Tied Probabilistic Linear Discriminant Analysis for Speech Recognition

Tied Probabilistic Linear Discriminant Analysis for Speech Recognition

Liang Lu and Steve Renals

***Abstract*—Acoustic models using probabilistic linear discriminant analysis (PLDA) capture the correlations within feature vectors using subspaces which do not vastly expand the model. This allows high dimensional and correlated feature spaces to be used, without requiring the estimation of multiple high dimension covariance matrices. In this letter we extend the recently presented PLDA mixture model for speech recognition through a tied PLDA approach, which is better able to control the model size to avoid overfitting. We carried out experiments uisng the Switchboard corpus, with both mel frequency cepstral coefficient features and bottleneck feature derived from a deep neural network. Reductions in word error rate were obtained by using tied PLDA, compared with the PLDA mixture model, subspace Gaussian mixture models, and deep neural networks.**

***Index Terms*—acoustic modelling, probabilistic linear discriminant analysis, parameters tying**

I. Introduction

ACOUSTIC models for speech recognition have advanced substantially over the past 25 years, but the front-end feature processing has been largely unchanged, based on mel frequency cepstral coefficients (MFCCs) [1] and perceptual linear prediction (PLP) features [2]. To a large degree this has been due to the use of acoustic models based on hidden Markov models (HMMs) with Gaussian mixture models (GMMs) [3]–[5], which are well matched to feature representations which have decorrelated components and are relatively low-dimensional.

Deep neeural network (DNN) acoustic models [6] address these limitations and have achieved significant reductions in word error rate (WER) across many speech recogniiton datasets [7]. Compared to the hybrid neural network / hidden Markov model (HMM) architecture studied in the early 1990s [8], [9], DNNs typically use more hidden layers and a wider output layer. Moreover, DNNs can be also used as a good feature extractor, for instance through the inference of bottleneck features which may append the features used in GMM-based speech recognition systems [10], [11]. However, in order to be compatible with GMMs using diagonal covariances, such augmented feature vectors must typically be relatively low-dimensional and decorrelated.

We have addressed the limitations of GMMs through an acoustic model based on probabilistic linear discriminant analysis (PLDA) [12], which can employ higher dimensional, correlated feature vectors. PLDA is a probabilistic extension of linear discriminant analysis (LDA) [13], which has been very well studied for speaker recognition in the joint factor analysis (JFA) [14] and i-vector [15]–[17] frameworks. A PLDA acoustic model factorizes the acoustic variability using HMM state dependent variables which are expected to be consistent across different acoustic conditions, and observation dependent variables which characterise per frame level acoustic changes [12]. Similarly to a subspace GMM (SGMM) [18], the factorisation is based on the inference of subspaces. However, while the SGMM uses a set of full covariance matrices to directly model the per frame acoustic variability, the PLDA model introduces another set of projections to model this variability in lower-dimension subspaces.

We have previously investigated using a PLDA mixture model for acoustic modelling [12], [19]. Though good results have been obtained, this model has a large number of HMM state dependent variables, and is thus prone to overfitting. In this letter we mitigate the problem by tying the PLDA state variables in PLDA, an approach analogous to the use of tied state vectors in SGMMs [18].

II. PLDA-based Acoustic Model

The PLDA-based acoustic model is a generative model in which the distribution over acoustic feature vectors $\mathbf{y}_t \in \mathbb{R}^d$ from the j-th HMM state at time t is expressed as:

$$\mathbf{y}_t|j = \mathbf{U}\mathbf{x}_{jt} + \mathbf{G}\mathbf{z}_j + \mathbf{b} + \epsilon_{jt}, \quad \epsilon_{jt} \sim \mathcal{N}(\mathbf{0}, \Lambda)\,. \quad (1)$$

