CNN-based joint mapping of short and long utterance i-vectors for speaker verification using short utterances

Jinxi Guo, Usha Amrutha Nookala and Abeer Alwan

Department of Electrical Engineering, University of California, Los Angeles, CA, 90095, USA

lennyguo@g.ucla.edu, ushanookala@ucla.edu, alwan@ee.ucla.edu

Abstract

Text-independent speaker recognition using short utterances is a highly challenging task due to the large variation and content mismatch between short utterances. I-vector and probabilistic linear discriminant analysis (PLDA) based systems have become the standard in speaker verification applications, but they are less effective with short utterances. To address this issue, we propose a novel method, which trains a convolutional neural network (CNN) model to map the i-vectors extracted from short utterances to the corresponding long-utterance i-vectors. In order to simultaneously learn the representation of the original short-utterance i-vectors and fit the target long-version i-vectors, we jointly train a supervised-regression model with an autoencoder using CNNs. The trained CNN model is then used to generate the mapped version of short-utterance i-vectors in the evaluation stage. We compare our proposed CNN-based joint mapping method with a GMM-based joint modeling method under matched and mismatched PLDA training conditions. Experimental results using the NIST SRE 2008 dataset show that the proposed technique achieves up to 30% relative improvement under duration mismatched PLDA-training conditions and outperforms the GMM-based method. The improved systems also perform better compared with the matched-length PLDA training condition using short utterances.

Index Terms: speaker verification, text-independent, short utterances, i-vectors, CNNs, joint modeling, PLDA

1. Introduction

The i-vector based framework [1] has defined the state-of-the-art for text-independent speaker recognition. It performs well if long (e.g. more than 30 seconds) enrollment and test utterances are available, but the performance degrades rapidly when not enough data are available. There are several reasons why long utterances perform significantly better than short ones. First, long utterances convey richer speaker-specific information. Second, they have richer phonetic information, which alleviates the context mismatch between enrollment and testing utterances. Third, the variation of the i-vectors extracted from long utterances are smaller than those of short utterances. However, in real applications, it is difficult to collect enough enrollment and test data, and instead short utterances (10 or 5 seconds, for example) are commonly available. To address this issue, a range of techniques has been studied on different aspects of this problem.

There has been a number of methods to model the variation of short utterance i-vectors. In [2], a Full Posterior Distribution PLDA (FP-PLDA) is proposed to exploit the covariance of the i-vector distribution, which improves the standard G-PLDA model [3] by accounting for the uncertainty of i-vector extraction. In [4], the effect of short utterance i-vectors was analyzed, and the duration variability is modeled as additive noise in the i-vector space. The work in [5] introduces a short utterance variance normalization technique and a short utterance variance modeling approach at the i-vector feature level, which makes use of the covariance matrices of long and short i-vectors to do the normalization.

Alternatively, several approaches have been proposed that leverage phonetic information to perform content matching. The work in [6] proposes a Gaussian Mixture Model (GMM) based subregion framework where speaker models are trained for each subregion defined by phonemes. Test utterances are then scored with subregion models. In [7], the authors use the local session variability vectors estimated from certain phonetic components instead of computing the i-vector from the whole utterance. The phonetic classes are obtained by clustering similar senones that are estimated from posterior probabilities of a DNN trained for phone state classification. Another approach was proposed in [8] which matches the zero-order statistics of test and enrollment utterances using posteriors of each phone state, before computing the i-vector.

In addition, a few studies have focused on the role of feature extraction and score calibration. In [9], the authors proposed a DNN-based method to estimate the speaker specific subglottal acoustic features, which are more stationary compared to MFCCs, largely phoneme independent, and can alleviate the phoneme mismatch between training and testing utterances. Besides this, [4] proposes QMF (Quality Measure Function) which is a score-calibration mechanism that compensates for the duration mismatch in the trial scores.

