Improved Speaker Adaptation by Combining I-Vector and fMLLR with Deep Bottleneck Networks

Thai Son Nguyen, Kevin Kilgour, Matthias Sperber and Alex Waibel

Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology
thai.nguyen@kit.edu

Abstract. This paper investigates how deep bottleneck neural networks can be used to combine the benefits of both i-vectors and speaker-adaptive feature transformations. We show how a GMM-based speech recognizer can be greatly improved by applying feature-space maximum likelihood linear regression (fMLLR) transformation to outputs of a deep bottleneck neural network trained on a concatenation of regular Mel filterbank features and speaker i-vectors. The addition of the i-vectors reduces word error rate of the GMM system by 3-7 % compared to an identical system without i-vectors. We also examine Deep Neural Network (DNN) systems trained on various combinations of i-vectors, fMLLR-transformed bottleneck features and other feature space transformations. The best approach results speaker-adapted DNNs which showed 15-19 % relative improvement over a strong speaker-independent DNN baseline.

Keywords: DNN, fMLLR, i-vector, bottleneck extraction

1 Introduction

In statistical speech recognition, speaker adaptation techniques can fall into two categories: Model adaptation involves modifying the parameters of the acoustic model to fit the actual speech data from a target speaker. Maximum Likelihood Linear Regression (MLLR) [5] and Maximum A Posteriori (MAP) [7] are the powerful model adaptation techniques that improve Gaussian Mixture Models (GMMs). However, there is no similar technique for Deep Neural Network (DNN) models which have become prominent in recent years. Due to their many large hidden layers, DNNs have a significantly higher number of parameters. It is therefore hard to adapt DNNs with only a small amount of data. Several studies [11] [19] have shown that DNN models have greater invariance to speaker variations resulting in model adaptation being less effective than for GMMs. Further, model adaptation usually results in new models for individual speakers, significantly increasing complexity and required storage space.

Unlike model adaptation, feature adaptation techniques use regular acoustic features and adaptation data to provide new features which better fit the trained acoustic model, thus improving recognition accuracy without the need

to change the model. Feature adaptation is attractive for dealing with the limitations of model adaptation, especially for DNNs. Feature-space MLLR (fMLLR) [5] is a well-known adaptation technique which makes better inputs for GMMs. However, providing good fMLLR features for DNNs is challenging: Due to the huge difference between DNN and GMM models, fMLLR features which are optimized for GMMs are not guaranteed to be better for DNNs than other regular features. Recently, identity vectors (i-vectors) for speaker representation have been introduced [3], and have been successfully used in speaker verification and speaker recognition. Further research [18, 20] proved that i-vectors can be used in conjunction with regular features to improve DNN performance.

In this paper we examine how i-vectors and fMLLR transformations can be combined in order to improve both GMM and DNN systems. In particular we analyse speaker-adaptive bottleneck features (SA-BNF), where log scale Mel filterbank (FBANK) features are concatenated with i-vectors to form their input features and investigate how both speaker-adaptive bottleneck features and speaker-independent bottleneck features can be further transformed and augmented before being used as DNN or GMM input features.

The paper is organized as follows: Section 2 reviews speaker adaptation using fMLLR and i-vector techniques. In Section 3, the hierarchical combination of fMLLR and i-vectors is presented. The experiments and results are explained in sections 4 and 5. In Section 6, we conclude and discuss future work.

2 Speaker Adaptation Using fMLLR and i-Vector

fMLLR is a commonly used adaptation technique for ASR systems. When a small amount of adaptation data for an individual speaker is available, fMLLR can be applied with a trained GMM to employ an affine transformation which transforms acoustic features for speaker normalization. The transformed features are well-known to be better inputs for the GMM system. [16] showed DNN systems can also be improved when using fMLLR features. In their study, the best input features for DNN system are obtained using a sequence of transformations including Linear Discriminate Analysis (LDA), global Semi-tied covariance (STC) and fMLLR. The authors presented 3% absolute improvement of the proposed DNN over a very good adapted GMM. In [14], the authors proposed to estimate fMLLR transforms using simple target models (STM) and combine with FBANK features to improve DNN performance.

