real-world ASR applications, particularly for stand-alone systems. In what follows, we will show that the proposed BSRS approach, while using almost the same amount of computing resources, can yield superior performance to the conventional training procedure, under both the maximum-likelihood (ML) training and discriminative training, given the limited amount of training data.

The remainder of the paper is organized as follows. Section II describes the subbagging of multiple randomized HMMs. Section III elaborates on the restructuring of the aggregated HMM, which includes Gaussian clustering and model refinement. A variety of clustering criteria and model refinement algorithms are studied. Experimental results on Pashto and Dari are presented in Section IV followed by a discussion in Section V. Finally we conclude with a summary in Section VI.

II. Subbagging of Randomized HMMs

Suppose $\mathcal{L}$ is the training set with R labeled utterances, $\mathcal{L} = \{(\mathcal{O}_r, W_r),\ r = 1, \cdots, R\}$, where the acoustic feature sequence for the rth utterance is $\mathcal{O}_r = \{O_1^r, \cdots, O_{\mathsf{T}_r}^r\}$ with T_r frames and W_r is the corresponding label sequence. We assume $(\mathcal{O}_r, W_r)$ are independently drawn from an underlying distribution $\mathcal{P}(\mathcal{O}, W)$. In ASR, one learns from $\mathcal{L}$ an HMM λ that predicts the word sequence W for an unknown speech feature sequence $\mathcal{O}$

$$\varphi_\lambda(O, \mathcal{L}) : O \longmapsto W. \tag{1}$$

The classification is accomplished by computing a discriminant function, $D_\lambda(\mathcal{O}, W)$, for all possible word sequences W and choosing the one with the maximum score as the predicted word sequence

$$W^* = \arg\max_W D_\lambda(\mathcal{O}, W). \tag{2}$$

For HMM-based classifiers, the discriminant function has the following form:

$$D_\lambda(\mathcal{O}, W) = \log[P_\lambda(\mathcal{O}|W)P(W)] \tag{3}$$

where $P(W)$ is the language model probability for W and $P_\lambda(\mathcal{O}|W)$ is the acoustic model likelihood for $\mathcal{O}$ given W. The latter is computed by the HMM λ which can be estimated under various criteria (e.g., ML [16], maximum mutual information (MMI) [17], boosted maximum mutual information (BMMI) [18], or minimum phone error (MPE) [19]). In ASR, this sequence classification process is well known as the Viterbi decoding [20] in which the optimal state sequence is found and mapped into the word sequence. Here we assume $P(W)$ is a constant that can be ignored and focus on the acoustic model for simplicity of discussion. Assume there are T frames in $\mathcal{O}$ so that $\mathcal{O} = \{O_1, \cdots, O_t, \cdots, O_{\mathsf{T}}\}$ and its corresponding Viterbi state sequence is $\mathcal{S} = \{S_1, \cdots, S_t, \cdots, S_{\mathsf{T}}\}$. The discriminant function in (3) can be written as

$$\begin{aligned} D_\lambda(\mathcal{O}, S) &= \log\left[\pi_{s_1} f_{s_1}(O_1) \prod_{t=2}^{T} a_{s_{t-1}s_t} f_{s_t}(O_t)\right] \\ &= \sum_{t=1}^{T} \log f_{s_t}(O_t) + \sum_{t=2}^{T} \log a_{s_{t-1}s_t} + \log \pi_{s_1} \end{aligned} \tag{4}$$

where π_{s_t}, $a_{s_{t-1}s_t}$, and $f_{s_t}(O_t)$ are initial probabilities, transition probabilities and state observation probability densities of the HMM λ. From the Viterbi decoding perspective, one can write the HMM-based sequence classifier as

$$\varphi_\lambda(\mathcal{O}, \mathcal{L}) = \arg\max_S D_\lambda(\mathcal{O}, \mathcal{S}). \tag{5}$$

If $\mathcal{L}$ is sparse, the classifier $\varphi_\lambda(\mathcal{O}, \mathcal{L})$ might be statistically unreliable. To stabilize the classifier given the sparse $\mathcal{L}$, the strategy of aggregation by bootstrapping is applied. In bootstrap, the true underlying distribution $\mathcal{P}(\mathcal{O}, W)$ is approximated by the empirical distribution $\mathcal{F}(\mathcal{O}, W)$ with mass $1/R$ concentrated at each sample $(\mathcal{O}, W)$ in the original training set $\mathcal{L}$. From $\mathcal{L}$, N subsets of data, $\{\mathcal{L}_1^B, \mathcal{L}_2^B, \cdots, \mathcal{L}_N^B\}$, are generated by resampling from $\mathcal{L}$ without replacement. Each subset covers a fraction, δ, of the original data, namely, $|\mathcal{L}_i^B| = \delta \cdot |\mathcal{L}|, 0 < \delta \leq 1, i = 1, \cdots, N$. A Gaussian mixture HMM, λ_i^B, is estimated from each bootstrapped subset $\mathcal{L}_i^B$. We assume that the N HMMs have the same initial and transition probabilities and further assume that they also share the same linear discriminant analysis (LDA) matrix, semi-tied covariance (STC) matrix and decision tree.

With the randomized HMMs estimated from the bootstrapped data, the classification can be done by finding the optimal state sequence using the aggregated discriminant function

$$\varphi_A(\mathcal{O}) = \arg\max_S \mathbf{E}_{\mathcal{L}^B}\{D_{\lambda^B}(\mathcal{O}, S)\} \tag{6}$$

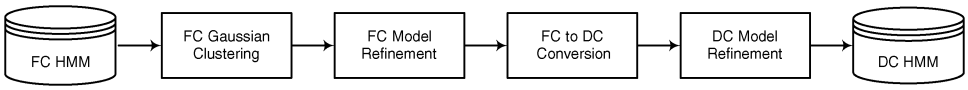


Fig. 1. Pipeline of restructuring of the aggregated HMM from the full covariance (FC) space to the diagonal covariance (DC) space.

where $D_{\lambda^B}(\mathcal{O}, S)$ are the discriminant functions for the randomized HMMs λ^B estimated from $\mathcal{L}^B$ and $\mathbf{E}_{\mathcal{L}^B}$ denotes that the average is with respect to the bootstrapped data $\mathcal{L}^B$. Overall, the aggregated sequence classifier is denoted by $\varphi_A(\mathcal{O})$.

Under the assumption of equal initial and state transition probabilities and the same LDA, STC and decision tree, the aggregated discriminant function is computed as

$$\mathbf{E}_{\mathcal{L}^B}\left\{D_{\lambda^B}(\mathcal{O}, S)\right\} = \sum_{t=1}^{T} \mathbf{E}_{\mathcal{L}^B}\left\{\log f_{s_t}^B(O_t)\right\} + \sum_{t=2}^{T} \log a_{s_{t-1}s_t} + \log \pi_{s_1} \quad (7)$$

which can be approximated by its upper bound under Jensen's inequality

$$\mathbf{E}_{\mathcal{L}^B}\left\{D_{\lambda^B}(\mathcal{O}, S)\right\} \approx \sum_{t=1}^{T} \log \mathbf{E}_{\mathcal{L}^B}\left\{f_{s_t}^B(O_t)\right\} + \sum_{t=2}^{T} \log a_{s_{t-1}s_t} + \log \pi_{s_1}. \quad (8)$$

Supposing the state observation distribution of HMM λ_i^B in state s is a Gaussian mixture model (GMM), one has

$$f_s^{B_i}(O_t) = \sum_{k \in \lambda_i^B(s)} c_k \mathcal{N}(O_t; \mu_k, \Sigma_k) \quad (9)$$

where μ_k, Σ_k, and c_k are the means, covariances and weights, respectively. Based on that, the aggregated likelihood score from state s is computed as

$$\begin{aligned} f_s^A(O_t) &= \mathbf{E}_{\mathcal{L}^B}\left\{f_s^B(O_t)\right\} \\ &\approx \frac{1}{N}\sum_{i=1}^{N} f_s^{B_i}(O_t) \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k \in \lambda_i^B(s)} c_k \mathcal{N}(O_t; \mu_k, \Sigma_k) \\ &\triangleq \sum_{i=1}^{N}\sum_{k \in \lambda_i^B(s)} c'_k \mathcal{N}(O_t; \mu_k, \Sigma_k) \end{aligned} \quad (10)$$

with $c'_k = (c_k/N)$.

It can be observed from (10) that the approximation of the aggregation in (8) results in a new GMM in $f_s^A(O_t)$ with $M_s = \sum_{i=1}^{N}\sum_{k \in \lambda_i^B(s)} 1$ Gaussian components in state s. Therefore, it also gives rise to a single aggregated HMM λ^A which corresponds to the aggregated discriminant function. Sharing LDA, STC, and the decision tree will sacrifice some diversity of the randomized HMMs. However, the resulting aggregated HMM λ^A does not require multiple decodings, which makes it preferable for real-world deployment with constrained resources (CPU/memory) and a low response latency requirement.

Note that subsampling is used in this work where the bootstrapped sets are generated by sampling without replacement. This is different from the conventional bagging approaches and can be categorized as subbagging (subsample bagging) [21], [22]. Subbagging, analogous to bagging, can also be shown to reduce the variance of continuous regression functions or classifiers after aggregation to stabilize the estimators or classifiers with less expensive computations. Analysis of the stability on bagging or subbagging can be found in [13], [14], [21], [23]. Theoretically, subbagging can be shown to yield approximately the same accuracy as bagging [23]. Empirically, we have also observed that, in this particular work, sampling without replacement performs better than with replacement.

