Deep Neural Networks and Hidden Markov Models in i-vector-based Text-Dependent Speaker Verification

Hossein Zeinali [1,2], Lukáš Burget [2], Hossein Sameti [1], Ondřej Glembek [2], Oldřich Plchot [2]

[1] Sharif University of Technology, Tehran, Iran
[2] Brno University of Technology, Czech Republic

Odyssey Speaker and Language Recognition Workshop
June 2016

Introduction

- Text-Dependent Speaker Verification (TD-SV) is the task of verifying both speaker and phrase
  - We know the phrase information
- Using phrase-independent HMM model for frame alignment
  - By HMM, we can use the phrase information.
  - We can take into account the frame order.
  - We can reduce the i-vector estimation uncertainty.
    - HMM can reduce the uncertainty about 20% relatively
- Using Deep Neural Networks (DNNs) for reducing the gap between GMM and HMM alignment
- Using Bottleneck features for improving the HMM performance

General i-vector based system

- Utterance-dependent supervector s modeled as:

$$\mathbf{s} = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{1}$$

- We need *zero and first-order statistics* $\mathbf{n}_{\mathcal{X}} = [N_{\mathcal{X}}^{(1)}, \ldots, N_{\mathcal{X}}^{(C)}]'$ and $\mathbf{f}_{\mathcal{X}} = [\mathbf{f}_{\mathcal{X}}^{(1)'}, \ldots, \mathbf{f}_{\mathcal{X}}^{(C)'}]'$ for training and i-vector extraction, where:

$$N_{\mathcal{X}}^{(c)} = \sum_t \gamma_t^{(c)} \tag{2}$$

$$\mathbf{f}_{\mathcal{X}}^{(c)} = \sum_t \gamma_t^{(c)} \mathbf{o}_t \, , \tag{3}$$

- $\gamma_t^{(c)}$ is the posterior probability of frame $\mathbf{o}_t$ being generated by the mixture component c
- $\gamma_t^{(c)}$ can be computed using UBM, DNN or HMM (our method)

Using HMM as UBM in i-vector based TD-SV

- Using phrase-dependent HMM models
  - Need phrase dependent i-vector extractor
  - Suitable for common pass-phrase and text-prompted SV
  - Need sufficient training data from each phrase
  - Not practical for TD-SV
- Tied mixture HMMs [Kenny et al.]
- Phrase-independent HMM models (our method)
  - Using a mono-phone structure same as speech recognition
    - Create phrase models by using their transcription
    - Construct the final unique shape statistics from phrase dependent statistics
  - We don't need large amount of training data for each phrase
    - HMMs can be train totally phrase-independent using any transcribed data

Phrase-independent HMM models

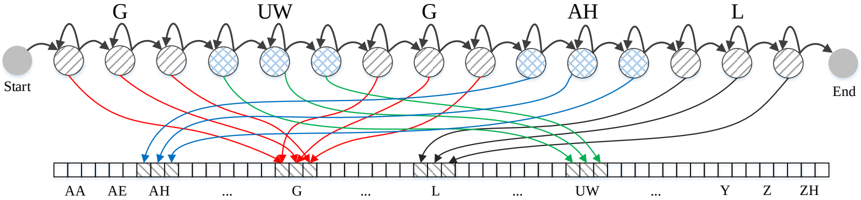


Figure 1: The process of estimating sufficient statistics: In the top, the left-to-right phrase-specific model is shown. The vector in the bottom shows one of the zero or first order statistic vectors. Here, each cell shows a part of the statistics associated with state *s*.

Channel compensation and scoring in TD-SV

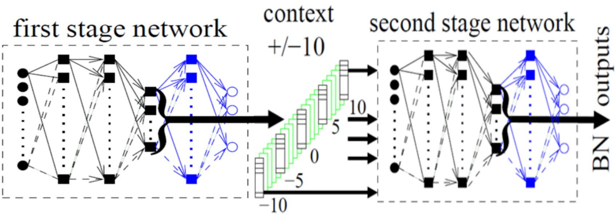
- The performance of PLDA is not acceptable in text-dependent SV [Stafylakis et al. 2013]
- Because of limited training data in TD-SV (number of speakers and samples per phrase), we cannot use simple LDA and WCCN
- We suggest using Regularized WCCN (RWCCN) [RLDA in Friedman, 1989]

$$\mathbf{S}_w = \frac{1}{S}\sum_{s=1}^{S}\left(\alpha\mathbf{I} + \frac{1}{N_s}\sum_{n=1}^{N_s}(\mathbf{w_s}^n - \overline{\mathbf{w}_s})(\mathbf{w_s}^n - \overline{\mathbf{w}_s})^t\right) \quad (4)$$

- We have to use phrase-dependent RWCCN
  - i-vectors of two different phrases are very different especially in HMM alignment
- Cosine similarity is used for scoring and S-Norm for normalization

