Probabilistic Linear Discriminant Analysis with Bottleneck Features for Speech Recognition

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
INTERSPEECH-2014

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Probabilistic Linear Discriminant Analysis with Bottleneck Features for Speech Recognition

Liang Lu, Steve Renals
Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK
{liang.lu, s.renals}@ed.ac.uk

Abstract
We have recently proposed a new acoustic model based on probabilistic linear discriminant analysis (PLDA) which enjoys the flexibility of using higher dimensional acoustic features, and is more capable to capture the intra-frame feature correlations. In this paper, we investigate the use of bottleneck features obtained from a deep neural network (DNN) for the PLDA-based acoustic model. Experiments were performed on the Switchboard dataset — a large vocabulary conversational telephone speech corpus. We observe significant word error reduction by using the bottleneck features. In addition, we have also compared the PLDA-based acoustic model to three others using Gaussian mixture models (GMMs), subspace GMMs and hybrid deep neural networks (DNNs), and PLDA can achieve comparable or slightly higher recognition accuracy from our experiments.

Index Terms: speech recognition, bottleneck features, probabilistic linear discriminant analysis

1. Introduction
Deep neural network (DNN) approaches have recently produced significant increases in the accuracy of acoustic modelling for speech recognition, across a range of application domains and evaluation datasets [1, 2]. Compared to the hybrid neural network / hidden Markov model (HMM) architecture studied in the early 1990s [3, 4], DNNs typically use more hidden layers and a wider output layer. The deep architecture enables a DNN to learn more invariant and discriminative features before performing classification using the final softmax output layer. However, there has been only limited success in the adaptation of DNN-based acoustic models [5, 6], and in general they have to “learn by seeing” [7] — high recognition accuracy is usually obtained in matched training and test conditions. Thus hybrid DNN/HMM approaches may perform poorly in unseen acoustic conditions, especially if there is limited in-domain training data.

Tandem systems [8] use a neural network to provide features for a conventional Gaussian mixture model (GMM) based system, and a particular example of the neural network derived features is known as the bottleneck features [9, 10]. This method can take advantages of DNN feature extraction while enjoying the efficient adaptation algorithms for GMMs. However, GMMs typically employ diagonal covariance matrices, which limits their ability to learn feature correlations, as well as effectively restricting the bottleneck features to a limited dimensionality for computational reasons. As a result, a pre-processing approach such as principal component analysis (PCA) is often used to decorrelate and reduce the dimensionality of the bottleneck features.

We recently introduced a new acoustic model based on probabilistic linear discriminant analysis (PLDA) [11], which aims to overcome these constraints. It can be viewed as an extension of the GMM which is able to use higher dimensional feature vectors and can learn feature correlations in subspaces. PLDA was originally proposed for face recognition [12], and is now very well studied for speaker recognition using the i-vector framework [13, 14, 15]. PLDA is a probabilistic extension of linear discriminant analysis (LDA) [12]; similar to joint factor analysis (JFA) [16], PLDA factorizes the variability of the observations for a specific class (e.g. one speaker) using two latent variables: a within-class variable which is shared by all the observations of this class, and a between-class variable which is used to explain the variability to each observation. Furthermore when applied to speaker identification JFA operates in the GMM mean supervector domain, while the PLDA-based acoustic model directly operates in the acoustic feature domain.

We have previously demonstrated the feasibility of the PLDA-based acoustic model, its flexibility in using feature vectors of various dimensions, and its ability to learn feature correlations [11]. In this paper, we investigate the use of bottleneck features with PLDA-based acoustic models, in order to take the advantage of DNNs as feature extractors. In experiments on the Switchboard corpus [17], we compare this model to three other acoustic modelling approaches: GMMs, subspace GMMs (SGMMs) [18] and hybrid DNN/HMMs [1], and show that comparable or better recognition accuracy can be obtained.

2. PLDA-based Acoustic Model
The PLDA-based acoustic model is a generative model in which an acoustic feature vector \( y_t \in \mathbb{R}^p \) from the \( j \)-th HMM state at time index \( t \) is expressed as:

\[
y_{t|j} = U x_{jt} + G z_{jt} + b + \epsilon_{jt}, \quad \epsilon_{jt} \sim \mathcal{N}(0, \Lambda),
\]