III. Model Restructuring

The aggregated HMM λ^A is a more reliable model after subbagging given the sparse training data $\mathcal{L}$. Nevertheless, it has a large model size consisting of a substantial number of Gaussians. The direct usage of this model is computationally prohibitive in real scenarios and therefore model complexity control should be applied. Model complexity control aiming at selecting an appropriate model structure with an optimal complexity has been actively studied for Large vocabulary continuous speech recognition (LVCSR) [24]–[28]. Various criteria on model selection and Gaussian elimination have been proposed or investigated to automatically determine the model complexity in ASR systems, e.g., minimum description length (MDL) [27], [28], Bayesian information criterion (BIC) [25], Gaussian importance measure (GIM) [26], and marginalized discriminative growth function [24]. In this section, the model complexity control is accomplished by a model restructuring process whose goal is to downsize the model to a reasonable size for practical usage while maintaining decent performance.

Fig. 1 illustrates the model restructuring process which is composed of Gaussian clustering and model refinement. Since the full covariance can give a better description of the structure of the data than the diagonal covariance, it is used in the HMM training as well as the Gaussian clustering and model refinement. On the other hand, almost only the diagonal covariance model is practical in real-world applications. Hence, the informative full covariance model is used in the restructuring process as long as possible until the final stage where it is converted to the diagonal covariance space followed by another refinement. Overall, the model restructuring downsizes a full covariance HMM with a large number of Gaussians to a diagonal covariance HMM with a small number of Gaussians. Meanwhile, the performance loss due to the downsizing is minimized according to some criterion.

A. Full Covariance Gaussian Clustering

Gaussian clustering aims to reduce the model size by merging Gaussian components in the GMM distribution in each state of the aggregated HMM. Merging of Gaussian distributions has been widely studied in acoustic modeling and other statistical modeling areas [29]–[31]. Here, similar to those previous works, the Gaussian clustering is carried out in a greedy bottom-up fashion in the full covariance space. In each iteration, the two Gaussians that are most similar under a certain measurement are merged into a new Gaussian with the mean and covariance [30], [31] shown in (11) and (12):

$$\mu = \frac{w_1}{w_1+w_2}\mu_1 + \frac{w_2}{w_1+w_2}\mu_2 \tag{11}$$

$$\Sigma = \frac{w_1}{w_1+w_2}\Sigma_1 + \frac{w_2}{w_1+w_2}\Sigma_2 + \frac{w_1 w_2}{(w_1+w_2)^2}(\mu_1-\mu_2)(\mu_1-\mu_2)^{\mathsf{T}}. \tag{12}$$

The merging procedure keeps going until the total number of Gaussians meets a predefined target number. To measure the similarity between Gaussians, a variety of measurements are investigated which include Kullback–Leibler (KL) divergence, entropy and Bayes error.

1) KL Divergence: The KL divergence between two distributions $f_1(x)$ and $f_2(x)$ is defined as [32]

$$D_{kl}(f_1\|f_2) = \int f_1(x)\log\frac{f_1(x)}{f_2(x)}dx. \tag{13}$$

When $f_1(x)$ and $f_2(x)$ are multivariate Gaussian distributions, $x \in \mathcal{R}^d$, (13) can be expressed as

$$D_{kl}(f_1\|f_2) = \frac{1}{2}\left\{\log\frac{|\Sigma_2|}{|\Sigma_1|} + \mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1 - I_d\right) + (\mu_1-\mu_2)^{\mathsf{T}}\Sigma_2^{-1}(\mu_1-\mu_2)\right\} \tag{14}$$

where μ_1, μ_2, Σ_1, and Σ_2 are means and covariances of the two Gaussians, respectively. The symmetric KL divergence shown in (15) is actually used for clustering:

$$D_{kls}(f_1, f_2) = D_{kl}(f_1\|f_2) + D_{kl}(f_2\|f_1). \tag{15}$$

2) Entropy: The entropy criterion measures the change of entropy after the merge of two Gaussians. Merging similar Gaussians gives a small change of entropy. Let w_1 and w_2 denote the counts associated with the two Gaussians, then the change of entropy is computed as

$$D_{ent}(f_1\|f_2) = (w_1+w_2)\log|\Sigma| - w_1\log|\Sigma_1| - w_2\log|\Sigma_2| \tag{16}$$

where Σ_1 and Σ_2 are the covariances of the two Gaussians and Σ is the covariance of the merged Gaussian computed according to (12).

3) Bayes Error: The Bayes error measures the overlap between two distributions which is defined as [33]

$$D_{bayes}(f_1\|f_2) = \int \min\left(f_1(x), f_2(x)\right)dx. \tag{17}$$

To merge two Gaussians that are most similar, the negative Bayes error $-D_{bayes}(f_1\|f_2)$ is actually used to measure their similarity.

There is no closed-form solution for the Bayes error for arbitrary distributions including high-dimensional multivariate Gaussian distributions. In this work, (17) is approximated by the Chernoff distance.

From the Chernoff function

$$C(s) = C_s(f_1\|f_2) = \int f_1(x)^s f_2(x)^{1-s}dx, \quad 0 \le s \le 1 \tag{18}$$

the Chernoff distance is defined as

$$D_{chern}(f_1\|f_2) = \min_{0\le s\le 1} C(s) = \min_{0\le s\le 1}\int f_1(x)^s f_2(x)^{1-s}dx. \tag{19}$$

It can be proven that the Chernoff distance is an upper bound of the Bayes error [33]:

$$D_{bayes}(f_1\|f_2) \le D_{chern}(f_1\|f_2). \tag{20}$$

and therefore the Chernoff distance is employed to approximate the Bayes error. In this work, we introduce a Newton approach to searching for the minimum of $C(s)$.

Express the Gaussian distribution as an exponential family:

$$f(x) = \frac{1}{Z(\theta)}e^{\theta^{\mathsf{T}}\Phi(x)}, \qquad Z(\theta) = \int e^{\theta^{\mathsf{T}}\Phi(x)}dx \tag{21}$$

with θ being the parameter, $\Phi(x)$ the features and $Z(\theta)$ the normalization term. Let $P \triangleq \Sigma^{-1}$ and $\psi \triangleq P\mu$, then one has

$$\theta = \left[\mathrm{vec}(P)^{\mathsf{T}}, \psi^{\mathsf{T}}\right]^{\mathsf{T}} \tag{22}$$

$$\Phi(x) = \left[-\frac{1}{2}\mathrm{vec}(xx^{\mathsf{T}})^{\mathsf{T}}, x^{\mathsf{T}}\right]^{\mathsf{T}} \tag{23}$$

$$\log Z(\theta) = -\frac{1}{2}\left[d\log(2\pi) - \log|P| + \psi^{\mathsf{T}}P^{-1}\psi\right] \tag{24}$$

where the vectorization operator $\mathrm{vec}(A)$ stacks the columns of the matrix A into a single column vector and d is the dimension of the feature space.

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

$$c(s) = \log Z\left(s\theta_1 + (1-s)\theta_2\right) - s\log Z(\theta_1) - (1-s)\log Z(\theta_2) \tag{25}$$

which can be proven to be a convex function of s. Starting from some initial value s_0, s is iteratively updated by

$$s_{k+1} = s_k - \frac{c'(s)}{c''(s)} \tag{26}$$

where $c'(s)$ and $c''(s)$ are the first- and second-order derivatives of $c(s)$. A reasonable choice of s_0 is $s_0 = 1/2$ which corresponds to the Bhattacharyya distance [33].

It can be shown that (see Appendix A for the details)

$$c'(s) = \log\frac{Z(\theta_2)}{Z(\theta_1)} + \sum_{i=1}^{d}\left[\frac{u_i v_i + s u_i^2 - \frac{1}{2}\xi_i}{1+s\xi_i} - \frac{\frac{1}{2}\xi_i(v_i+su_i)^2}{(1+s\xi_i)^2}\right] \quad (27)$$

$$c''(s) = \sum_{i=1}^{d}\left[\frac{u_i^2}{1+s\xi_i} - \frac{2\xi_i u_i v_i + 2s\xi_i u_i^2 - \frac{1}{2}\xi_i^2}{(1+s\xi_i)^2} + \frac{\xi_i^2(v_i+su_i)^2}{(1+s\xi_i)^3}\right] \quad (28)$$

where

$$\Delta_p = P_1 - P_2 = \Sigma_1^{-1} - \Sigma_2^{-1} \quad (29)$$

$$\Delta_\psi = \psi_1 - \psi_2 = \Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2 \quad (30)$$

$$P_2^{-\frac{1}{2}}\Delta_P P_2^{-\frac{1}{2}} = Q^\top \Lambda Q \quad (31)$$

$$u = Q P_2^{-\frac{1}{2}}\Delta_\psi \quad (32)$$

$$v = Q P_2^{-\frac{1}{2}}\psi_2. \quad (33)$$

Q and $\Lambda = \mathrm{diag}\{\xi_1, \cdots, \xi_i, \cdots, \xi_d\}$ are the matrices composed of eigenvectors and eigenvalues of $P_2^{-1/2}\Delta_p P_2^{-1/2}$. Note that with the above matrices precomputed the run-time computation is reduced from $\mathcal{O}(d^3)$ to $\mathcal{O}(d)$.