$\mathbf{z}_j \in \mathbb{R}^q$ is the state variable (equivalent to the between-class identity variable in JFA) shared by the whole set of acoustic frames generated by the j-th state and $\mathbf{x}_{jt} \in \mathbb{R}^p$ is the frame variable (equivalent to the within-class channel variable in JFA) which explains the per-frame variability. Usually, the dimensionality of these two latent variables is smaller than that of the feature vector $\mathbf{y}_t$, i.e. $p, q \leq d$. $\mathbf{U} \in \mathbb{R}^{d\times p}$ and $\mathbf{G} \in \mathbb{R}^{d\times q}$ are two low rank matrices which span the subspaces to capture the major variations for $\mathbf{x}_{jt}$ and $\mathbf{z}_j$ respectively. They are analogous to the within-class and between-class subspaces in the standard LDA formulation, but are estimated probabilistically. $\mathbf{b} \in \mathbb{R}^d$ denotes the bias and $\epsilon_{jt} \in \mathbb{R}^d$ is the residual noise which is assumed to be Gaussian with zero mean and diagonal covariance. By marginalising out the residual noise variable ϵ_{jt}, we obtain the following likelihood function:

$$p(\mathbf{y}_t|\mathbf{x}_{jt}, \mathbf{z}_j, j) = \mathcal{N}(\mathbf{y}_t; \mathbf{U}\mathbf{x}_{jt} + \mathbf{G}\mathbf{z}_j + \mathbf{b}, \Lambda) \quad (2)$$

Liang Lu, and Steve Renals are with University of Edinburgh, UK; email: {liang.lu, s.renals}@ed.ac.uk

The research was supported by EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

A. PLDA Mixture Model

A single PLDA has a limited modelling capacity since it only approximates a single Gaussian distribution. An M-component PLDA mixture model [12] results in the following component distribution:

$$\mathbf{y}_t|j,m = \mathbf{U}_m\mathbf{x}_{jmt} + \mathbf{G}_m\mathbf{z}_{jm} + \mathbf{b}_m + \epsilon_{jmt}, \quad (3)$$

$$\epsilon_{jmt} \sim \mathcal{N}(\mathbf{0}, \Lambda_m) \quad (4)$$

If c to be the component indicator variable, then the prior (weight) of each component is $P(c=m|j) = \pi_{jm}$. Given the latent variables $\mathbf{x}_{jmt}$ and $\mathbf{z}_{jm}$, the state-level distribution over features is:

$$p(\mathbf{y}_t|j) = \sum_m \pi_{jm}\mathcal{N}(\mathbf{y}_t; \mathbf{U}_m\bar{\mathbf{x}}_{jmt} + \mathbf{G}_m\bar{\mathbf{z}}_{jm} + \mathbf{b}_m, \Lambda_m)\,.$$

$\bar{\mathbf{x}}_{jmt}$ and $\bar{\mathbf{z}}_{jm}$ are point estimates of the latent variables. Since the projection matrices $\mathbf{U}_m$ and $\mathbf{G}_m$ are globally shared, a large number of components can be used to improve the model capacity, e.g. $M = 400$ [12].

B. Tied PLDA

To avoid overfitting in the PLDA mixture model, those components which are responsible for a small number of feature vectors may be deactivated. Alternatively, the state variables $\mathbf{z}_{jm}$ may be tied across components, resulting in the following component distribution:

$$\mathbf{y}_t|j,m = \mathbf{U}_m\mathbf{x}_{jmt} + \mathbf{G}_m\mathbf{z}_j + \mathbf{b}_m + \epsilon_{jmt}, \quad (5)$$

$$\epsilon_{jmt} \sim \mathcal{N}(\mathbf{0}, \Lambda_m)\,. \quad (6)$$

Tying the state variables may over-simplify the model. In this case, a "mixing-up" strategy can be used, analogous to SGMM sub-state splitting [18]:

$$\mathbf{y}_t|j,k,m = \mathbf{U}_m\mathbf{x}_{jkmt} + \mathbf{G}_m\mathbf{z}_{jk} + \mathbf{b}_m + \epsilon_{jkmt}, \quad (7)$$

$$\epsilon_{jkmt} \sim \mathcal{N}(\mathbf{0}, \Lambda_m)\,, \quad (8)$$

where k denotes the sub-state index, and $\mathbf{z}_{jk}$ is the sub-state variable. This makes Tied PLDA model more scalable compared to PLDA mixture model as we can balance the number of the sub-state variables according to the amount of available training data. Tied PLDA is equivalent to SGMM if we remove the per-frame latent variable x_{jkmt} and use full covariance Λ_m to model the residual noise. Given the latent variables, the state-level likelihood function can be written as