where \( z_{jt} \in \mathbb{R}^q \) 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. \( x_{jt} \in \mathbb{R}^p \) is the observation 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 \( y_t \), i.e. \( p \leq d \), \( q \leq d \). \( U \in \mathbb{R}^{d \times p} \) and \( G \in \mathbb{R}^{d \times q} \) are two low rank matrices which span the subspaces to capture the major variations for \( x_{jt} \) and \( z_{jt} \) respectively. They are analogous to the within-class and between-class subspaces in the standard LDA formulation, but are estimated probabilistically. \( b \in \mathbb{R}^d \) denotes the bias and \( \epsilon_{jt} \in \mathbb{R}^d \) is the residual noise which is assumed to be Gaussian with zero mean and di-
where we have used $\mathcal{N}(\mathbf{0}, \Lambda_m)$ as the prior distribution for $P(\mathbf{x}_{jmt})$, following the practice used for speaker and face recognition using JFA and PLDA [19, 20]. Note that the likelihood can be efficiently computed without inverting matrices $U_m U_m^T + \Lambda_m$ directly, but by using the following Woodbury matrix inversion lemma as in [19, 20]:

$$(U_m U_m^T + \Lambda_m)^{-1} = \Lambda_m^{-1} - \Lambda_m^{-1} U_m (I + U_m^T \Lambda_m^{-1} U_m)^{-1} U_m^T \Lambda_m^{-1} \Lambda_m^{-1} - LL^T \Lambda_m^{-1}$$

where $L = \Lambda_m^{-1/2} U_m (I + U_m^T \Lambda_m^{-1} U_m)^{-1/2}$. This makes it computationally feasible when $y_t$ is high dimensional. As discussed in [11], this acoustic model is closely related to factor analysed HMMs [21] and SGMMs [18].

### 3. Model Training

A PLDA-based acoustic model may be trained using an EM algorithm [11], which is based on the assumption that the two latent variables $x_{jmt}$ and $z_{jmt}$ are conditionally independent. This allows the updates of the projection matrices $U_m$ and $G_m$ to be interleaved. Conditional independence of the latent variables was not assumed when using PLDA for face recognition [12, 20], but a joint model training approach was used. For PLDA-based acoustic modelling, such kind of training algorithm can be derived by representing the model from stacking the $T$ frames of $j$-th HMM state and $m$-th PLDA component:

$$\begin{pmatrix}
\mathbf{y}_1 \\
\mathbf{y}_2 \\
\vdots \\
\mathbf{y}_T
\end{pmatrix} = \begin{pmatrix}
\mathbf{G}_m & \mathbf{U}_m & \cdots & \mathbf{0} \\
\mathbf{G}_m & \mathbf{0} & \cdots & \mathbf{U}_m \\
\mathbf{b}_m \\
\vdots \\
\mathbf{b}_m
\end{pmatrix} \begin{pmatrix}
x_{jmt} \\
z_{jmt}
\end{pmatrix} + \begin{pmatrix}
\epsilon_{jmt1} \\
\epsilon_{jmtT}
\end{pmatrix}
$$

or in the form of the new notation

$$\mathbf{y}_j, m = \mathbf{H}_m \mathbf{v}_{jmt} + \mathbf{b}_m + \epsilon_{jmt}. \quad (8)$$

This is a factor analysis model, and the model training algorithms for mixtures of factor analysers may be used [22, 16]. Unfortunately, this approach is computationally demanding for acoustic modelling, since the number of frames is normally very large, which results in significant computational and memory demands.

This difficulty can be circumvented if $x_{jmt}$ and $z_{jmt}$ are assumed to be conditionally independent. In this case, the EM auxiliary function to update $U_m$ is

$$Q(U_m) = \sum_{j,t} \int P(j, m | y_t) P(x_{jmt} | y_t, z_{jmt}, j, m)$$

$$\times \log p(y_t | x_{jmt}, z_{jmt}, j, m) dx_t$$

$$= \sum_{j,t} \gamma_{jmt} E \left[ -\frac{1}{2} x_{jmt}^T U_m^{-1} U_m x_{jmt} + x_{jmt}^T U_m^{-1} (y_t - G_m z_{jmt} - b_m) \right] + k$$

$$= \int \sum_{j,t} \gamma_{jmt} \left( \Lambda_m^{-1} - \frac{1}{2} U_m E[x_{jmt} x_{jmt}^T] U_m^T \right)$$

$$+ (y_t - G_m z_{jmt} - b_m) E^T [x_{jmt} U_m^T] + k$$

where $k$ is a constant that is independent of $U_m$, $\gamma_{jmt}$ denotes the component posterior probability $P(j, m | y_t)$, and $z_{jmt}$ denotes the mean of the posterior distribution of $z_{jmt}$ which is computed from the previous iteration. $E[.]$ is the expectation operation over the posterior distribution of $x_{jmt}$, which can be computed from the Bayes’ rule:

$$P(x_{jmt} | y_t, z_{jmt}, j, m)$$

$$= \int p(y_t | x_{jmt}, z_{jmt}, j, m) P(x_{jmt}) dx_{jmt}. \quad (9)$$
Using $N(0, I)$ as the prior for $x_{jm}$, and with some algebraic rearrangement, we can obtain

$$P(x_{jm} | y_t, z_{jm}) = N(x_{jm} | V_m^{-1} W_{jm}, V_m^{-1})$$

We then set $\partial Q(U_m) / \partial U_m = 0$ to obtain the update for $U_m$

$$U_m = \left( \sum_j \gamma_{jm}(y_t - G_m z_{jm} - b_m) E^T [x_{jm}] \right)$$

$$\times \left( \sum_j \gamma_{jm} E [x_{jm} x_{jm}^T] \right)^{-1}$$

The updates for $\{G_m, b_m, \Lambda_m\}$ can be derived similarly.

4. Experiments

We have performed experiments using the Switchboard corpus [17], in which the training set comprises about 300 hours of conversational telephone speech. The Hub-5 Eval 2000 data [23] is used as the test set. The experiments were performed using the Kaldi speech recognition toolkit [24], which we have extended with an implementation of the PLDA-based acoustic model. Our current implementation does not yet support speaker adaptation and discriminative training, and hence in the following experiments, we have used maximum likelihood estimation and have not employed speaker adaptation (aside from speaker-based cepstral mean and variance normalisation). We used the pronunciation lexicon that was supplied by the Mississippi State transcriptions [25] which has more than 30,000 words, and a trigram language model was used for decoding.

4.1. Baseline systems

Table 1 shows the word error rates (WERs) of four types of speaker-independent acoustic models without sequence discriminative training: GMM, SGMM, PLDA, and hybrid DNN. Each model was trained using about 33 hours of Switchboard training data, and we show separate results for the Callhome (CHM) and Switchboard (SWB) evaluation sets. The number of tied HMM states is around 2,400 for each of the acoustic models shown in this table. The GMM system has about 30,000 Gaussian components. The number of components in the background model is 400 for both PLDA and SGMM systems. There were 20,000 sub-states (with 40-dimensional sub-state vectors) in the SGMM system. In the PLDA system, $x_{jm}$ and $y_{jm}$ are also 40-dimensional. In the DNN system, the feature input was obtained by splicing 12-dimensional MFCCs with zeroth, delta, and delta-delta coefficients (MFCC, $\Delta$, $\Delta\Delta$) using a context size of 9 frames (i.e. $\pm 4$). Six hidden layers each with 1,024 nodes were used. The size of DNN output was the number of tied HMM states (around 2400), giving a total of about 8 million parameters. We have also studied different forms of feature input for GMM and PLDA acoustic models. From Table 1, we see that the best PLDA system has a consistently lower WER than the GMM systems with and without linear discriminant analysis (LDA) and semi-tied covariance matrix (STC) [26] when using spliced MFCCs as feature input. It is also comparable with SGMMs, but is more flexible with respect to the feature vector dimensionality. The DNN system has a significantly lower WER compared to the other acoustic models.

4.2. Bottleneck features

We trained bottleneck DNNs — using the same training data and the same kind of feature input — by reducing the size of the fifth hidden layer. We evaluated different sizes of the bottleneck layer, ranging from 26 to 80. The WERs of BN-DNNs were only slightly higher than the standard DNN system (0.5%
4.3. Increased training data

To investigate whether the conclusion from the previous experiments holds in case of increased training data, we performed experiments using around 109 hours of Switchboard training data. In this case, we still used 6 hidden layers for the hybrid DNN system, but increase the size of each hidden layer to be 1200. The number of output nodes is around 4000, giving a total of approximately 12.5 million parameters. Again, the bottleneck DNN system has the same configuration as the hybrid DNN except that the size of the bottleneck layer is set to be 26. The GMM systems have around 90,000 Gaussian components, and the SGMM system has about 60,000 sub-state vectors. Again, $M = 400$ for the PLDA system, which is the same size as the UBM in the SGMM acoustic model. A summary of the results is given in Table 3 where we can see a similar trend as in Table 2. The “GMM+Tandem+LDA+STC” system is able to achieve almost the same WER as the hybrid DNN system, while the PLDA and SGMM systems can perform slightly better.