Aside from the greedy bottom-up Gaussian clustering strategy discussed here, various non-local and multipass clustering schemes were extensively investigated in [34] with comparisons of the three clustering criteria in terms of speed and accuracy.

B. Full Covariance Model Refinement

After the original aggregated HMM is downsized to a desirable size using Gaussian clustering, the downsized model is then subjected to further refinement in the full covariance space. The goal is to minimize the KL divergence between the downsized model and the original aggregated model. The refinement is performed state by state.

Assume $f_1(x)$ is the aggregated GMM distribution in a state as shown in (10), and $f_2(x)$ is the downsized GMM distribution to be optimized in the same state, one wants to find $f_2(x)$ in a GMM space with a smaller number of Gaussian components such that

$$f_2^*(x) = \arg\min_{f_2(x)} \int f_1(x)\log\frac{f_1(x)}{f_2(x)}dx = \arg\max_{f_2(x)} \int f_1(x)\log f_2(x)dx. \quad (34)$$

The GMM distribution obtained by Gaussian clustering in Section III-A is used as the starting point for the optimization problem in (34).

Two model refinement approaches are investigated in this section: variational EM (VEM) and Monte Carlo based KL minimization on GMM (MCGMM).

1) Variational EM: The variational EM algorithm was first proposed in [35] and refined in [36]. Given two GMM distributions $f_1(x)$ and $f_2(x)$ with $f_1(x)$ being the reference distribution, VEM enables updating the parameters of $f_2(x)$ such that the KL divergence $D_{kl}(f_1\|f_2)$ is minimized, and the updated distribution $f_2(x)$ matches the reference $f_1(x)$ better. In the context of model refinement in (34), we assume that

$$f_1(x) = \sum_a \pi_a^{(1)} g_a^{(1)}(x) = \sum_a \pi_a^{(1)}\mathcal{N}\left(x;\mu_a^{(1)},\Sigma_a^{(1)}\right) \quad (35)$$

$$f_2(x) = \sum_b \pi_b^{(2)} g_b^{(2)}(x) = \sum_b \pi_b^{(2)}\mathcal{N}\left(x;\mu_b^{(2)},\Sigma_b^{(2)}\right). \quad (36)$$

Define

$$L(f_1\|f_2) = \int f_1(x)\log f_2(x)dx \quad (37)$$

which is not known in general for mixture models like GMMs, as it becomes

$$L(f_1\|f_2) = \sum_a \pi_a^{(1)}\int g_a^{(1)}(x)\log\sum_b \pi_b^{(2)} g_b^{(2)}(x)dx \quad (38)$$

where the integral $\int g_a^{(1)}\log\sum_b \pi_b^{(2)} g_b^{(2)}$ has no closed-form solution. A solution presented in [36] is to define a variational approximation to $L(f_1\|f_2)$. To derive a variational approximation to (38), variational parameters $\phi_{b|a}$ are introduced as a measure of the affinity between the Gaussian component $g_a^{(1)}$ of $f_1(x)$ and component $g_b^{(2)}$ of $f_2(x)$. The variational parameters must satisfy the constraints

$$\phi_{b|a} \geq 0 \quad \text{and} \quad \sum_b \phi_{b|a} = 1. \quad (39)$$

Using Jensen's inequality, a lower bound is obtained for (38)

$$L(f_1\|f_2) \geq \sum_a \pi_a^{(1)}\sum_b \phi_{b|a}\left(\log\frac{\pi_b^{(2)}}{\phi_{b|a}} + L\left(g_a^{(1)}\|g_b^{(2)}\right)\right) \triangleq \mathbb{L}_\phi(f_1\|f_2). \quad (40)$$

For model refinement, we can now update the parameters of $f_2(x)$ to match the reference model $f_1(x)$ by maximizing $\mathbb{L}_\phi(f_1\|f_2)$. This leads to a variational expectation-maximization (VEM) algorithm where $\mathbb{L}_\phi(f_1\|f_2)$ is first maximized with respect to (w.r.t.) ϕ. With ϕ fixed, $\mathbb{L}_\phi(f_1\|f_2)$ is then maximized w.r.t. the parameters of $f_2(x)$. In the E-step, one has

$$\widehat{\phi}_{b|a} = \frac{\pi_b^{(2)}\mathrm{e}^{-D_{kl}\left(g_a^{(1)}\|g_b^{(2)}\right)}}{\sum_{b'}\pi_{b'}^{(2)}\mathrm{e}^{-D_{kl}\left(g_a^{(1)}\|g_{b'}^{(2)}\right)}}. \quad (41)$$

which can be shown to give the best lower bound among $\phi_{b|a}$.

For a fixed $\phi_{b|a} = \widehat{\phi}_{b|a}$, it is now possible to find the parameters $\{\pi_b^{(2)}, \mu_b^{(2)}, \Sigma_b^{(2)}\}$ of $f_2(x)$ that maximize $\mathbb{L}_\phi(f_1\|f_2)$. This is the M-step with the following updates:

$$\pi_b^{(2)*} = \sum_a \pi_a^{(1)}\phi_{b|a}, \quad \mu_b^{(2)*} = \frac{\sum_a \pi_a^{(1)}\phi_{b|a}\mu_a^{(1)}}{\sum_{a'}\pi_{a'}^{(1)}\phi_{b|a'}} \quad (42)$$

$$\Sigma_b^{(2)*} = \frac{\sum_a \pi_a^{(1)}\phi_{b|a}\left[\Sigma_a^{(1)} + \left(\mu_a^{(1)} - \mu_b^{(2)*}\right)\left(\mu_a^{(1)} - \mu_b^{(2)*}\right)^\top\right]}{\sum_{a'}\pi_{a'}^{(1)}\phi_{b|a'}}. \quad (43)$$

The algorithm alternates between the E-step and M-step, increasing the variational likelihood in each step.

2) Monte Carlo Based KL Minimization on GMM: Monte Carlo (MC) methods are a class of computational algorithms that rely on repeated random sampling which are often used for simulation and numerical integration [37]–[39]. In particular, for the optimization problem posed in (34), one has

$$\begin{aligned} f_2^*(x) &= \arg\max_{f_2(x)} \int f_1(x) \log f_2(x) dx \\ &= \arg\max_{f_2(x)} \mathbf{E}_{f_1} \{\log f_2(x)\} \\ &\approx \arg\max_{f_2(x)} \frac{1}{M} \sum_{i=1}^{M} \log f_2(x_i) \end{aligned} \tag{44}$$

where the M samples $\{x_i\}_{i=1}^M$ are generated by sampling the reference distribution $f_1(x)$ which is the aggregated large GMM in the state. A closer inspection of the last step of (44) shows that the parameters of $f_2(x)$ maximize the log-likelihood of the M generated samples. Since $f_2(x)$ is a GMM distribution, the conventional EM algorithm [40] can be readily applied for the ML estimation starting from the downsized GMM distribution obtained by the Gaussian clustering.

C. Full to Diagonal Covariance Conversion

So far both the Gaussian clustering and model refinement are carried out in the full covariance space. Full covariance models are often not practical for real-world applications due to the computational cost. Hence, the models have to be converted to the diagonal covariance space for efficiency. Given the same number of Gaussian components, the conversion from the full covariance space to the diagonal covariance space will inevitably incur a loss of information for the description of the intrinsic structure of the data. In this section, we try to carry out the conversion by minimizing the KL divergence between the two GMM distributions before and after the conversion. This optimization problem follows an analogous mathematical treatment to (34). In this case, the reference distribution $f_1(x)$ is the refined full covariance GMM distribution obtained from Section III-B and $f_2(x)$ is the diagonal covariance GMM distribution to be optimized. Both $f_1(x)$ and $f_2(x)$ have the same number of Gaussian components.

A common practice for full to diagonal covariance conversion is the direct diagonalization operation on each Gaussian component

$$\Sigma_{\text{diag}} = \text{diag}(\Sigma_{\text{full}}) \tag{45}$$

where only the diagonal components of Σ_{full} are kept in Σ_{diag} while the off-diagonal elements are set to zero. It is trivial to prove that the diagonalization this way is equivalent to a Gaussian-component-wise minimization of the KL divergence between full and diagonal covariance Gaussians in the GMM distribution [41]. A better approach is to take into account the whole GMM distribution rather than each Gaussian component, again via the MC method. Analogous to (44), one has

$$\begin{aligned} f_2^*(x) &= \arg\max_{f_2(x)} \mathbf{E}_{f_1} \{\log f_2(x)\} \\ &\approx \arg\max_{f_2(x)} \frac{1}{M} \sum_{i=1}^{M} \log f_2(x_i) \end{aligned} \tag{46}$$

where x_i are samples drawn from $f_1(x)$. The ML estimate of the diagonal covariance GMM distribution $f_2(x)$ can be obtained by the EM algorithm. The Gaussian-component-wise diagonalization in (45) can serve as the initial estimate of $f_2(x)$ for the EM iterations.

D. Diagonal Covariance Model Refinement

After the conversion from the full covariance space to the diagonal covariance space, the model goes through another refinement in the diagonal covariance space. In this final step, the refinement is conducted by the Monte Carlo based KL minimization on HMM (MCHMM).