$$p(\mathbf{y}_t|j) = \sum_{mk} c_{jk} \times \pi_{jm}\mathcal{N}(\mathbf{y}_t; \mathbf{U}_m\bar{\mathbf{x}}_{jmkt} + \mathbf{G}_m\bar{\mathbf{z}}_{jk} + \mathbf{b}_m, \Lambda_m)$$

$$= \sum_{mk} w_{jkm}\mathcal{N}(\mathbf{y}_t; \mathbf{U}_m\bar{\mathbf{x}}_{jmkt} + \mathbf{G}_m\bar{\mathbf{z}}_{jk} + \mathbf{b}_m, \Lambda_m) \quad (9)$$

where c_{jk} is the sub-state weight, π_{jm} is the component weight which is shared for all the sub-state models, and $w_{jkm} = c_{jk} \times \pi_{jm}$. This is different to an SGMM in which a weight projection matrix is used to derive the component-dependent weights:

$$p^{\text{SGMM}}(\mathbf{y}_t|j) = \sum_k c_{jk} \sum_m \pi_{jkm}\mathcal{N}(\mathbf{y}_t; \mathbf{G}_m\bar{\mathbf{z}}_{jk}, \mathbf{\Sigma}_m) \quad (10)$$

$$\pi^{\text{SGMM}}_{jkm} = \frac{\exp \mathbf{w}_m^T\bar{\mathbf{z}}_{jk}}{\sum_{m'} \exp \mathbf{w}_{m'}^T\bar{\mathbf{z}}_{jk}} \quad (11)$$

where $\mathbf{w}$ denotes the weight projection matrix, and $\mathbf{w}_m$ denotes its m-th column. We do not use softmax weight normalisation in order to simplify the model training; empirical findings (Section IV) indicates that linear normalisation works well. Tied PLDA also differs from the SGMM by using another subspace projection (matrix $\mathbf{U}_m$) to model feature correlations. It is more scalable to high dimensional feature inputs than the direct feature covariance modelling used in SGMMs.

III. Maximum Likelihood Training

A. Likelihoods

For tied PLDA, the likelihood may be computed according to equation (9) by make use of the MAP estimates of the latent variables $\mathbf{x}_{jkmt}$ and $\mathbf{z}_{jk}$, referred to as the point estimate in [12]. However, this approach does not work well in practice because of the large uncertainty of the estimation of $\mathbf{x}_{jkmt}$, i.e. the large variance of its posterior distribution.

Another approach is to marginalise out the observation variable $\mathbf{x}_{jkmt}$, which is referred as the *uncertainty estimate* in [12]. Using $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as a prior, which is the same prior used in model training for consistency (cf. equation(15)), this likelihood function can be obtained as

$$p(\mathbf{y}_t|j) = \sum_{mk} w_{jkm} \int p(\mathbf{y}_t|\mathbf{x}_{jkmt}, j, k, m)P(\mathbf{x}_{jkmt})d\mathbf{x}_{jkmt}$$

$$= \sum_{mk} w_{jkm}\mathcal{N}\left(\mathbf{y}_t; \mathbf{G}_m\bar{\mathbf{z}}_{jk} + \mathbf{b}_m, \mathbf{U}_m\mathbf{U}_m^T + \Lambda_m\right)$$

This method is similar to the channel integration evaluation method used for JFA based speaker recognition [20], [21]. Note that the likelihood can be efficiently computed without inverting matrices $\mathbf{U}_m\mathbf{U}_m^T + \Lambda_m$ directly, but by using the Woodbury matrix inversion lemma as in [20], [22]:

$$(\mathbf{U}_m\mathbf{U}_m^T + \Lambda_m)^{-1}$$

$$= \Lambda_m^{-1} - \Lambda_m^{-1}\mathbf{U}_m(\mathbf{I} + \mathbf{U}_m^T\Lambda_m^{-1}\mathbf{U}_m)^{-1}\mathbf{U}_m^T\Lambda_m^{-1} \quad (12)$$

$$= \Lambda_m^{-1} - \mathbf{L}\mathbf{L}^T \quad (13)$$

where $\mathbf{L} = \Lambda_m^{-1}\mathbf{U}_m(\mathbf{I} + \mathbf{U}_m^T\Lambda_m^{-1}\mathbf{U}_m)^{-1/2}$. This makes it computationally feasible when $\mathbf{y}_t$ is high dimensional.