In this paper, motivated by the good properties of long utterance i-vectors, we want to learn a mapping from the short utterance i-vector to a corresponding long utterance version, such that the improved version of i-vectors can be estimated from unseen short utterance i-vectors. To learn such mapping, the authors in [10] proposed a probabilistic approach, in which a GMM-based joint model between long and short utterance i-vectors was trained, and a MMSE (minimum mean square error) estimator was applied to transform a short i-vector to its long version. Here, we propose a novel semi-supervised mapping algorithm which uses CNNs to extract useful information from short utterance i-vectors and maps it to a corresponding long version. Specifically, in order to learn better feature representation and get better generalization for supervised regression, we jointly train the regression model with an autoencoder, which regularizes networks by minimizing the reconstruction error. Our experiments show that the proposed mapping algorithm significantly alleviates the duration mismatch problem when PLDA is trained on long utterances and evaluated on short utterances. This improved system even gives better performance compared with matched-length PLDA training conditions using only short utterances. Therefore, our framework eliminates the necessity to use matched-length PLDA training models for evaluation on different durations.

2. I-vector mapping between short and long sessions

In the following two subsections we describe the GMM-based mapping algorithm as in [10] and our proposed CNN-based mapping algorithm in details.

2.1. GMM-based joint probability model

Let us define two random variables x and y representing i-vectors extracted from short and long sessions respectively. Let $z = [x^T y^T]^T$ denote the joint random vector. The simplest way to model the distribution of z would be a K-component GMM:

$$p(z) = \sum_{k=1}^{K} c_k \mathcal{N}(z; \mu_{z,k}, \Sigma_{z,k}) \tag{1}$$

where $c_k \mathcal{N}(z; \mu_{z,k}, \Sigma_{z,k})$ denotes the probability density function of the k^{th} mixture component, with mean $\mu_{z,k}$, covariance $\Sigma_{z,k}$, and weight c_k.

Once the joint GMM is trained, the marginal and joint statistics of x and y can be obtained by decomposing the mean and covariance matrix as:

$$\mu_{z,k} = \begin{pmatrix} \mu_{x,k} \\ \mu_{y,k} \end{pmatrix} \tag{2}$$

$$\Sigma_{z,k} = \begin{pmatrix} \Sigma_{xx,k} & \Sigma_{xy,k} \\ \Sigma_{yx,k} & \Sigma_{yy,k} \end{pmatrix} \tag{3}$$

Using the MMSE estimator, for a given test i-vector x_0 corresponding to a short utterance, its long version can be computed as the conditional mean of y:

$$\hat{y} = E[y|x_0] = \sum_k p(k|x_0)(f_k x_0 + g_k) \tag{4}$$

where $E[.]$ denotes the expectation operator, and $p(k|x_0)$, f_k and g_k are defined as:

$$p(k|x_0) = \frac{c_k \mathcal{N}(x_0; \mu_{x,k}, \Sigma_{xx,k})}{\sum_{k'=1}^{K} c_{k'} \mathcal{N}(x_0; \mu_{x,k'}, \Sigma_{xx,k'})} \tag{5}$$

$$f_k = \Sigma_{yx,k} \Sigma_{xx,k}^{-1} \tag{6}$$

$$g_k = \mu_{y,k} - \Sigma_{yx,k} \Sigma_{xx,k}^{-1} \mu_{x,k} \tag{7}$$

From Eq.(4), we can observe that the GMM-based mapping is actually a weighted sum of linear functions, and the weights are the conditional probabilities of each Gaussian component given test utterance x_0. Even though the GMM-based joint modeling method gives significant improvement for the mismatched condition between short and long session i-vectors [10], there are still some shortcomings of this method. Learning a mapping from short session i-vector to its long version, is a very complex and nonlinear transform. A weighted sum of linear functions may not be complex enough to model this mapping. Moreover, it's well known that GMMs are statistically inefficient for modeling data that lie on or near a nonlinear manifold in the feature space. Motivated by these reasons, we propose a novel CNN-based learning algorithm, which jointly trains an autoencoder and a supervised regression model to learn this transformation.