In [42], the computation of the Bhattacharyya divergence was extended to HMMs under the rationale that HMM is a generalization of GMM to sequences of observations. In this light, the mathematical treatments of the latent variables of the two (state sequence versus component index) are similar if an appropriate definition of the sequence distribution is provided. Following the definitions in [42], 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 diagonal covariance HMM to be refined, respectively. We want to optimize $f_2(x_{1:n})$ so that it minimizes the KL divergence with respect to the reference HMM:

$$\begin{aligned} f_2^*(x_{1:n}) &= \arg\min_{f_2(x_{1:n})} D_{kl}(f_1 \| f_2) \\ &= \arg\max_{f_2(x_{1:n})} \mathbf{E}_{f_1} \{\log f_2(x_{1:n})\} \\ &\approx \arg\max_{f_2(x_{1:n})} \frac{1}{M} \sum_{i=1}^{M} \log f_{2i}(x_{1:n}). \end{aligned} \tag{47}$$

From (47), HMM $f_2^*(x_{1:n})$ is an ML estimate using the M utterances drawn from the reference distribution $f_1(x_{1:n})$. Here we assume that HMM $f_1(x_{1:n})$ is the "ground truth" underlying distribution of the data and is approximated by the empirical distribution $\mathcal{F}(\mathcal{O}, W)$. Therefore, drawing $f_2^*(x_{1:n})$ for M sequences can be approximated by sampling utterances with repetition from the training data. In this work, the ensemble of all utterances (with repetition) from the bootstrapped subsets $\mathcal{L}_i^B, i = 1, \cdots, N$, is used as the M sequence samples in (47). Accordingly, the model refinement described by (47) is equal to the ML retraining of the diagonal covariance HMM using all the bootstrapped utterances starting from the converted diagonal model obtained from (46).

IV. Experimental Results

Experiments based on the proposed approach are conducted on Pashto and Dari, two major languages spoken in Afghanistan. The acoustic modeling of the two languages is addressed in the context of the DARPA TRANSTAC program for automatic real-time S2S translation. The deployment of stand-alone S2S systems (English–Pashto and English–Dari) is targeted for laptop computers or smartphones where speed and memory efficiency is as important as recognition accuracy. Both Pashto and Dari have limited annotated data for acoustic modeling. There are 150 hours of labeled data collected for each language in the final phase of the program and even less for the previous phases.

TABLE I
BREAKDOWN PERFORMANCE OF THE BSRS APPROACH ON THE PASHTO HELD-OUT DATA ON FULL COVARIANCE (FC) AGGREGATION, FULL COVARIANCE (FC) CLUSTERING, FULL COVARIANCE (FC) REFINEMENT, FULL-TO-DIAGONAL COVARIANCE (F2D) CONVERSION AND DIAGONAL COVARIANCE (DC) REFINEMENT. WORD ERROR RATES (WERS) AND CORRESPONDING MODEL SIZES (STATES/GAUSSIANS) ARE SHOWN UNDER 35 HOURS (35 h), 60 HOURS (60 h), 105 HOURS (105 h), AND 135 HOURS (135 h) OF TRAINING DATA, RESPECTIVELY. THREE CLUSTERING CRITERIA, KL DIVERGENCE (KL), ENTROPY (ENT), AND BAYES ERROR (BAY), ARE COMPARED FOR THE FC CLUSTERING. TWO REFINEMENT METHODS USING THE ENTROPY CLUSTERING CRITERION, VEM AND MCGMM, ARE COMPARED FOR THE FC REFINEMENT

Model		35h		60h		105h		135h	
		Size	WER	Size	WER	Size	WER	Size	WER
Baseline	ML1	1.2K/60K	**47.2**	2K/70K	**45.9**	3K/80K	**41.1**	3.5K/100K	**39.6**
	ML2	3K/60K	46.8	4K/70K	45.7	5K/80K	40.6	6K/100K	39.7
	ML3	3K/60K	47.0	4K/70K	45.8	5K/80K	40.5	6K/100K	39.5
FC Aggregation		3K/900K	**44.1**	4K/1.2M	**42.6**	5K/1.4M	**36.7**	6K/1.8M	**35.2**
FC Clustering	KL	3K/60K	44.2	4K/70K	42.9	5K/80K	37.3	6K/100K	36.0
	ENT	3K/60K	**44.2**	4K/70K	**42.9**	5K/80K	**37.1**	6K/100K	**35.8**
	BAY	3K/60K	44.2	4K/70K	42.9	5K/80K	37.3	6K/100K	36.0
FC Refinement (ENT)	VEM	3K/60K	44.1	4K/70K	42.8	5K/80K	37.1	6K/100K	35.8
	MCGMM	3K/60K	**44.2**	4K/70K	**42.8**	5K/80K	**37.0**	6K/100K	**35.7**
F2D MCGMM Conversion		3K/60K	**45.2**	4K/70K	**44.2**	5K/80K	**39.5**	6K/100K	**38.6**
DC MCHMM Refinement		3K/60K	**45.2**	4K/70K	**44.0**	5K/80K	**39.0**	6K/100K	**38.1**

The feature space is constructed by splicing nine 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. After the ML training, discriminative training is applied in both the feature space (FMMI [18]) and model space (BMMI [18]). This is the conventional state-of-the-art acoustic model training procedure which will be treated as baseline in this work. The proposed BSRS approach is applied in the same feature space. First, the original training data is bootstrapped into multiple subsets with each subset covering 70% of the original data $(\delta = 0.7)$. The subsets are then pooled together (with utterance repetition) for the training of LDA, STC and the decision tree. With more data in the ensemble, a deeper decision tree with more leaves is built. When the shared LDA, STC and the decision tree are in place, HMMs are trained on each subset, which is followed by the aggregation and restructuring. Once the ML model is obtained from the restructuring, discriminative training by FMMI and BMMI is performed. Since Pashto and Dari are morphologically rich, a hybrid lexicon with a blend of vowelized pronunciations and graphemes is used. The vowelization is manually generated while the graphemes are automatically created from spelling.

A. *Pashto Experiments*

There are two sets of test data used for the Pashto experiments. One is a held-out data set from the DARPA delivered data and the other is the TRANSTAC 2009 offline evaluation data. The held-out data set has about 10 hours of speech from 22 speakers, while the offline evaluation data has about 0.7 hours of speech data from 11 speakers. Acoustic models are trained using 35 hours, 60 hours, 105 hours, and 135 hours of data, respectively, to investigate the performance of the proposed method under various amounts of training data. The Viterbi decoder runs on a finite state graph which compiles the language model (LM), dictionary and decision tree into a static network for fast decoding [43]. The trigram LM consists of 1.2 M n-grams built on 26 K words with 30 K pronunciations. For BSRS, the original training data set is bootstrapped into 15 subsets.

Table I presents the breakdown performance on the held-out data set for the different stages of the proposed BSRS approach including full covariance (FC) aggregation, FC Gaussian clustering, FC model refinement, full-to-diagonal (F2D) covariance conversion, and diagonal covariance (DC) refinement. Word error rates (WERs) and the corresponding model sizes in terms of the numbers of states and Gaussians (states/Gaussians) are shown under 35 hours (35 h), 60 hours (60 h), 105 hours (105 h), and 135 hours (135 h) of training data, respectively.

Performance for the three baseline models (ML1, ML2, and ML3) is given in the first three rows of the table. ML1 is estimated on the original training data, which is considered as the baseline model in the conventional sense. In the BSRS approach, the decision tree is trained on the ensemble of bootstrapped data. Therefore, the absolute amount of data is larger with utterance repetitions. Consequently, the decision tree grows deeper. In the end, the BSRS model will have the same number of Gaussians as the baseline model after downsizing but with more states. To make a fair comparison, we also built baseline models with exactly the same size as the BSRS model in terms of both states and Gaussians, which are denoted by ML2 and ML3 in the baseline section. The distinction between the two is that ML2 is still trained using the original training data while ML3 is trained on the same ensemble of the bootstrapped data with utterance repetitions. The baseline HMMs are trained using the same initialization procedure as follows: A context independent HMM is first built by the K-means algorithm from the training data. Afterwards, a context dependent quinphone HMM is trained whose LDA, STC, and decision tree are built from the alignments generated by the context independent HMM. The quinphone HMM is then estimated by a total of 20 EM iterations with Gaussian mixture split. After the EM estimation is over, this quinphone HMM is used to regenerate the alignments of the training data for another round of quinphone HMM training with new LDA, STC and decision tree. The final HMM is obtained after 20 EM iterations. From the table, we can see that ML2 and ML3 are marginally better than ML1. The best case happens in ML3 using 105 hours of training data with the relative improvement of 1.5%. In most cases, the three baseline models yield almost

TABLE II
WORD ERROR RATES (WERS) OF THE PASHTO BASELINE AND BSRS MODELS TRAINED UNDER MAXIMUM-LIKELIHOOD (ML) CRITERION AND DISCRIMINATIVE CRITERION (FMMI + BMMI) ON HELD-OUT DATA SET AND OFFLINE EVALUATION DATA SET. THE MODELS ARE TRAINED USING 35 HOURS (35 h), 60 HOURS (60 h), 105 HOURS (105 h), AND 135 HOURS (135 h) OF TRAINING DATA

Model		35h		60h		105h		135h	
		heldout	offline	heldout	offline	heldout	offline	heldout	offline
Baseline	ML	47.2	44.0	45.9	41.3	41.4	38.0	39.6	36.5
	FMMI+BMMI	42.1	38.4	41.2	35.3	35.5	32.6	34.1	31.7
BSRS	Aggregated FC ML	44.1	39.8	42.6	38.5	36.7	35.0	35.2	33.6
	Restructured DC ML	45.2	42.3	44.0	39.7	39.0	36.6	38.1	35.6
	FMMI+BMMI	41.2	36.7	39.8	34.1	34.3	31.5	33.2	30.7