It is also possible to marginalise out the state variable $\mathbf{z}_{jk}$ alone or jointly with $\mathbf{x}_{jkmt}$ similar to the methods used in [21]. However, we did not obtain a consistent improvement using this approach in our preliminary experiments. This may be the case because the variance of the posterior distribution of $\mathbf{z}_{jk}$ is small owing to increased training data used for the posterior estimation. This model-based uncertainty approach is similar to Bayesian predictive classification (BPC) for GMM-based acoustic models [23], in contrast to feature space uncertainty approaches used for noise robust speech recognition [24]–[26].

B. Model update

We used the Variational Bayesian inference to train the model where $\mathbf{x}_{jkmt}$ and $\mathbf{z}_{jk}$ are assumed to be conditionally independent. A joint model training algorithm could be obtained without making use of this assumption: however, it may be computationally infeasible in practice [19]. Similar to

the PLDA mixture model [12], the EM auxiliary function to update $\mathbf{U}_m$ in tied PLDA is

$$\begin{aligned}\mathcal{Q}(\mathbf{U}_m) &= \sum_{jkt} \int P(j,k,m|\mathbf{y}_t) P(\mathbf{x}_{jkmt}|\mathbf{y}_t, \bar{\mathbf{z}}_{jk}, j,k,m) \\ &\quad \times \log p(\mathbf{y}_t|\mathbf{x}_{jkmt}, \bar{\mathbf{z}}_{jk}, j,k,m) d\mathbf{x}_t \\ &= \sum_{jkt} \gamma_{jkmt} \mathbb{E}\Bigg[-\frac{1}{2}\mathbf{x}_{jkmt}^T \mathbf{U}_m^T \Lambda_m^{-1} \mathbf{U}_m \mathbf{x}_{jkmt} \\ &\quad + \mathbf{x}_{jkmt}^T \mathbf{U}_m^T \Lambda_m^{-1} \left(\mathbf{y}_t - \mathbf{G}_m \bar{\mathbf{z}}_{jk} - \mathbf{b}_m\right) \Bigg] + \text{const} \\ &= \sum_{jkt} \gamma_{jkmt} \mathrm{Tr}\Bigg(\Lambda_m^{-1} \Big(-\frac{1}{2}\mathbf{U}_m \mathbb{E}[\mathbf{x}_{jkmt}\mathbf{x}_{jkmt}^T] \mathbf{U}_m^T \\ &\quad + (\mathbf{y}_t - \mathbf{G}_m \bar{\mathbf{z}}_{jm} - \mathbf{b}_m) \mathbb{E}^T[\mathbf{x}_{jkmt}] \mathbf{U}_m^T \Big) \Bigg) + \text{const}\end{aligned}$$

where γ_{jkmt} denotes the component posterior probability as

$$\begin{aligned}\gamma_{jkmt} &= P(j,k,m|\mathbf{y}_t) \\ &= P(j|\mathbf{y}_t) \frac{w_{jkm} p(\mathbf{y}_t|\bar{\mathbf{z}}_{jk}, j,k,m)}{\sum_{km} w_{jkm} p(\mathbf{y}_t|\bar{\mathbf{z}}_{jk}, j,k,m)} . \end{aligned} \tag{14}$$

$P(j|\mathbf{y}_t)$ is the HMM state posterior which can be obtained using the forward-backward algorithm. $\mathbb{E}[\cdot]$ is the expectation operation over the posterior distribution of $\mathbf{x}_{jkmt}$:

$$\begin{aligned}&P(\mathbf{x}_{jkmt}|\mathbf{y}_t, \bar{\mathbf{z}}_{jk}, j,k,m) \\ &\quad = \frac{p(\mathbf{y}_t|\mathbf{x}_{jkmt}, \bar{\mathbf{z}}_{jk}, j,k,m) P(\mathbf{x}_{jkmt})}{\int p(\mathbf{y}_t|\mathbf{x}_{jkmt}, \bar{\mathbf{z}}_{jk}, j,k,m) P(\mathbf{x}_{jkmt}) d\mathbf{x}_{jkmt}} .\end{aligned} \tag{15}$$