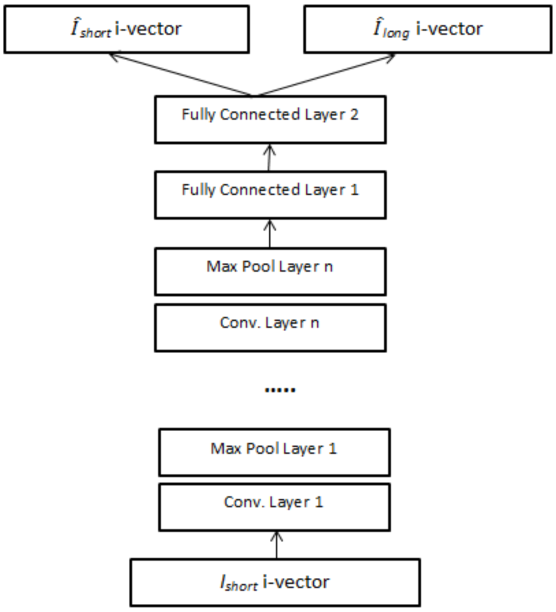

Figure 1: *CNN-based joint i-vector mapping framework*

2.2. CNN-based joint mapping

Mapping short i-vectors to their long version is a many-to-one mapping, i.e. many short session i-vectors can be mapped to the same long session i-vector extracted from the full-length utterance, which they belong to. There are two key factors for this mapping. First, it's important to find a good feature representation of short session i-vectors, which is invariant to different phonemes. Second, there is redundant information from i-vectors for this transformation, which needs to be filtered out. By using multiple convolution layers to extract high-order features and reduce the variability, we can find a good feature representation of i-vectors; by using max pooling layers, we can get rid of redundant information and avoid over-fitting.

In this paper, we use an Alexnet-like [11] neural network structure, which comprises several stacked pairs of convolution and max-pooling layers and two fully connected layers with the output layer on the top. Unlike the regular neural-network-based regression methods, which only map the input tensor to the target tensor, we propose a joint framework of representation learning and supervised regression. More specifically, we not only map the short version i-vector to its long version, but also map it to itself in order to jointly train an autoencoder. The autoencoder can regularize the supervised regression model by forcing the networks to learn the representation of the original short-utterance i-vectors, which can lead to good generalization. Figure 1 shows the framework of the proposed method. We define a new objective function to jointly train the network. Let us denote a target long session i-vector as I_{long} and target short session i-vector as I_{short} (which is the same as the input), and use $\hat{I}_{long}$ and $\hat{I}_{short}$ to represent the output from the regression model and autoencoder respectively. We can define the objective loss function L_{total} which combines the loss from the regression model and autoencoder in a weighted fashion as:

$$L_{total} = \alpha * L_r + (1 - \alpha) * L_a \tag{8}$$

where L_r is the loss of regression model defined as

$$L_r(I_{short}, I_{long}; \theta_r) = \frac{1}{N} \sum_{n=1}^{N} \|\hat{I}_{long} - I_{long}\|^2 \tag{9}$$

and L_a is loss of an autoencoder defined as:

$$L_a(I_{short}, I_{short}; \theta_a) = \frac{1}{N} \sum_{n=1}^{N} \|(\hat{I}_{short} - I_{short})\|^2. \tag{10}$$

Moreover, θ_r and θ_a are parameters of the regression model and autoencoder respectively, which are jointly trained and share the weights before the output layer. α is a scalar weight ranging between 0 and 1. For testing, we only use the output from the regression model as the mapped i-vector.