We then investigated the generalisation ability of the bottleneck features. We extracted the bottleneck features for the 109 hours of training data using the bottleneck DNN trained from 33 hours of data discussed in section 4.2. We then reproduced the Tandem systems using the GMM, SGMM and PLDA acoustic modelling approaches. The WERs are given in Table 4. By looking at the GMM systems, we observe that using such kind of bottleneck features can still lead to much higher recognition accuracy compared to using the MFCCs alone, yet the overall WERs are considerably higher than those reported in Table 3. This may indicate that using more and matched training data to train the bottleneck DNN to extract the features is beneficial for the Tandem system. Note that the comparison was performed using the Switchboard database which has limited differences in the acoustic conditions. We may expect a further drop in terms of recognition accuracy when there is a mismatch between the training and test conditions. However, as we discussed in section 1, there are efficient adaptation algorithms for acoustic models within the Gaussian family which can mitigate this problem. It is necessary to point out that the hybrid DNN system can be significantly improved by using feature space MLLR (fMLLR) transformation [27] and sequence training criterion [28, 29, 30]. It is worthwhile to look at if the bottleneck features extracted from such kind DNN systems can further improve the Tandem system. In addition, the PLDA-based acoustic model may be further improved by tying the state variables across the components [11], which is one of our future works.

5. Conclusions

In this paper, we reviewed the recently proposed PLDA-based acoustic modelling approach, and investigated the use of bottleneck features from a DNN for this model. We demonstrated the flexibility of this model in making use of such kind of feature representation, and have obtained comparable or higher recognition accuracy compared to other acoustic model including GMMs, SGMMs, and hybrid DNNs. The current implementation of PLDA for acoustic modelling may be improved by sharing model parameters, e.g. through tying the state variables across the model components which would be analogous to state vectors used in SGMMs. Future work also include speaker adaptation and discriminative training for this model.

<table>
<thead>
<tr>
<th>System</th>
<th>Feature</th>
<th>CHM</th>
<th>SWB</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN hybrid</td>
<td>MFCC $\Delta+\Delta$</td>
<td>36.7</td>
<td>22.1</td>
<td>30.4</td>
</tr>
<tr>
<td>GMM</td>
<td>MFCC $\Delta+\Delta$</td>
<td>44.9</td>
<td>28.0</td>
<td>36.5</td>
</tr>
<tr>
<td>GMM</td>
<td>Tandem</td>
<td>39.7</td>
<td>25.5</td>
<td>32.6</td>
</tr>
<tr>
<td>GMM</td>
<td>Tandem + LDA+STC</td>
<td>36.7</td>
<td>22.1</td>
<td>29.5</td>
</tr>
<tr>
<td>SGMM</td>
<td>Tandem + LDA+STC</td>
<td>36.2</td>
<td>21.7</td>
<td>29.0</td>
</tr>
<tr>
<td>PLDA</td>
<td>Tandem</td>
<td>35.9</td>
<td>21.6</td>
<td>28.8</td>
</tr>
</tbody>
</table>

Table 3: WERs (%) using 109 hours Switchboard training data.

## Table 2: WERs (%) using 33 hours Switchboard training data

<table>
<thead>
<tr>
<th>System</th>
<th>Feature</th>
<th>CHM</th>
<th>SWB</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN hybrid</td>
<td>MFCC $\Delta+\Delta+\Delta$</td>
<td>48.9</td>
<td>31.0</td>
<td>40.1</td>
</tr>
<tr>
<td>GMM</td>
<td>MFCC $\Delta+\Delta+\Delta$</td>
<td>44.9</td>
<td>28.0</td>
<td>36.5</td>
</tr>
<tr>
<td>GMM</td>
<td>Tandem</td>
<td>39.7</td>
<td>25.5</td>
<td>32.6</td>
</tr>
<tr>
<td>GMM</td>
<td>Tandem + LDA+STC</td>
<td>36.7</td>
<td>22.1</td>
<td>29.5</td>
</tr>
<tr>
<td>SGMM</td>
<td>Tandem + LDA+STC</td>
<td>36.2</td>
<td>21.7</td>
<td>29.0</td>
</tr>
<tr>
<td>PLDA</td>
<td>Tandem</td>
<td>35.9</td>
<td>21.6</td>
<td>28.8</td>
</tr>
</tbody>
</table>

Table 4: WERs (%) using 109 hours Switchboard training data. Here, the bottleneck features were extracted from the bottleneck DNN system trained using 33 hours of data.

## Table 3: WERs (%) using 109 hours Switchboard training data.
6. References