Using $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as the prior distribution for $\mathbf{x}_{jkmt}$ we can obtain

$$P(\mathbf{x}_{jkmt}|\mathbf{y}_t, \bar{\mathbf{z}}_{jk}, j,k,m) = \mathcal{N}(\mathbf{x}_{jkmt}; \mathbf{V}_m^{-1}\mathbf{p}_{jkmt}, \mathbf{V}_m^{-1}) \tag{16}$$

$$\mathbf{V}_m = \mathbf{I} + \mathbf{U}_m^T \Lambda_m^{-1} \mathbf{U}_m \tag{17}$$

$$\mathbf{p}_{jkmt} = \mathbf{U}_m^T \Lambda_m^{-1} (\mathbf{y}_t - \mathbf{G}_m \bar{\mathbf{z}}_{jk} - \mathbf{b}_m) \tag{18}$$

Note that using $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as a prior is reasonable since, after convergence, a nonzero mean can be accounted for by $\mathbf{b}_m$, and the variance can be modified by rotating and scaling the matrix $\mathbf{U}_m$. A similar form of posterior distribution can be obtained for $\mathbf{z}_{jk}$.

By setting $\partial \mathcal{Q}(\mathbf{U}_m)/\partial \mathbf{U}_m = 0$ we obtain

$$\begin{aligned}\mathbf{U}_m &= \left(\sum_{jkt} \gamma_{jkmt} (\mathbf{y}_t - \mathbf{G}_m \bar{\mathbf{z}}_{jk} - \mathbf{b}_m) \mathbb{E}^T[\mathbf{x}_{jkmt}] \right) \\ &\quad \times \left(\sum_{jkt} \gamma_{jkmt} \mathbb{E}\left[\mathbf{x}_{jkmt}\mathbf{x}_{jkmt}^T\right] \right)^{-1}\end{aligned} \tag{19}$$

Similarly, the update for other parameters are as follows.

$$\begin{aligned}\mathbf{G}_m &= \left(\sum_{jkt} \gamma_{jkmt} (\mathbf{y}_t - \mathbf{U}_m \bar{\mathbf{x}}_{jkmt} - \mathbf{b}_m) \mathbb{E}^T[\mathbf{z}_{jk}] \right) \\ &\quad \times \left(\sum_{jkt} \gamma_{jkmt} \mathbb{E}\left[\mathbf{z}_{jk}\mathbf{z}_{jk}^T\right] \right)^{-1}\end{aligned} \tag{20}$$

$$\mathbf{b}_m = \frac{\sum_{jkt} \gamma_{jkmt} (\mathbf{y}_t - \mathbf{U}_m \bar{\mathbf{x}}_{jkmt} - \mathbf{G}_m \bar{\mathbf{z}}_{jk})}{\sum_{jkt} \gamma_{jkmt}} \tag{21}$$

$$\Lambda_m = \mathrm{diag}\left(\frac{\sum_{jkt} \gamma_{jkmt} \left(\mathbf{y}_{jkmt}\mathbf{y}_{jkmt}^T + \mathbf{U}_m \mathbf{V}_m^{-1} \mathbf{U}_m^T \right)}{\sum_{jkt} \gamma_{jkmt}} \right) \tag{22}$$

where we have defined

$$\mathbf{y}_{jkmt} = \mathbf{y}_t - \mathbf{U}_m \bar{\mathbf{x}}_{jkmt} - \mathbf{G}_m \bar{\mathbf{z}}_{jk} - \mathbf{b}_m \tag{23}$$

The sub-state and component weights can be updated as

$$c_{jk} = \frac{\sum_{mt} \gamma_{jkmt}}{\sum_{kmt} \gamma_{jkmt}}, \quad \pi_{jm} = \frac{\sum_{kt} \gamma_{jkmt}}{\sum_{kmt} \gamma_{jkmt}} \tag{24}$$

When using a large number of components, e.g. $M = 400$ in this work, the weight should be floored by a small value for numerical stability. For computational efficiency, a background model based on a mixtures of factor analysers is used to select a small subset of the components for each frame for training and decoding, which is described in more detail in [12].