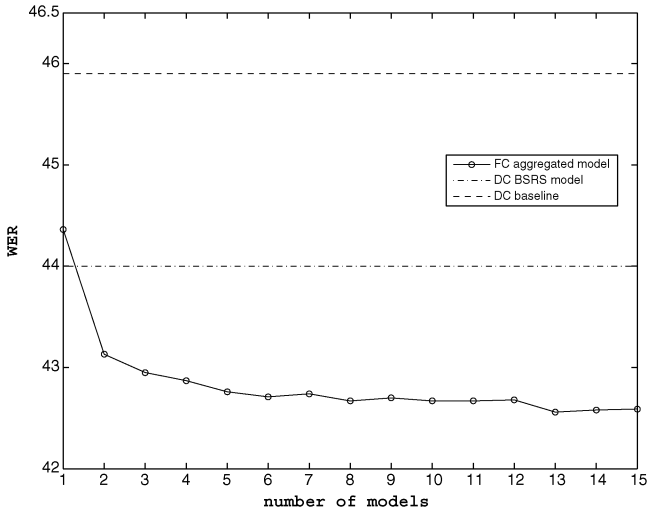


Fig. 2. WER(%) as a function of the number of Pashto full covariance subbagging models (solid line). Also shown in the figure are the WERs(%) of the diagonal covariance baseline (dashed line) and the diagonal covariance model after bootstrap and restructuring (BSRS, dash-dot line).

the same performance. Hence, given the sparse training data, simply increasing the model size or increasing the training data size by repeating the samples will not significantly help the performance.

Results of full covariance HMM aggregation are in the fourth row in Table I from which significant improvements (3%–4% absolute) can be observed. However, the aggregation also leads to a model of substantial size (900 K FC Gaussians for 35 h, 1.2 M FC Gaussians for 60 h, 1.4 M FC Gaussians for 105 h, and 1.8 M FC Gaussians for 135 h).

Following the aggregation performance in Table I are the results of the model restructuring process categorized by its four stages. First, in the FC Gaussian clustering, the performance of the three investigated clustering criteria, namely, KL divergence (KL), entropy (ENT), and Bayes error (BAY), is given. From the WERs, no criterion has obvious superiority over the others. Considering the computational advantage, the entropy criterion is chosen for the rest of the experiments. Second, the FC model refinement consists of performance using VEM and MCGMM which, according to the WERs, has no significant difference. Also from the computational perspective, the MCGMM method is used for the rest of the experiments. Third, the conversion from the full covariance model to the diagonal covariance model based on MCGMM is shown in the next row. This is where the loss of structural information occurs which understandably comes with a performance drop compared to the full covariance model after model refinement. Lastly, the last row shows the results after the MCHMM model refinement in the diagonal covariance space. Compared to the model right after the conversion, the refinement improves the performance in the 60 h, 105 h, and 135 h cases. Overall, by comparing the baseline model ML1 in the first row with the BSRS diagonal model in the last row, we see that the BSRS models have obtained 2%, 1.9%, 2.1%, and 1.5% absolute improvements for the 35 h, 60 h, 105 h, and 135 h training scenarios, respectively.

As an example, Fig. 2 illustrates the decrease of WERs of the aggregated full covariance models as the number of subbagging models increases from 1 to 15 (solid line) in the 60 h condition. The same figure also shows the WERs of the baseline diagonal covariance model (45.9%) and the final BSRS diagonal covariance model (44.0%). As can be viewed from the figure, most of the gain from the aggregation takes place in the first few models and the WER decreases only slowly afterwards. After model restructuring, the diagonal covariance BSRS model degrades from the aggregated full covariance model but is still significantly improved from the baseline model (1.9% absolute).

Table II summarizes the performance in WERs of the baseline models and BSRS models on both the held-out data set and the DARPA offline evaluation data set under the ML and discriminative training (FMMI+BMMI) with different amounts of data. The BSRS models after discriminative training have shown consistent improvements on both data sets. For the held-out set, the BSRS obtains 0.9%, 1.4%, 1.2%, and 0.9% absolute improvements after FMMI + BMMI on 35 h, 60 h, 105 h, and 135 h, respectively, and for the DARPA offline evaluation data, the gains are 1.7%, 1.2%, 1.1%, and 1.0% absolute on 35 h, 60 h, 105 h, and 135 h, respectively.

Although the final BSRS model has superior performance over the baseline model, the former has a model size similar to the latter. Since the numbers of Gaussians are the same for the two, the decoding speed is almost identical. Moreover, Table III shows the sizes of the static decoding graphs after compacting for both the baseline models and the BSRS models under the 35 h, 60 h, 105 h, and 135 h training scenarios. Even though the BSRS models have a relatively larger decision tree, the sizes of the graphs almost stay the same. As a result, no more memory consumption is required for the BSRS model at run-time.

B. Dari Experiments

To investigate the consistency of performance across languages, experiments have also been conducted on Dari. The acoustic models are trained with 125 hours of data. There

TABLE III
SIZES OF MODEL (STATES/GAUSSIANS) AND GRAPH (BYTES) OF THE PASHTO BASELINE MODELS AND BSRS MODELS TRAINED USING 35 HOURS (35 h), 60 HOURS (60 h), 105 HOURS (105 h), AND 135 HOURS (135 h) OF TRAINING DATA

Model	35h		60h		105h		135h	
	Size	Graph	Size	Graph	Size	Graph	Size	Graph
Baseline model	1.2K/60K	100M	2K/70K	102M	3K/80K	102M	3.5K/100K	102M
BSRS model	3K/60K	100M	4K/70K	101M	5K/80K	102M	6K/100K	103M

TABLE IV
WORD ERROR RATES (WERS), SIZES OF MODELS (STATES/GAUSSIANS) AND SIZES OF GRAPHS (BYTES) OF THE DARI BASELINE AND BSRS MODELS TRAINED UNDER MAXIMUM LIKELIHOOD (ML) CRITERION AND DISCRIMINATIVE CRITERION (FMMI + BMMI) ON HELD-OUT DATA SET AND OFFLINE EVALUATION DATA SET

Model		Size	Graph	WER	
				heldout	live
Baseline	ML	3K/60K	18.0M	46.0	38.8
	FMMI+BMMI	3K/60K		37.9	32.6
BSRS	Aggregated FC ML	5K/540K	18.2M	42.0	34.1
	Restructured DC ML	5K/60K		43.1	36.9
	FMMI+BMMI	5K/60K		36.5	31.3

are also two sets of test data used in the experiments. One is a held-out data set consisting of 11 hours of data from 25 speakers and the other is the DARPA live evaluation data composed of 1.4 hours of data from eight speakers. The trigram LM with 176 K n-grams is built on 46 K words and 48 K pronunciations. The acoustic modeling is carried out in the context of a smartphone S2S application whose stringent footprint requirement asks for more economical model and graph sizes in the experiments as compared to the Pashto counterpart.

Table IV presents the WERs of the Dari baseline models and the BSRS models under ML and discriminative training $(\text{FMMI} + \text{BMMI})$ on the two test sets. Also shown in the table are the sizes of the models in terms of the states and Gaussians and the sizes of the static decoding graphs. For the ML models, BSRS yields 2.9% absolute improvement for the held-out set and 1.9% absolute improvement for the live evaluation set. For the FMMI + BMMI models, BSRS yields 1.4% absolute improvement for the held-out set and 1.3% absolute improvement for the live evaluation set. On the other hand, the graphs from the baseline model and BSRS model are about the same size (18.2 M versus 18 M) and therefore they have almost identical memory consumption at run-time.

V. DISCUSSION

It can be shown that in theory bagging of randomized HMMs as sequence classifiers can lead to a variance reduction in the discriminant function (see Appendix B for the details). However, in practice, various approximations have to be made inevitably. First of all, the learning sets are bootstrapped based on the empirical distribution which concentrates mass $1/R$ at each sample point $(\mathcal{O}, W)$ to approximate the true underlying distribution $\mathcal{P}(\mathcal{O}, W)$. Second, the expectation of classifier over infinite number of learning sets $\mathcal{L}_i$ has to be approximated by a finite number of learning sets. The approximation of (7) using its upper bound by the convexity of the logarithm function in (8) is introduced mainly for computational convenience of the model restructuring. Although theoretically it will not always guarantee a smaller variance any more, it greatly simplifies the framework of multiple decoding and makes the following model restructuring step possible. It is the tradeoff made under the memory and CPU constraints. In the meantime, moving the expectation inside the logarithm function will always result in a higher likelihood from the model combination perspective.

Comparing to those truly well-resourced languages such as English, Mandarin, or modern standard Arabic (MSA) with thousands of hours of transcribed data (e.g., in the DARPA community), a language with about 150 hours of transcribed data still can be considered low-resourced. However, its data is not sparse any more if compared to a large number of truly under-resourced languages worldwide such as Vietnamese and Amharic reported in [3] and [1] in which case the availability of labeled data can come on the order of only a few hours. Hence, the concept of low-resourced language in this paper is relative. Bootstrap is typically known for its usefulness when dealing with insufficient data and one of the factors for the success of bagging is to require the randomized classifiers to be relatively weak and unstable. Specifically, to the proposed BSRS acoustic modeling investigated in this paper, when the training data increases the randomized HMMs will become relatively strong and as a consequence the gain from bagging will not be as significant. In general, bigger gains are obtained with less data as shown in Table II, especially for the ML models. Therefore, the proposed BSRS approach is expected to be most helpful when the training data is sparse. In the experiments conducted on Dari and Pashto, decent gains after discriminative training have been observed up to around 125 to 135 hours of training data. Therefore, the effectiveness of the BSRS approach can cover a wide range of low-resourced languages.