3. Evaluation setup

3.1. I-vector baseline system

We evaluate our proposed method in the state-of-the-art i-vector/PLDA framework using the Kaldi toolkit [12]. The NIST SRE 2004, 2005, 2006 and Switchboard II datasets are used as development data. The first 20 MFCC coefficients (discarding the zeroth coefficient) and their first and second order derivatives are extracted from the detected speech segments after voice activity detection (VAD). A 20 ms Hamming window, a 10 ms frame shift, and a 23-channel filterbank are used. Only male data are used here. Universal background models with 2048 Gaussian components are trained using a subset of the development dateset (randomly select 8k utterances). The total variability subspace with low rank (400) is trained using all the utterances for male speakers in the development dataset. After the i-vectors are extracted, length normalization is applied and the PLDA model is trained.

For training the i-vector mapping model and PLDA, we select 18601 long utterances, each having more than 60s of speech after VAD, from the development dataset. Short utterances are generated by selecting 5s speech segments from the long utterances and in the end we have around 1 million short utterances with 5s speech each. Each short-utterance i-vector together with its corresponding long-utterance i-vector are used as training pairs (around 1 million pairs) for both GMM-based and CNN-based mapping models.

The NIST SRE 2008 “short2-10sec” male condition is used for evaluation. The enrollment and testing utterances are truncated to 5s speech for our short-utterance speaker verification tasks. There are 7799 test trials representing telephone speech. We will compare the baseline system, GMM-based mapping algorithm and our proposed CNN-based mapping algorithm under the described 5s-5s task.

3.2. Neural network training

CNNs are trained using the Adam optimization strategy [13] with mean square error criterion and scheduled learning rate starting from 0.005. The networks are initialized with 2 different initialization methods, which are Gaussian random normal distributed weights with standard deviation equals to 0.05 (denoted as G-int) and an Xavier initializer [14] (denoted as X-int). We adopt two neural network structures here: 1) 3 convolution layers and 2 fully connected layers (denoted as 3C2F); 2) 5 convolution layers and 2 fully connected layers (denoted as 5C2F). Parameter details are shown in Table 1. The relu activation function is used for all layers. For each layer, before passing the tensors to the nonlinearity function, a batch normalization layer [15] is applied to normalize the tensors and speed up the convergence. For the combined loss, we set equal weights ($\alpha = 0.5$) for both regression and autoencoder loss. The shuffling mechanism is applied on each epoch. All neural networks are trained from scratch. The Tensorflow toolkit [16] is used for neural network training.

CNN	3C2F		5C2F	
structure	filter size	stride	filter size	stride
conv1	7*1*64	1	7*1*64	1
maxpool1	2*1	2	2*1	2
conv2	5*1*192	1	5*1*192	1
maxpool2	2*1	2	2*1	2
conv3	3*1*384	1	3*1*384	1
maxpool3	2*1	2	—	
conv4	—		3*1*256	1
conv5	—		3*1*256	1
maxpool5	—		2*1	2
fc1	512 units			
fc2	512 units			
output	400 (reg), 400 (ae)			

Table 1: *The parameters of the trained CNN structures. ”conv” represents convolutional layer, ”maxpool” represents max-pooling layer and ”fc” represents fully connected layer. ”reg’ represents the regression model and ”ae” represents the autoencoder.*

4. Experiment and results

In this section, we show and discuss the performance of our mapping algorithm under different PLDA training conditions, when only short utterances are available for evaluation. Previous work [10, 17] highlights the importance of duration matching in PLDA model training. For instance when the PLDA is trained using long utterances and evaluated on short utterances, we note considerable degradation in speaker verification performance compared to PLDA trained using matched-length short utterances. Moreover, results in [10] show that irrespective of training conditions of the PLDA model, the long length evaluation utterances always give better performance compared with short evaluation utterances.

Therefore, we show our experimental results under three different PLDA training conditions which use i-vectors extracted from purely long utterances, purely short utterances, and mixed long and short utterances. For mixed long and short utterances case, we choose a similar amount of long and short utterances and add them together for PLDA training. The results of C6 (Telephone speech), C7 (English telephone speech) and C8 (English telephone speech spoken by native U.S. English speakers) conditions are shown for the NIST SRE08 evaluation dataset. We report the best results with absolute and relative improvement for every condition using the Equal Error Rate (EER). As a reference for the short-utterance evaluation task, our speaker verification system gives 2-4% EERs for standard full-length (long) utterance evaluation task (SRE08 core task).