IV. Experiments

We performed experiments using the Switchboard corpus[1] [27]. The Hub-5 Eval 2000 data [28] is used as the test set, which contains the Switchboard (SWB) and CallHome (CHM) evaluation subsets. The experiments were performed using the Kaldi speech recognition toolkit[2] [29], which we extended with an implementation of the PLDA-based acoustic model. In the following experiments, we have used maximum likelihood estimation without speaker adaptation or adaptive training. We used the pronunciation lexicon that was supplied by the Mississippi State transcriptions [30] and a trigram language model was used for decoding.

A. MFCC features

The first set of experiments used mel frequency cepstral coefficients (MFCCs) as features. We used the standard 39-dimensional MFCCs with first and second derivatives (MFCC_0_Δ_$\Delta\Delta$). To take advantage of longer context information, for the GMM and SGMM systems we have also performed experiments using spliced MFCC_0 of differing context window size, followed by a global LDA transformation to reduce the feature dimensionality to be 40, and a global semi-tied covariance (STC) matrix transform [31] to de-correlate the features. The PLDA systems directly used the concatenated MFCCs with various size of context window, without de-correlation and dimensionality reduction.

Table I shows the results of using a 33 hour subset of the training data, and the number of active model parameters[3]. In this case, there are about 2,400 clustered triphone states in the GMM systems, corresponding to about 30,000 Gaussians. The PLDA and SGMM systems have a similar number of clustered

[1] https://catalog.ldc.upenn.edu

[2] http://kaldi.sourceforge.net

[3] For PLDA systems, a component is considered active if its weight is above a threshold (0.01 in this work).

TABLE I
WER (%) USING 33 HOURS SWITCHBOARD TRAINING DATA, WITH DIFFERENT FEATURE DIMENSIONS AND DIFFERENT NUMBER OF ACTIVE MODEL PARAMETERS

System	Feature	Feature dim	#State-dependent parameters	#State-indepdent parameters	CHM	SWB	Avg
GMM	MFCC_0+Δ+$\Delta\Delta$	39	2.40×10^6	-	54.0	36.6	45.4
GMM	MFCC_0($\pm$2)+LDA_STC	40	2.43×10^6	-	54.4	34.4	43.7
GMM	MFCC_0($\pm$3)+LDA_STC	40	2.43×10^6	-	50.6	33.5	42.2
GMM	MFCC_0($\pm$4)+LDA_STC	40	2.43×10^6	-	50.7	33.3	42.1
GMM	MFCC_0($\pm$5)+LDA_STC	40	2.43×10^6	-	50.9	34.1	42.4
SGMM	MFCC_0+Δ+$\Delta\Delta$	39	0.8×10^6	0.97×10^6	48.5	31.4	40.1
SGMM	MFCC_0($\pm$2)+LDA_STC	40	0.8×10^6	0.99×10^6	45.7	30.0	38.0
SGMM	MFCC_0($\pm$3)+LDA_STC	40	0.8×10^6	0.99×10^6	45.1	29.7	37.5
SGMM	MFCC_0($\pm$4)+LDA_STC	40	0.8×10^6	0.99×10^6	45.1	29.3	37.4
SGMM	MFCC_0($\pm$5)+LDA_STC	40	0.8×10^6	0.99×10^6	45.7	29.5	37.7
mix-PLDA	MFCC_0 ($\pm$2)	65	2.34×10^6	2.11×10^6	51.4	33.1	42.3
mix-PLDA	MFCC_0 ($\pm$3)	91	2.22×10^6	2.94×10^6	49.5	32.4	41.1
mix-PLDA	MFCC_0 ($\pm$4)	117	2.16×10^6	3.78×10^6	49.3	31.5	40.6
mix-PLDA	MFCC_0 ($\pm$5)	143	2.12×10^6	4.61×10^6	49.7	33.2	41.6
tied-PLDA	MFCC_0 ($\pm$2)	65	0.86×10^6	2.11×10^6	48.6	31.9	40.4
tied-PLDA	MFCC_0 ($\pm$3)	91	0.86×10^6	2.94×10^6	47.9	31.0	39.5
tied-PLDA	MFCC_0 ($\pm$4)	117	0.86×10^6	3.78×10^6	47.5	31.2	39.4
tied-PLDA	MFCC_0 ($\pm$5)	143	0.86×10^6	4.61×10^6	48.7	32.2	40.6
tied-PLDA	MFCC_0($\pm$3)+LDA_STC	40	0.85×10^6	1.61×10^6	45.7	29.5	37.7