VI. SUMMARY

In this paper, we have proposed and investigated an acoustic modeling approach based on bootstrap and restructuring to cope with data sparsity. Experiments on Pashto and Dari have shown that the proposed approach can achieve superior performance to acoustic models trained using the conventional training procedure under both ML and discriminative training. This approach has demonstrated its effectiveness for up to 125 to 135 hours of training data in our experiments. In the meantime, the acoustic models trained by the proposed approach have almost the same decoding speed, memory consumption and response latency at run-time when compared to the acoustic models trained by the conventional training procedure.

APPENDIX A

Suppose distribution $f(x)$ is from the exponential family and $x \in \mathcal{R}^d$

$$f(x) = \frac{1}{Z(\theta)} e^{\theta^{\mathsf{T}}\Phi(x)} \tag{48}$$

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

$$Z(\theta) = \int e^{\theta^{\mathsf{T}}\Phi(x)} dx. \tag{49}$$

Then the Chernoff function

$$\begin{aligned} C(s) &= \int f_1(x)^s f_2(x)^{1-s} dx, \\ &= \int \frac{e^{s\theta_1^{\mathsf{T}}\Phi(x)}}{Z(\theta_1)^s} \cdot \frac{e^{(1-s)\theta_2^{\mathsf{T}}\Phi(x)}}{Z(\theta_2)^{1-s}} dx \\ &= \frac{Z\left(s\theta_1 + (1-s)\theta_2\right)}{Z(\theta_1)^s Z(\theta_2)^{1-s}}. \end{aligned} \tag{50}$$

Particularly, when $f(x)$ is Gaussian, let $\underline{P \triangleq \Sigma^{-1}}$ and $\underline{\psi \triangleq P\mu}$ be the exponential model parameters

$$\theta = \left[\mathrm{vec}(P)^{\mathsf{T}}, \psi^{\mathsf{T}}\right]^{\mathsf{T}} \tag{51}$$

$$\Phi(x) = \left[-\frac{1}{2}\mathrm{vec}(xx^{\mathsf{T}})^{\mathsf{T}}, x^{\mathsf{T}}\right]^{\mathsf{T}} \tag{52}$$

$$\log Z(\theta) = -\frac{1}{2}\left[d\log(2\pi) - \log|P| + \psi^{\mathsf{T}}P^{-1}\psi\right]. \tag{53}$$

Let $c(s) = \log C(s)$, then

$$c(s) = \log\left[Z\left(s\theta_1 + (1-s)\theta_2\right)\right] - s\log\left[Z(\theta_1)\right] - (1-s)\log\left[Z(\theta_2)\right]. \tag{54}$$

Since $\log Z(\theta)$ is convex with respect to θ, in fact,

$$\frac{\partial^2}{\partial\theta\partial\theta^{\mathsf{T}}} \log Z(\theta) = \mathrm{Cov}_\theta\left[\Phi(x)\right] \geq 0 \tag{55}$$

it is trivial to show that $c(s)$ is also a convex function of s whose minimum can be found using Newton's method

$$s_{k+1} = s_k - \frac{c'(s)}{c''(s)}. \tag{56}$$

Let $\theta = s\theta_1 + (1-s)\theta_2$ and $\underline{\Delta_\theta = \theta_1 - \theta_2}$, so that $\theta = \theta_2 + s\Delta_\theta$.

$$\begin{aligned} \frac{\partial}{\partial s} c(s) &= \frac{\partial}{\partial s}\log\left(Z(\theta)\right) + \log\frac{Z(\theta_2)}{Z(\theta_1)} \\ &= \Delta_\theta^T E_\theta\left[\Phi(x)\right] + \log\frac{Z(\theta_2)}{Z(\theta_1)}. \end{aligned} \tag{57}$$

When it is Gaussian distribution,

$$\begin{aligned} \Delta_\theta^{\mathsf{T}} E_\theta\left[\Phi(x)\right] &= -\frac{1}{2}\mathrm{trace}\left[\Delta_P(\Sigma + \mu\mu^{\mathsf{T}})\right] + \Delta_\psi^{\mathsf{T}}\mu \\ &= -\frac{1}{2}\mathrm{trace}(\Delta_P P^{-1}) - \frac{1}{2}\psi^{\mathsf{T}}P^{-1}\Delta_P P^{-1}\psi \\ &\quad + \Delta_\psi^{\mathsf{T}} P^{-1}\psi. \end{aligned} \tag{58}$$

Let $P = P_2 + s\Delta_P$ and $\psi = \psi_2 + s\Delta_\psi$:

$$\begin{aligned} P^{-1} &= (P_2 + s\Delta_P)^{-1} \\ &= P_2^{-\frac{1}{2}}\left(I + sP_2^{-\frac{1}{2}}\Delta_P P_2^{-\frac{1}{2}}\right)^{-1} P_2^{-\frac{1}{2}} \end{aligned} \tag{59}$$

where $P_2^{-1/2}\Delta_P P_2^{-1/2}$ is symmetric. So one has the eigenvalue decomposition

$$\underline{P_2^{-\frac{1}{2}}\Delta_P P_2^{-\frac{1}{2}} = Q^{\mathsf{T}}\Lambda Q} \tag{60}$$

with eigenvalues ξ_i on the diagonal of Λ and eigenvectors in Q.

Then one has

$$\begin{aligned} \mathrm{trace}(\Delta_P P^{-1}) &= \mathrm{trace}\left[\Delta_P P_2^{-\frac{1}{2}}\left(I + sP_2^{-\frac{1}{2}}\Delta_P P_2^{-\frac{1}{2}}\right)^{-1} P_2^{-\frac{1}{2}}\right] \\ &= \mathrm{trace}\left[\Lambda(I+s\Lambda)^{-1} Q P_2^{-\frac{1}{2}} P_2^{\frac{1}{2}} Q^{\mathsf{T}}\right] \\ &= \mathrm{trace}\left[\Lambda(I+s\Lambda)^{-1}\right] = \sum_{i=1}^{d} \frac{\xi_i}{1+s\xi_i} \end{aligned} \tag{61}$$

$$\begin{aligned} \Delta_\psi^{\mathsf{T}} P^{-1}\psi &= \Delta_\psi^{\mathsf{T}} P^{-1}\psi_2 + s\Delta_\psi^{\mathsf{T}} P^{-1}\Delta_\psi \\ &= \Delta_\psi^{\mathsf{T}} P_2^{-\frac{1}{2}} Q^{\mathsf{T}}(I+s\Lambda)^{-1} Q P_2^{-\frac{1}{2}}\psi_2 \\ &\quad + s\Delta_\psi^{\mathsf{T}} P_2^{-\frac{1}{2}} Q^{\mathsf{T}}(I+s\Lambda)^{-1} Q P_2^{-\frac{1}{2}}\Delta_\psi \\ &= u^{\mathsf{T}}(I+s\Lambda)^{-1} v + s u^{\mathsf{T}}(I+s\Lambda)^{-1} u \\ &= \sum_{i=1}^{d} \frac{u_i v_i + s u_i^2}{1+s\xi_i} \end{aligned} \tag{62}$$

where $\underline{u = QP_2^{-1/2}\Delta_\psi}$ and $\underline{v = QP_2^{-1/2}\psi_2}$

$$\begin{aligned} \psi^{\mathsf{T}} P^{-1}\Delta_P P^{-1}\psi &= \psi^{\mathsf{T}} P_2^{-\frac{1}{2}} Q^{\mathsf{T}}(I+s\Lambda)^{-1} Q P_2^{-\frac{1}{2}}\Delta_P P_2^{-\frac{1}{2}} Q^{\mathsf{T}} \\ &\quad \times (I+s\Lambda)^{-1} Q P_2^{-\frac{1}{2}}\psi \\ &= (\psi_2 + s\Delta_\psi)^{\mathsf{T}} P_2^{-\frac{1}{2}} Q^{\mathsf{T}}(I+s\Lambda)^{-1}\Lambda \\ &\quad \times (I+s\Lambda)^{-1} Q P_2^{-\frac{1}{2}}(\psi_2 + s\Delta_\psi) \\ &= (v+su)^{\mathsf{T}}(I+s\Lambda)^{-1}\Lambda(I+s\Lambda)^{-1}(v+su) \\ &= \sum_{i=1}^{d} \frac{\xi_i(v_i+su_i)^2}{(1+s\xi_i)^2} \end{aligned} \tag{63}$$