I-vectors describes a speaker's identity and are successfully used in speaker verification and speaker recognition tasks. This powerful technique is also useful for speech recognition since i-vectors encapsulate the speaker relevant information in a low-dimensional representation. Applied to speech recognition, [18] and [20] augment regular acoustic features with i-vectors as a speaker adaptation for their DNNs. Both works showed that i-vectors possibly provide additional information allowing for an improving recognition performance. Saon et al. presented 10% relative improvement on 300 hours of Switchboard data, while Senior et al. just showed 4% relative improvement on 1700 hours of Google Voice data.

[12] introduced speaker adaptive training for DNN (SAT-DNN) which learns an adaptation neural network to convert i-vectors to speaker-specific linear feature shifts. The original features (e.g. MFCC) are then speaker-normalized by adding theses shifts. Their SAT-DNN model achieved 13.5% relative improvement on 118 hours of TED talks. In [1] i-vectors are incorporated with a bottleneck extraction architecture to improve low-resource ASR systems.

Recently, Tan et al. [22] have investigated to use i-vectors at different layers of a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to normalise speaker variability. They reduced word error rates by 6.5% relative when using FMLLR features which are transformed from MFCC with LDA and STC. Tomashenko et al. [23] proposed to use bottleneck features for GMM-derived feature extraction and combine with fMLLR features to be DNN input.

In terms of speaker adaptation, fMLLR tries to remove speaker variability while i-vector provides more speaker information. Both techniques help to improve feature processing in different ways. The aforementioned study [18] also proposed to simply augment their fMLLR features with i-vectors to further improve their recognition results. Our study is motivated by that paper and investigates how to best combine fMLLR and i-vectors.

3 Combining i-vector and fMLLR

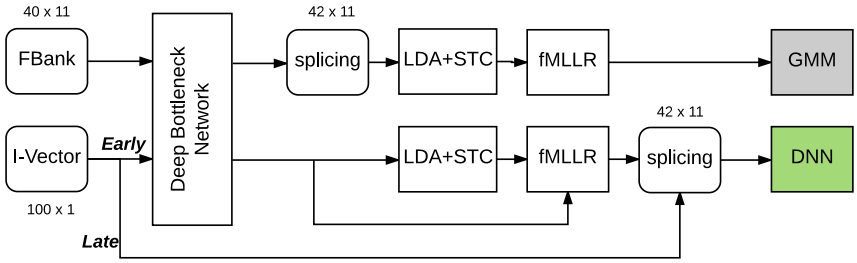

Fig. 1. Hierarchical combination of bottleneck, fMLLR and i-vector features for either early or late combinations.

Deep bottleneck network (DBNF) has been shown to extract effective speaker-independent bottleneck features (SI-BNF) to both GMM and DNN models. In this study, we use DBNF to perform several combinations of i-vector and fMLLR adaptations. These combinations yield improved speaker-adapted features for GMMs and DNNs. An overview of our proposed feature extraction process is shown in Figure 1. We propose to use i-vectors as additional features to a DBNF for extracting speaker-adapted bottleneck features, then perform fMLLR and other linear transformations before feeding them to GMM systems. To build speaker-adapted features for DNNs, we combine i-vectors and fMLLR features that are estimated on top of the bottleneck features.

In Sections 3.1 and 3.3 we discuss two different approaches of integrating i-vectors. Section 3.2 explains how fMLLR and other transformations can be used in our feature extraction pipeline.

3.1 Early I-Vectors

Having a similar architecture to DNNs, DBNFs are also capable of modeling high-dimensional correlated input features. We investigate the ability of incorporating acoustic features and i-vectors to train DBNFs. In our approach, regular acoustic features (e.g. FBANK) are spliced for 11 consecutive frames and then concatenated with i-vector features to be fed into DBNFs. After the training, we are able to build speaker-adapted bottleneck neural networks which can extract speaker-adapted bottleneck features (SA-BNF).