TABLE II
WER (%) USING 33 AND 109 HOURS SWITCHBOARD TRAINING DATA

System	Feature	33 hours		109 hours	
		CHM	SWB	CHM	SWB
DNN hybrid	MFCC_0+Δ+$\Delta\Delta$ ($\pm$4)	43.1	27.6	36.3	22.0
BN hybrid	MFCC_0+Δ+$\Delta\Delta$ ($\pm$4)	44.0	28.8	37.7	22.7
GMM	MFCC_0+Δ+$\Delta\Delta$	54.0	36.6	48.9	31.0
GMM	MFCC_0($\pm$3)+LDA_STC	50.6	33.5	44.9	28.0
GMM	BN_MFCC	44.8	30.9	39.7	25.5
GMM	BN_MFCC + LDA_STC	43.2	27.4	36.7	22.1
SGMM	BN_MFCC + LDA_STC	41.7	26.7	36.2	21.7
mix-PLDA	BN_MFCC	42.6	27.1	35.9	21.6
tied-PLDA	BN_MFCC	41.7	26.8	35.1	21.4

triphone states, and a 400-component background model is used for each. The state vector of SGMMs and latent variables of PLDA are all 40-dimensional. We used 20,000 sub-state vectors and state variables in the SGMM and tied PLDA systems, respectively. These results demonstrate the flexibility of PLDA systems in using different dimensional acoustic features, i.e. the spliced MFCC_0 without any frontend feature transformations. Tied PLDA systems also offer consistently lower WERs than their counterparts based on the PLDA mixture model. Using the same low dimensional features as MFCC_0($\pm$3)+LDA_STC, the tied PLDA system achieved comparable recognition accuracy to SGMMs. This system is better than tied PLDA systems using spliced MFCC_0 of various context windows, which means that removing the non-discriminative dimensions in feature space is still beneficial to tied PLDAs.

B. Bottleneck features

Table II shows the WERs of DNN and bottleneck systems using 33 hours and 109 hours of training data, respectively. The DNN system has six hidden layers, each with 1024 hidden units when using 33 hours of training data. The number of hidden units is increased to be 1200 when the amount of training data is 109 hours.The bottleneck DNN system (BN hybrid) used the same training data and the same kind of feature input — while reducing the size of the fifth hidden layer to be 26. Using a larger bottleneck layer was not found to be helpful [19]. We concatenated the bottleneck and MFCC_0+Δ+$\Delta\Delta$ coefficients (referred as BN_MFCC), and then used them to retrain our GMM and PLDA systems. We used LDA to reduce the dimensionality of the concatenated features from 65 to be 40 followed by STC to de-correlate the features for GMM and SGMM systems. Without the front-end feature transforms, the PLDA systems were able to achieve comparable or higher recognition accuracy by directly capturing the correlations between MFCCs and bottleneck features in subspaces. Again, the results demonstrate the flexibility of PLDA acoustic models in terms of using input feature vectors of varying dimension.

V. CONCLUSIONS

Building upon our previous work on acoustic modelling using the PLDA mixture model, we have presented a tied PLDA based acoustic model, which is more scalable to the amount of training data. Experiments show that this model can achieve higher recognition accuracy while still enjoying the flexibility of using acoustic features of various dimension as the PLDA mixture model. Other types of acoustic feature representations can be more freely explored using this acoustic model. Along this line, we have demonstrated that the bottleneck feature from a DNN can used without any front-end feature transformation for dimensionality reduction and de-correlation. Future works include speaker adaptation and discriminative training for this model, and moreover, we are also interested in learning speech representations in an unsupervised fashion using a deep auto-encoder for this model. The source code and recipe used in this work are available from http://homepages.inf.ed.ac.uk/llu/code/plda-v1.tgz.