SRE08 5s - 5s	C6	C7	C8
Baseline (no mapping)	30.12	29.23	31.06
GMM (K=1)	28.54	28.46	27.27
GMM (K=3)	27.95	27.31	25.76
CNN (3C2F, G-int)	27.17	25.77	23.48
CNN (5C2F, X-int)	**25.59**	**25.00**	**21.97**
Absolute improvement	4.53 %	4.23 %	9.15 %
Relative improvement	15.04 %	14.47 %	29.46 %

Table 2: *EERs for different mapping methods obtained from a trained PLDA using purely long-utterance i-vectors. C6 represents telephone speech, C7 represents English telephone speech and C8 represents English telephone speech spoken by native U.S. English speaker. "K" represents the number of Gaussian components. "G-int" represents the Gaussian initializer and "X-int" represents the Xavier initializer for CNN initialization. Boldface numbers indicate best results.*

4.1. PLDA using long utterances

In this experiment, we train the PLDA using only long utterance i-vectors. From Table 2, we observe that both CNN and GMM based mapping give significant improvement, and the CNN-based mapping outperforms GMM method considerably in all conditions. For GMM mapping, the Gaussian model with more mixture components gives better performance, but for $K > 3$ conditions, no significant improvement can be further achieved. For CNN mapping, increasing the number of convolutional layers from 3 to 5 with Xavier initialization reduces the EER. This can be attributed to the fact that more convolutional layers can increase the nonlinearity of the mapping network and further reduce the variance. Xavier initialization ensures that the total weights for each layer are equivalent which results in faster convergence and optimal global minima. For CNN (5C2F, X-int), the C8 condition shows the highest improvement with more than 9 % absolute improvement and around 30 % relative improvement from baseline and around 15% relative improvement from the GMM-based (k =3) mapping algorithm. Moreover, with the same configuration we obtained improved performance, compared to the baseline, for C6 and C7 conditions of, 15.04 % and 14.47 % respectively.

4.2. PLDA using mixed short and long utterances

In this experiment, we train the PLDA with mixed short and long utterances. This gives improved baseline results compared to purely long i-vectors conditions, since it takes the short utterances into account for PLDA training. Table 3 illustrates that the results are consistent with the observation of the previous section. Similarly a larger number of Gaussian mixtures improves the GMM-based mapping algorithm, and more convolution layers with Xavier initializer give better performance for CNN-based mapping. We achieved the highest relative improvement of 17.16 % with CNNs (5C2F, X-int) under condition C8, and this algorithm also outperforms the GMM-based (K=3) model by 9.36% relative percent. The best performance of CNN-based mapping method achieved under the mixed-length PLDA training condition is better than the purely long-utterance PLDA training condition.

SRE08 5s - 5s	C6	C7	C8
Baseline (no mapping)	27.95	27.69	26.52
GMM (K=1)	26.38	26.92	25.76
GMM (K=3)	25.98	26.54	24.24
CNN (3C2F, G-int)	25.98	24.62	22.73
CNN (5C2F, X-int)	**24.80**	**24.23**	**21.97**
Absolute improvement	3.15 %	3.46 %	4.55 %
Relative improvement	11.27 %	12.46 %	17.16 %

Table 3: *EERs for different mapping methods obtained from a trained PLDA using mixed long and short utterance i-vectors.*

4.3. PLDA using short utterances

In this experiment, we train the PLDA using only short utterances. This gives the best results among all baseline systems, since the PLDA training and evaluation conditions are matched. It is interesting to note that from our experiments, neither CNN nor GMM mapping algorithms give improvement from the baseline, but instead slightly degrade the performance. The reason might be the fact that the mapped i-vectors may not exactly represent the corresponding long-utterance i-vector. However, our best results using CNN mapping, reported in the previous two sections, still perform better than the baseline we obtained by training PLDA using only short utterance i-vectors. We show the best results we obtain from the previous sections using GMM-based and CNN-based models in Table 4. In fact, we note that the best results we obtained using GMM-based mapping are similar to the current baseline and under the C8 condition it gives worse performance, while our proposed CNN-based mapping algorithm is consistently better than the baseline system under all conditions.

