Probabilistic Linear Discriminant Analysis (PLDA) with Bottleneck Features for Speech Recognition

Liang Lu, and Steve Renals
University of Edinburgh

- Motivation
- The model
- Experiments
- Conclusion

Motivation

- Deep learning for speech feature representations
  - Deep neural networks
  - Denoising autoencoder
  - Bottleneck features
- Do they fit GMMs?
  - Low and de-correlated feature input
  - weak covariance modelling
- PLDA-based acoustic models
  - Higher dimensional feature input
  - Approximated full-covariance modelling
- Bottleneck features from a DNN (this paper)

Probabilistic linear discriminant analysis

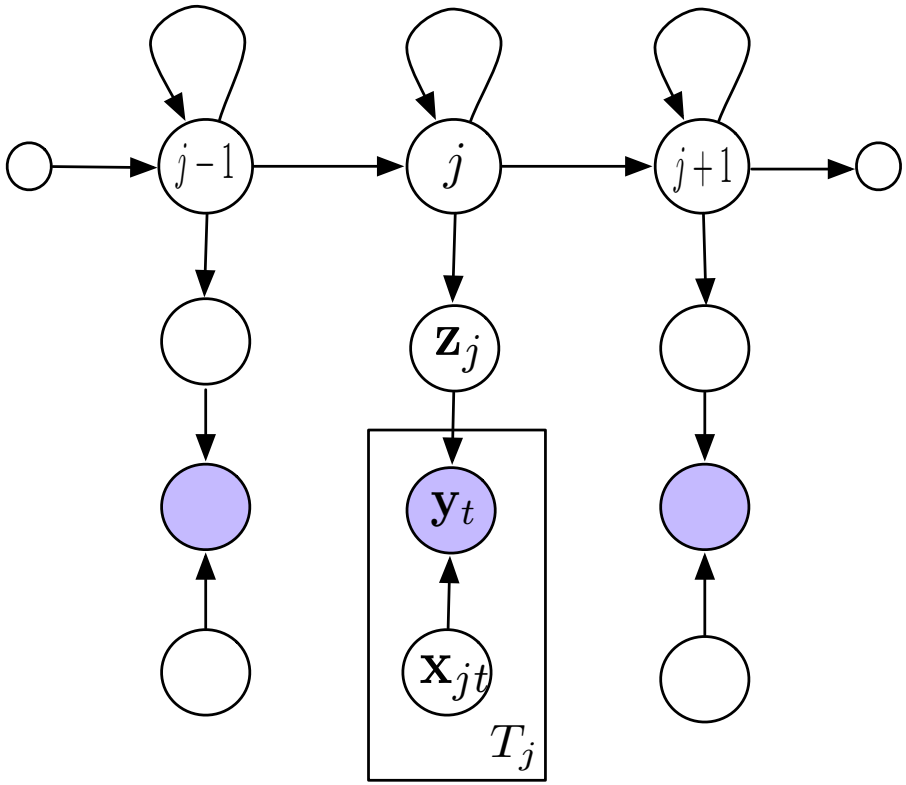
A factorisation model:

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

where

- j is the class index, e.g. HMM state index
- t is the frame index
- $\mathbf{z}_j$ is the *state variable* that depends on each state
- $\mathbf{x}_{jt}$ is the *frame variable* that depends on each frame
- $\mathbf{U}, \mathbf{G}$ are two factor loading matrices, and $\mathbf{b}$ is the bias vector

PLDA-HMM acoustic model



Gaussian PLDA

- If we assume the latent variables are Gaussian distributed, we obtain Gaussian PLDA

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

or we marginalise out $\mathbf{x}_{jt}$

$$p(\mathbf{y}_t|\mathbf{z}_j, j) = \mathcal{N}(\mathbf{y}_t; \mathbf{G}\mathbf{z}_j + \mathbf{b}, \underbrace{\mathbf{U}\mathbf{U}^T + \Lambda}_{covariance}) \quad (3)$$

- For higher dimensional features, $\mathbf{z}_j$ and $\mathbf{x}_{jt}$ can be low dimensional, fast to train
- Better covariance modelling
- Model can be trained using Variational Bayesian EM

PLDA-HMM acoustic model

Extensions:

- PLDA mixture model

$$\mathbf{y}_t|j,m = \mathbf{U}_m\mathbf{x}_{jmt} + \underline{\mathbf{G}_m\mathbf{z}_{jm}} + \mathbf{b}_m + \epsilon_{jmt} \quad (4)$$

- Tied PLDA mixture model, similar to SGMM

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

Related works:

- PLDA + iVectors for speaker verificaiton
- Joint factor analysis model
- Factorized cluster adaptive training (fCAT)

Setup

- Switchboard corpus
  - Using 33/109 hours of training data
- Bottleneck features (BN)
  - 33 hours data $\rightarrow$ 6 hidden layers $\times$1024 hidden units
  - 110 hours data $\rightarrow$ 6 hidden layers $\times$ 1200 hidden units
  - 5th hidden layer $\rightarrow$ bottleneck layer
- Tandem approach $\rightarrow$ (BN, MFCC)
- Maximum likelihood speaker independent systems