REFERENCES

[1] Steven Davis and Paul Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[2] Hynek Hermansky, "Perceptual linear predictive (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[3] Steve Young, "A review of large-vocabulary continuous-speech recognition," *Signal Processing Magazine, IEEE*, vol. 13, no. 5, pp. 45, 1996.

[4] J-L Gauvain and Lori Lamel, "Large-vocabulary continuous speech recognition: advances and applications," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1181–1200, 2000.

[5] M. Gales and S. Young, "The application of hidden markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[6] GE Dahl, D Yu, L Deng, and A Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[7] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Brain Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[8] Herve A Bourlard and Nelson Morgan, *Connectionist speech recognition: a hybrid approach*, vol. 247, Springer, 1994.

[9] Steve Renals, Nelson Morgan, Hervé Bourlard, Michael Cohen, and Horacio Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 161–174, 1994.

[10] František Grézl, Martin Karafiát, Stanislav Kontár, and J Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. ICASSP*. IEEE, 2007, vol. 4.

[11] Dong Yu and Michael L Seltzer, "Improved bottleneck features using pretrained deep neural networks.," in *INTERSPEECH*, 2011, pp. 237–240.

[12] L Lu and S Renals, "Probabilistic linear discriminant analysis for acoustic modelling," *IEEE Signal Processing Letters*, 2014.

[13] Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. ICCV*. IEEE, 2007, pp. 1–8.

[14] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[15] Patrick Kenny, "Bayesian speaker verification with heavy tailed priors," in *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2010.

[16] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[17] Pavel Matejka, Ondrej Glembek, Fabio Castaldo, Md Jahangir Alam, Oldrich Plchot, Patrick Kenny, Lukas Burget, and Jan Cernocky, "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification," in *Proc. ICASSP*. IEEE, 2011, pp. 4828–4831.

[18] D Povey, L Burget, M Agarwal, P Akyazi, F Kai, A Ghoshal, O Glembek, N Goel, M Karafiát, A Rastrow, RC Rose, P Schwarz, and S Thomas, "The subspace Gaussian mixture model—A structured model for speech recognition," *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011.

[19] L Lu and S Renals, "Probabilistic linear discriminant analysis with bottleneck features for speech recognition," in *Proc. INTERSPEECH*, 2014.

[20] Ondrej Glembek, Lukas Burget, Najim Dehak, Niko Brummer, and Patrick Kenny, "Comparison of scoring methods used in speaker recognition with joint factor analysis," in *Proc. ICASSP*. IEEE, 2009, pp. 4057–4060.

[21] Xianyu Zhao and Yuan Dong, "Variational bayesian joint factor analysis models for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing,*, vol. 20, no. 3, pp. 1032–1042, 2012.

[22] Peng Li, Yun Fu, Umar Mohammed, James H Elder, and Simon JD Prince, "Probabilistic models for inference about identity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 144–157, 2012.

[23] Qiang Huo and Chin-Hui Lee, "A bayesian predictive classification approach to robust speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 2, pp. 200–204, 2000.

[24] J Droppo, A Acero, and L Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. ICASSP*. IEEE, 2002.

[25] H Liao and MJF Gales, "Joint uncertainty decoding for noise robust speech recognition," in *Proc. INTERSPEECH*. Citeseer, 2005.

[26] L Lu, KK Chin, A Ghoshal, and S Renals, "Joint uncertainty decoding for noise robust subspace Gaussian mixture models," *IEEE Transactions on Audio, Speech, and Language Processing*, 2013.

[27] John J Godfrey, Edward C Holliman, and Jane McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*. IEEE, 1992, pp. 517–520.

[28] Christopher Cieri, David Miller, and Kevin Walker, "Research methodologies, observations and outcomes in (conversational) speech data collection," in *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 2002, pp. 206–211.

[29] D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlıcek, Y Qian, P Schwarz, J Silovský, G Semmer, and K Veselý, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[30] Neeraj Deshmukh, Aravind Ganapathiraju, Andi Gleeson, Jonathan Hamaker, and Joseph Picone, "Resegmentation of SWITCHBOARD," in *Proc. ICSLP*, 1998.

[31] MJF Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.