From (61), (62), and (63), one has

$$\begin{aligned} \frac{\partial}{\partial s}\log Z(\theta) &= -\frac{1}{2}\sum_{i=1}^{d}\frac{\xi_i}{1+s\xi_i} - \frac{1}{2}\sum_{i=1}^{d}\frac{\xi_i(v_i+su_i)^2}{(1+s\xi_i)^2} \\ &\quad + \sum_{i=1}^{d}\frac{u_i v_i + s u_i^2}{1+s\xi_i} \\ &= \sum_{i=1}^{d}\left[\frac{u_i v_i + s u_i^2 - \frac{1}{2}\xi_i}{1+s\xi_i} - \frac{\frac{1}{2}\xi_i(v_i+su_i)^2}{(1+s\xi_i)^2}\right] \end{aligned} \tag{64}$$

and

$$\begin{aligned} &\frac{\partial}{\partial s^2}\log^2 Z(\theta) \\ &\quad = \sum_{i=1}^{d}\left[\frac{u_i^2}{1+s\xi_i} - \frac{2\xi_i u_i v_i + 2s\xi_i u_i^2 - \frac{1}{2}\xi_i^2}{(1+s\xi_i)^2} + \frac{\xi_i^2(v_i+su_i)^2}{(1+s\xi_i)^3}\right]. \end{aligned} \tag{65}$$

Therefore,

$$c'(s) = \log\frac{Z(\theta_2)}{Z(\theta_1)} + \sum_{i=1}^{d}\left[\frac{u_i v_i + s u_i^2 - \frac{1}{2}\xi_i}{1+s\xi_i} - \frac{\frac{1}{2}\xi_i(v_i+su_i)^2}{(1+s\xi_i)^2}\right] \quad (66)$$

$$c''(s) = \sum_{i=1}^{d}\left[\frac{u_i^2}{1+s\xi_i} - \frac{2\xi_i u_i v_i + 2s\xi_i u_i^2 - \frac{1}{2}\xi_i^2}{(1+s\xi_i)^2} + \frac{\xi_i^2(v_i+su_i)^2}{(1+s\xi_i)^3}\right]. \quad (67)$$

To summarize,

$$\Delta_p = P_1 - P_2 = \Sigma_1^{-1} - \Sigma_2^{-1} \quad (68)$$

$$\Delta_\psi = \psi_1 - \psi_2 = \Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2 \quad (69)$$

$$P_2^{-\frac{1}{2}}\Delta_P P_2^{-\frac{1}{2}} = Q^\top \Lambda Q \quad (70)$$

$$u = QP_2^{-\frac{1}{2}}\Delta_\psi \quad (71)$$

$$v = QP_2^{-\frac{1}{2}}\psi_2 \quad (72)$$

where Q and $\Lambda = \mathrm{diag}\{\xi_1, \cdots, \xi_i, \cdots, \xi_d\}$ are the matrices composed of eigenvectors and eigenvalues of $P_2^{-1/2}\Delta_p P_2^{-1/2}$.

Appendix B

Suppose the true model θ_{true} lies in the HMM parameter space λ under discussion. This is obviously a raw assumption since HMM is only an approximation in modeling and most likely θ_{true} does not belong to the family λ. However, under certain conditions [44], there exists a classifier in the HMM parameter space that is consistent for classification (CFC). For such a CFC classifier, despite the fact that the true model does not belong to the parameter family under consideration it still has the same decoding output as the optimal decoder defined by the true model. Therefore, this CFC classifier can still lead to the best Bayesian accuracy from the decoder perspective. In practical scenarios, when the conditions for CFC do not hold, the estimated HMM λ is considered as an approximation to the CFC classifier. For notational simplicity, θ_{true} will be used in the derivation below.

Given the label sequence S, let the score of the discriminant function under the "true" model (or the CFC decoder) be $\tilde{Y}$, then one has

$$\tilde{Y} = D_{\theta_{\mathsf{true}}}(\mathcal{O}, S) = \sum_{t=1}^{T}\log\tilde{f}_{\mathsf{s}_t}(O_t) + \sum_{t=2}^{T}\log\tilde{a}_{s_{t-1}s_t} + \log\tilde{\pi}_{\mathsf{s}_1}. \quad (73)$$

Assume one aggregates randomized HMM classifiers estimated from learning sets $\mathcal{L}$ sampled from the true underlying distribution. The aggregated HMM classifier has the discriminant function computed as

$$D_A(\mathcal{O}, S) = E_{\mathcal{L}}\left[D_{\lambda,\mathcal{L}}(\mathcal{O}, S)\right] = \sum_{t=1}^{T}E_{\mathcal{L}}\log\left[f_{\mathsf{s}_t}(O_t)\right] + \sum_{t=2}^{T}\log a_{s_{t-1}s_t} + \log\pi_{\mathsf{s}_1} \quad (74)$$

where the initial and transition probabilities are assumed to be the same for $\mathcal{L}$.

The mean square error of the discriminant function score with respect to $\tilde{Y}$ is

$$e = E_{\mathcal{L}}E_{\mathcal{O},S}\left[\tilde{Y} - D_{\lambda,\mathcal{L}}(\mathcal{O}, S)\right]^2 \quad (75)$$

while the mean square error of the aggregated discriminant function score with respect to $\tilde{Y}$ is

$$e_A = E_{\mathcal{O},S}\left[\tilde{Y} - D_A(\mathcal{O}, S)\right]^2. \quad (76)$$

Following the same derivation in [13], it can be shown that

$$e \geq e_A. \quad (77)$$

In fact, from (75)

$$\begin{aligned} e &= E_{\mathcal{L}}E_{\mathcal{O},S}\left[\tilde{Y} - D_{\lambda,\mathcal{L}}(\mathcal{O}, S)\right]^2 \\ &= E_{\mathcal{O},S}[\tilde{Y}^2] - 2E_{\mathcal{O},S}[\tilde{Y}E_{\mathcal{L}}\left[D_{\lambda,\mathcal{L}}(\mathcal{O}, S)\right]] \\ &\quad + E_{\mathcal{O},S}E_{\mathcal{L}}\left[D_{\lambda,\mathcal{L}}(\mathcal{O}, S)^2\right]. \end{aligned} \quad (78)$$

Using the inequality $EZ^2 \geq (EZ)^2$ for a random variable Z, one has

$$E_{\mathcal{L}}\left[D_{\lambda,\mathcal{L}}(\mathcal{O}, S)^2\right] \geq \left(E_{\mathcal{L}}\left[D_{\lambda,\mathcal{L}}(\mathcal{O}, S)\right]\right)^2. \quad (79)$$

Therefore,

$$\begin{aligned} e &= E_{\mathcal{O},S}[\tilde{Y}^2] - 2E_{\mathcal{O},S}\left[\tilde{Y}D_A(\mathcal{O}, S)\right] \\ &\quad + E_{\mathcal{O},S}E_{\mathcal{L}}\left[D_{\lambda,\mathcal{L}}(\mathcal{O}, S)^2\right] \\ &\geq E_{\mathcal{O},S}[\tilde{Y}^2] - 2E_{\mathcal{O},S}\left[\tilde{Y}D_A(\mathcal{O}, S)\right] \\ &\quad + E_{\mathcal{O},S}\left(E_{\mathcal{L}}\left[D_{\lambda,\mathcal{L}}(\mathcal{O}, S)\right]\right)^2 \\ &= E_{\mathcal{O},S}[\tilde{Y}^2] - 2E_{\mathcal{O},S}\left[\tilde{Y}D_A(\mathcal{O}, S)\right] + E_{\mathcal{O},S}\left[D_A(\mathcal{O}, S)\right]^2 \\ &= E_{\mathcal{O},S}\left[\tilde{Y} - D_A(\mathcal{O}, S)\right]^2 = e_A. \end{aligned} \quad (80)$$

References

[1] S. T. Abate and W. Menzel, "Automatic speech recognition for an under-resourced language—Amharic," in *Proc. Interspeech*, 2007, pp. 1541–1544.

[2] T. Pellegrini and L. Lamel, "Using phonetic features in unsupervised word decompounding for ASR with application to a less-represented language," in *Proc. Interspeech*, 2007, pp. 1797–1800.

[3] V.-B. Le and L. Besacier, "Automatic speech recognition for under-resourced languages: Application to Vietnamese language," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 8, pp. 1471–1482, Nov. 2009.

[4] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Commun.*, vol. 35, pp. 31–51, 2001.

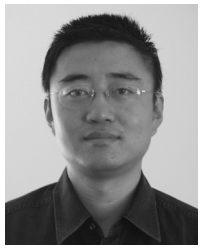
[5] C. Nieuwoudt and E. C. Botha, "Cross-language use of acoustic information for automatic speech recognition," *Speech Commun.*, vol. 38, pp. 101–113, 2002.

[6] C. V. Heerden, N. Kleynhans, E. Barnard, and M. Davel, "Pooling ASR data for closely related languages," in *Proc. Workshop Spoken Lang. Technol. for Under-Resourced Lang. (SLTU)*, 2010, pp. 17–23.

[7] T. Schultz and K. Kirchhoff, *Multilingual Speech Processing*. New York: Elsevier, Academic, 2006.

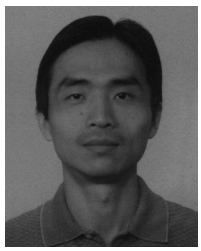
[8] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 1, no. 1, pp. 1–26, 1979.