Experiments - MFCCs

Table: WERs (%) using 33 hours Switchboard training data, SI systems

System	Feature	Dim	WER
GMM	MFCC_0+Δ+ΔΔ	39	36.6
GMM	MFCC_0(±2)+LDA_STC	40	34.4
GMM	MFCC_0(±3)+LDA_STC	40	33.5
GMM	MFCC_0±4+LDA_STC	40	33.3
mix-PLDA	MFCC_0(±2)	65	33.1
mix-PLDA	MFCC_0(±3)	91	32.4
mix-PLDA	MFCC_0(±4)	117	31.5
mix-PLDA	MFCC_0(±5)	143	33.2
mix-PLDA	MFCC_0+Δ+ΔΔ(±1)	117	32.4
mix-PLDA	MFCC_0+Δ+ΔΔ(±2)	195	34.0
SGMM	MFCC_0+Δ+ΔΔ	39	31.4
DNN	MFCC_0+Δ+ΔΔ(±4)	396	27.6

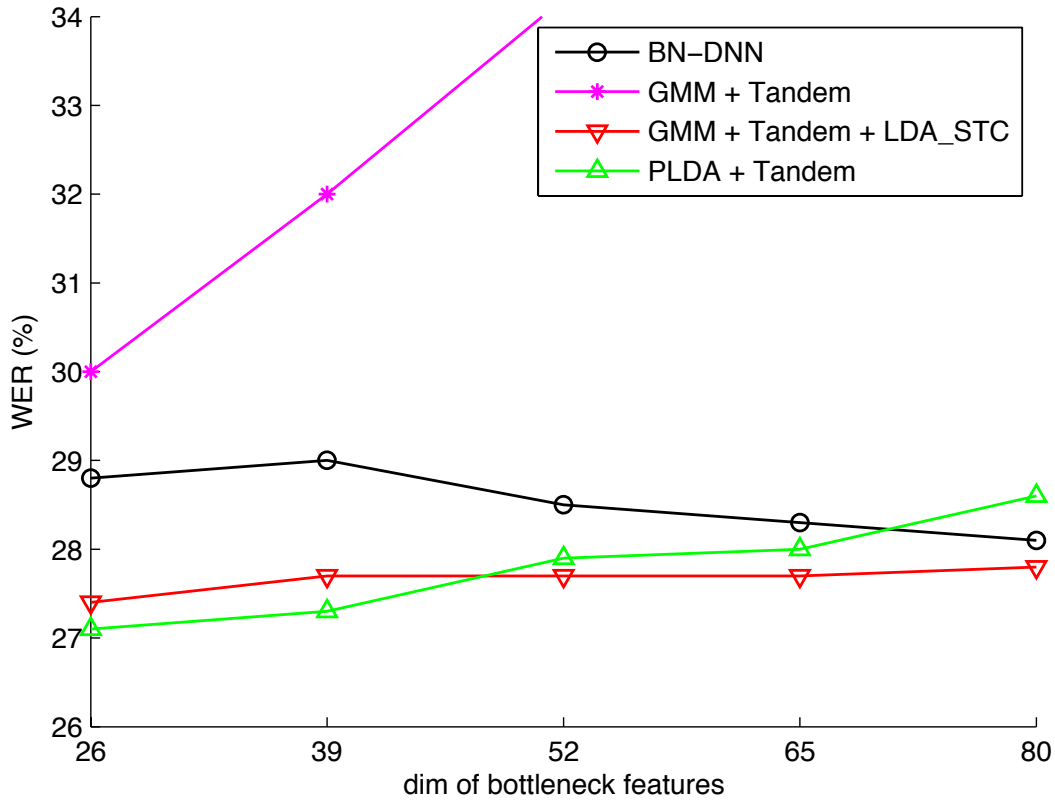
Experiments - bottleneck features

Table: WERs (%) using 33 hours Switchboard training data, 26-dim bottleneck features (65-dim Tandem), SI systems

System	Feature	WER
DNN	MFCC_0+Δ+ΔΔ (±4)	27.6
BN-DNN	MFCC_0+Δ+ΔΔ (±4)	28.8
GMM	MFCC_0+Δ+ΔΔ	36.6
GMM	MFCC_0(±3)+LDA_STC	33.5
GMM	Tandem	30.9
GMM	Tandem + LDA_STC	27.4
SGMM	Tandem + LDA_STC	26.7
mix-PLDA	Tandem	27.1
tied-PLDA	Tandem	26.8

Experiments - bottleneck features

- Results of using different size of bottleneck layer



Experiments

Table: WERs (%) using 109 hours Switchboard training data, SI systems

System	Feature	WER
DNN	MFCC_0+Δ+ΔΔ (±4)	22.0
BN-DNN	MFCC_0+Δ+ΔΔ (±4)	22.7
GMM	MFCC_0+Δ+ΔΔ	31.0
GMM	MFCC_0(±3) +LDA_STC	28.0
GMM	Tandem	25.5
GMM	Tandem + LDA_STC	22.1
SGMM	Tandem + LDA_STC	21.7
mix-PLDA	Tandem	21.6
tied-PLDA	Tandem	21.4

Summary

- PLDA for acoustic modelling, and results of using bottleneck features
- Discriminative training
- Speaker adaptation – another way for full-covariance adaptation
- FMLLR with log-Mel filter bank features for DNN adaptation

Thanks!