SRE08 5s - 5s	C6	C7	C8
Baseline (short utterance PLDA)	26.57	26.54	23.48
Best results of GMM (mixed-length utterance PLDA)	25.98	26.54	24.24
Best results of CNN (mixed-length utterance PLDA)	**24.80**	**24.23**	**21.97**
Absolute Improvement	1.77	2.31	1.51
Relative Improvement	6.7 %	8.7%	6.4 %

Table 4: *EER for the baseline obtained from a trained PLDA using purely short-utterance i-vectors and EER for the best results of different mapping methods obtained from a trained PLDA using mixed short and long utterance i-vectors*

5. Conclusion

In this paper, we propose a novel semi-supervised mapping algorithm, which jointly learns a supervised mapping from short utterance i-vector to its long version with an autoencoder using CNNs. When evaluated using the NIST SRE 08 dataset, the mapped short i-vectors can give up to around 30% relative improvement for short utterances speaker verification under mismatched training conditions, and also outperform the matched-PLDA training condition using short utterances. For future work, we will investigate and compare different neural network structures, and the impact of the weights for the combined regression and autoencoder loss. We will also compare the i-vector mapping results for both GMM-ivector and DNN-ivector systems.

6. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis For Speaker Verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[2] S. Cumani, "Fast scoring of full posterior PLDA models," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 11, pp. 2036–2045, 2015.

[3] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," *Interspeech*, 2011, pp. 249–252.

[4] T. Hasan, R. Saeidi, J. H. Hansen, et al, "Duration mismatch compensation for i-vector based speaker recognition systems," *ICASSP*, 2013, pp. 7663–7667.

[5] A. Kanagasundaram, D. Dean, S. Sridharan, et al, "Improving short utterance i-vector speaker verification using utterance variance modeling and compensation techniques," *Speech Communication*, 2014, 59:69–82.

[6] L. Li, D. Wang, C. Zhang, and T. Z. Zheng, "Improving short utterance speaker recognition by modeling speech unit classes," *IEEE Transactions on Audio, Speech, and Language Processing*,vol. 24,pp. 1129–1139, 2016.

[7] L. Chen, K. A. Lee, E. S. Chng, B. Ma, H. Li and L. R. Dai, "Content-aware local variability vector for speaker verification with short utterance," *ICASSP*, 2016, pp. 5485–5489.

[8] N. Scheffer and Y. Lei, "Content matching for short duration speaker recognition," *Interspeech*, 2014, pp. 1317-1321.

[9] J. Guo, G. Yeung,, D. Muralidharan, H. Arsikere, A. Afshan and A. Alwan, "Speaker Verification Using Short Utterances with DNN-Based Estimation of Subglottal Acoustic Features," *Interspeech*, 2016, pp. 2219–2222.

[10] W. B. Kheder, D. Matrouf, M. Ajili, J. F. Bonastre, "Probabilistic Approach Using Joint Long and Short Session i-Vectors Modeling to Deal with Short Utterances for Speaker Recognition," *Interspeech*, 2016, pp. 1830–1834.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, 2012, pp. 1106-1114.

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al, "The Kaldi speech recognition toolkit," *Proc. of ASRU*, 2011, pp. 1-4.

[13] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] X. Glorot, Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Aistats*, 2010.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, et al, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467*, 2016.

[17] A. K. Sarkar, D. Matrouf, P. M. Bousquet, and J. F. Bonastre, "Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification," *Interspeech*, 2012, pp. 2662–2665.