Using DNNs in TD-SV

- How can we reduce the gap between GMM and HMM alignments?
  - Calculate posterior probabilities using DNNs same as in text-independent SV
  - Using bottleneck (BN) features for improving GMM alignment (the better phone-like feature space clustering obtained)
- Network topology
  - We use Stacked Bottleneck Features [Matejka et al. 2014]
  - Input features: 36 log Mel-scale filter bank outputs augmented with 3 pitch features



Experimental Setup

- Data
  - RSR2015 data set Part I
  - 157 male and 143 female speakers, each pronouncing 30 different phrases from TIMIT in 9 distinct sessions
  - Only the *background* set is used for training, results are reported on the *evaluation* set.
  - Switchboard data is used for training DNNs.
- Features
  - 39-dimensional PLP features and 60-dimensional MFCC features (16kHz)
  - Two 80-dimensional bottleneck features (8kHz)
  - CMVN is applied after dropping initial and final silence.
- Systems
  - 400-dimensional i-vectors length-normalized before RWCCN
  - Phrase dependent RWCCN and S-Norm
  - Cosine distance scoring

GMM, HMM and DNN Alignment Comparison

Table 1: *Comparison of different features and alignment methods.*

		Male			Female		
Features	Alignment	EER [%]	$\mathrm{NDCF}_{\mathrm{old}}^{\mathrm{min}}$	$\mathrm{NDCF}_{\mathrm{new}}^{\mathrm{min}}$	EER [%]	$\mathrm{NDCF}_{\mathrm{old}}^{\mathrm{min}}$	$\mathrm{NDCF}_{\mathrm{new}}^{\mathrm{min}}$
MFCC	GMM	0.67	0.0382	0.1983	0.62	0.0355	0.1991
	HMM	0.37	0.0204	0.1142	0.49	0.0275	0.1533
	DNN	0.36	0.0203	0.1286	0.39	0.0218	0.1441
BN	GMM	0.59	0.0325	0.1564	0.40	0.0201	0.1066
	HMM	0.48	0.0242	0.1446	0.33	0.0151	0.0845
	DNN	0.77	0.0428	0.2026	0.59	0.0296	0.1416
MFCC+BN	GMM	0.31	0.0176	0.0955	0.28	0.0144	0.0898
	HMM	**0.30**	**0.0148**	**0.0927**	**0.27**	**0.0134**	**0.0809**
	DNN	0.43	0.0236	0.1410	0.45	0.0255	0.1291

Final fusion results

Table 2: *Results for different features, concatenated features and score fusions with HMM based systems.*

	Male			Female		
Features	EER [%]	$\mathrm{NDCF}_{\mathrm{old}}^{\mathrm{min}}$	$\mathrm{NDCF}_{\mathrm{new}}^{\mathrm{min}}$	EER [%]	$\mathrm{NDCF}_{\mathrm{old}}^{\mathrm{min}}$	$\mathrm{NDCF}_{\mathrm{new}}^{\mathrm{min}}$
MFCC	0.37	0.0204	0.1142	0.49	0.0275	0.1533
PLP	0.41	0.0217	0.1103	0.42	0.0207	0.1029
BN	0.48	0.0242	0.1446	0.33	0.0151	0.0845
BN1011	0.58	0.0308	0.1780	0.44	0.0193	0.1060
MFCC+BN	0.30	0.0148	0.0927	0.27	0.0134	0.0809
PLP+BN	0.27	0.0149	0.1019	0.27	0.0124	0.0627
MFCC, PLP fusion	0.25	0.0123	0.0712	0.27	0.0139	0.0721
MFCC, BN fusion	0.15	0.0088	0.0493	**0.16**	0.0078	0.0315
PLP, BN fusion	0.18	0.0096	0.0637	0.17	0.0073	0.0326
MFCC, PLP, BN fusion	**0.13**	**0.0070**	**0.0424**	**0.16**	**0.0058**	**0.0299**

Conclusions

- We proved that i-vector also has very good performance in TD-SV
- We verified that DNN based approaches are very effective for the RSR2015 dataset
  - Similar or better verification performance is obtained with DNN based alignment
- Excellent performance was obtained with DNN based bottleneck features especially when concatenated with the standard cepstral features
- In TD-SV, score domain fusion is outperformed feature level fusion unlike text-independent case
- The best results were obtained with a simple score level fusion of the three HMM based i-vector systems

Male results of RedDots Part-01

Method	Non-target trial type	EER [%]	$\text{NDCF}^{\min}_{\text{old}}$	$\text{NDCF}^{\min}_{\text{new}}$
GMM-UBM	Imposter-Correct	1.98	0.0848	0.2879
	Target-Wrong	4.01	0.1733	0.4960
	Imposter-Wrong	**0.34**	0.0135	0.0488
GMM/i-vector (dim: 600)	Imposter-Correct	2.07	0.0899	0.3105
	Target-Wrong	3.76	0.1762	0.4275
	Imposter-Wrong	0.43	0.0153	0.0435
HMM/i-vector (dim: 600)	Imposter-Correct	**1.88**	**0.0809**	0.2271
	Target-Wrong	**1.11**	**0.0338**	**0.0509**
	Imposter-Wrong	0.46	**0.0106**	**0.0228**