[9] B. Efron, "Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods," *Biometrika*, vol. 68, no. 3, pp. 589–599, 1981.
[10] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. Boca Raton, FL: Chapman & Hall/CRC, 1993.
[11] A. Davison, D. V. Hinkley, and A. Canty, *Bootstrap Methods and Their Application*. Cambridge, U.K.: Cambridge Univ. Press, 1999.
[12] A. M. Zoubir and D. R. Iskander, *Bootstrap Techniques for Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[13] L. Breiman, "Baggging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[14] L. Breiman, "Heuristics of instability and stabilization in model selection," *Ann. Statist.*, vol. 24, no. 6, pp. 2350–2383, 1996.
[15] X. Cui, J. Xue, P. L. Dognin, U. V. Chaudhari, and B. Zhou, "Acoustic modeling with bootstrap and restructuring for low-resourced languages," in *Proc. Interspeech*, 2010, pp. 2974–2977.
[16] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 2, pp. 179–190, Feb. 1983.
[17] Y. Normandin, "Hidden Markov models, maximum mutual information estimation and the speech recognition problem," Ph.D dissertation, McGill Univ., Montreal, QC, Canada, 1991.
[18] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 4057–4060.
[19] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D dissertation, Univ. of Cambridge, Cambridge, U.K., 2003.
[20] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[21] P. Bühlmann and B. Yu, "Analyzing bagging," *Ann. Statist.*, vol. 30, no. 4, pp. 927–961, 2002.
[22] F. Zaman and H. Hirose, "Effect of subsampling rate on subbagging and related ensembles of stable classifiers," in *Proc. Int. Conf. Pattern Recogn. Mach. Intell.*, 2009, pp. 44–49.
[23] A. Elisseeff, T. Evgeniou, and M. Pontil, "Stability of randomized learning algorithms," *J. Mach. Learn. Res.*, vol. 6, pp. 55–79, 2005.
[24] X. Liu and M. Gales, "Automatic model complexity control using marginalized discriminative growth functions," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1414–1424, May 2007.
[25] M. Padmanabhan and L. R. Bahl, "Model complexity adaptation using a discriminant measure," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 2, pp. 205–208, Mar. 2000.
[26] G. F. G. Yared, F. Violaro, and L. C. Sousa, "Gaussian elimination algorithm for HMM complexity reduction in continuous speech recognition systems," in *Proc. Interspeech*, 2005, pp. 377–380.
[27] K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous model complexity control by MDL principle," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 1995, pp. 717–720.
[28] X. He and Y. Zhao, "Model complexity optimization for nonnative English speakers," in *Proc. Interspeech*, 2001, pp. 1461–1464.
[29] C. Hennig, "Methods for merging gaussian mixture components," *Adv. Data Anal. Classific.*, vol. 4, no. 1, pp. 3–34, 2010.
[30] O. Bellot, D. Matrouf, P. Nocera, G. Linares, and J.-F. Bonastre, "Structural speaker adaptation using maximum a posteriori approach and a Gaussian distributions merging technique," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2003, pp. 121–124.
[31] W. Xu, J. Duchateau, K. Demuynck, and I. Dologlou, "A new approach to merging Gaussian densities in large vocabulary continuous speech recognition," in *Proc. IEEE Benelux Signal Process. Symp.*, 1998, pp. 231–234.
[32] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.
[33] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.
[34] X. Chen, X. Cui, J. Xue, P. A. Olsen, J. R. Hershey, and B. Zhou, "Full covariance bootstrapped acoustic model clustering," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 4496–4499.
[35] P. L. Dognin, J. R. Hershey, V. Goel, and P. A. Olsen, "Refactoring acoustic models using variational density approximation," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 4473–4476.
[36] P. L. Dognin, J. R. Hershey, V. Goel, and P. A. Olsen, "Refactoring acoustic models using variational Expectation-Maximization," in *Proc. Interspeech*, 2009, pp. 212–215.
[37] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*. London, U.K.: Methuen, 1975.
[38] G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*. New York: Springer, 1995.
[39] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Mont Carlo Method*, 2nd ed. New York: Wiley, 2007.
[40] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
[41] X. Cui, X. Chen, J. Xue, P. A. Olsen, J. R. Hershey, and B. Zhou, "Acoustic modeling with bootstrap and restructuring based on full covariance," in *Proc. Interspeech*, 2011, pp. 1697–1700.
[42] J. R. Hershey and P. A. Olsen, "Variational Bhattacharyya divergence for hidden Markov models," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 4557–4560.
[43] G. Saon, D. Povey, and G. Zweig, "Anatomy of an extremely fast LVCSR decoder," in *Proc. Interspeech*, 2005, pp. 549–552.
[44] P. S. Gopalakrishnan, D. Kotievsky, A. Nadas, D. Nahanloo, and M. A. Pichieny, "Decoder selection based on cross-entropies," in *Int. Conf. Acoust., Speech, Signal Process.*, 1988, pp. 20–23.



Xiaodong Cui received the B.S. degree (with highest honors) from Shanghai Jiao Tong University, Shanghai, China, in 1996, the M.S. degree from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from University of California, Los Angeles, in 2005, all in electrical engineering.

From 2005 to 2006, he was a Research Staff Member at DSP Solutions R&D Center, Texas Instruments, Dallas, TX, focusing on noise-robust issues for embedded speech recognition systems. Since 2006, he has been a Research Staff Member at Human Language Technologies, IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include multilingual speech-to-speech translation, speech recognition (particularly acoustic modeling), digital speech processing, statistical signal processing, machine learning, and pattern recognition.

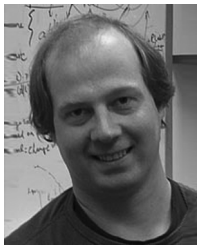


Jian Xue received the B.S. and M.S. degrees in electrical engineering from Southeast University, Nanjing, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from the University of Missouri, Columbia, in 2007.

He is currently a Speech Scientist with the IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include automatic speech recognition, speech-to-speech translation, machine learning, and digital signal processing.

Xin Chen received the B.S. degree in computer science from Tianjin University, Tianjin, China, in 2006, and the M.S. and Ph.D. degrees in computer science from the University of Missouri, Columbia, in 2008 and 2011, respectively.

He is currently a Research Scientist with the Pearson, Knowledge Technology Group, Menlo Park, CA. His research interests include acoustic modeling, automatic speech recognition, natural language processing, and machine learning.



Peder A. Olsen received the Ph.D. degree in mathematics from the University of Michigan, Ann Arbor, in 1996.

He is a Research Staff Member at the IBM T. J. Watson Research Center, Yorktown Heights, NY. After the Ph.D. degree, he joined the speech recognition group at IBM, where he has worked on acoustic modeling, noise robustness, speech separation, and large vocabulary speech recognition. Since April 2012, he has been working in the Business Analytics and Mathematics Department, IBM Research.

Pierre L. Dognin received the M.S. and Ph.D. degrees in electrical engineering from the University of Pittsburgh, Pittsburgh, PA, in 1999 and 2003, respectively.

From 2001 to 2003, he completed his Ph.D. work as a Visiting Researcher at BBN Technologies, Cambridge, MA. He then joined the Human Language Technologies (HLT) Department, IBM T. J. Watson Research Center, Yorktown Heights, NY. At IBM, he has been working on acoustic modeling, robust speech recognition, and core engine technologies. His research interests include statistical modeling, graphical models, machine learning, and speech technologies.

Upendra V. Chaudhari received the S.B., S.M., and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, in 1991, 1993, and 1997, respectively, where the main focus of his research was in the field of statistical signal processing and non-orthogonal signal expansions.

He became a Post-Doctoral Associate at MIT after finishing the Ph.D. degree and continued work on signal expansions, particularly for speech. While at MIT, he also worked on computational complexity, multi-access communication theory, and information retrieval. Later, he moved to the Human Language Technologies Group at the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he is a Research Staff Member. His work centers on statistical modeling and signal processing, with applications in speech and speaker recognition, meta-data modeling, and information retrieval. Recent work focuses on efficient methods for approximate search in audio to recover from recognition errors and handle vocabulary mismatch.

John R. Hershey received the Ph.D. degree from the Department of Cognitive Science, University of California at San Diego (UCSD), La Jolla. His thesis explored the use of generative graphical models for speech enhancement, face-tracking and combinations of the two.

He was a founding member of the Machine Perception Laboratory, UCSD. While at UCSD, he interned at Microsoft Research and at Mitsubishi Electric Research Labs (MERL), Cambridge, MA. In 2004, He was a Visiting Researcher in the Speech Group at Microsoft Research. In 2005, he moved to the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he was a Research Staff Member in the Speech Algorithms and Engines Group, and Team Leader of the noise robustness project in collaboration with Nuance Communications. In 2010, he moved to MERL, where he leads the speech and audio team, and works on research projects in the area of speech and audio signal separation, voice search, language processing, and user interfaces.

Bowen Zhou (M'03) received the Ph.D. degree from the University of Colorado at Boulder in 2003.

He has been with the IBM T. J. Watson Research Center, Yorktown Heights, NY, since 2003, where he is currently a Research Staff Member and manages the Multilingual Translation and Learning Department. He has been a key technical member and leader to IBM Research's speech-to-speech translation projects spanning areas of speech recognition, text-to-speech synthesis, and machine translation. He has also been a principal investigator for DARPA TRANSTAC and Transformative Apps programs, and currently also leads the Chinese–English machine translation efforts at IBM Research. He has authored and coauthored over 60 papers on top conferences and journals. He has a broad interest in natural language processing including both speech and text, and his recent interest includes statistical machine translation, machine learning and optimization, and large-scale data analytics.

Dr. Zhou is a member of the IEEE Speech and Language Technical Committee, and also served as area cochair of Machine Translation for NAACL/HLT 2012.