3.2 fMLLR

LDA can be used to extract the most useful features for the classification from many consecutive acoustic frames while STC transformation is applied to decorrelate the input features. These techniques are popularly adopted to transform Mel-frequency cepstral coefficients (MFCC) [16] or bottleneck features [8, 25] to become effective input for GMM models. Then, fMLLR transformation is further estimated and applied to make the acoustic data of individual speakers more accurately modeled by the trained GMMs. We also perform these transformations on top of SA-BNF to build speaker-adapted features for the GMMs. However, to keep the same temporal context of frames fed into the DNNs (i.e., wider context reduces the classification performance), we do not use LDA for feature-dimensionality reduction from concatenated features. Instead, we propose to either estimate fMLLR transformation directly on SI-BNF or SA-BNF without using LDA and STC transformations, or use them without applying context-window. So that, we can later splice 11 frames of fMLLR features as the input to the DNNs.

3.3 Late I-Vectors

After applying fMLLR transformation, new transformed features are supposed to have less speaker variability. Providing again speaker information with i-vectors can lead to improvement as suggested from [18]. We also concatenate the transformed SI-BNF or SA-BNF with i-vectors for different DNN input features.

4 Experimental Setup

4.1 Overall Setup

In the experiments, we used a big training dataset of 460 hours from 12000 English talks. This dataset includes the TED-LIUM [17], Quaero [21] and Broadcast News [9] corpora. We used the tst2013 and the tst2014 sets from the IWSLT evaluation campaign [2] which sequentially contains 27 and 15 talks.

The DBNFs were constructed with 5 hidden layers containing 2000 units, followed by a 42 units bottleneck layer and the final classification layer, using input as 11 stacked frames of 40-dimensional mel scale filterbank coefficients with or without concatenating i-vector features. All the DNN models also share the same architecture which has 6 hidden layers with 2000 units per layer. The input of the DNNs is 11 stacked frames of 42-dimensional transformed SI-BNF or SA-BNF, with or without combining i-vector features. We used sigmoid activation for hidden layers and soft-max for output layer.

DNN and DBNF systems were trained using cross-entropy loss function to predict 8000 context-dependent states. The same training method is applied for all DNNs and DBNFs, which includes pre-training with denoising auto-encoders and followed by fine-tuning with back-propagation. We used an exponential schedule for all of the trainings. The GMM models were trained using incremental splitting of Gaussians (MAS) [10] and followed by optimal space training (OFS) (a variant of STC [6]) if LDA features are used.

The systems were decoded using Janus Recognition Toolkit (JRTK) [4] with the Cantab 4-gram language model [24] from more than 150k words.

4.2 I-vector Extraction

To extract i-vectors, a full universal background model (UBM) with 2048 mixtures was trained on the training dataset using 20 Mel-frequency cepstral coefficients with delta and delta-delta features appended. The total variability matrices were estimated for extracting 100-dimensional i-vector which was observed to give the optimal recognition performance in [18] [20]. The UBM model training and i-vector extraction were performed by using the sre08 module from the Kaldi toolkit [15].

4.3 fMLLR Estimation

The GMMs trained with SI-BNF and SA-BNF were used to compute fMLLR transformations. The process of fMLLR estimation were performed as the traditional approach. During the training, we used the adaptation data of the same speaker and the reference transcriptions to do the alignment, while the same GMMs were used as first-pass systems to generate transcriptions in the testing.

5 Experimental Results

5.1 Baseline Systems

In our experiments, we used a DNN system with FBANK features as the speaker independent baseline (SI-DNN). This is a strong baseline since DNNs training with mel scale filterbank is known to outperform other regular features [13]. The other baseline is a speaker-adapted DNN (SA-DNN) using i-vectors. This baseline is similar to the speaker-adapted DNNs presented in [20] except our

i-vectors are extracted for speaker-level instead of utterance-level. The results of the baselines on two test sets are shown in Table 1. In our setup, we are able to reproduce the improvement when using i-vector adaptation for DNN systems in both the test sets. The improvement is not large as reported in [18], but is comparable to [20] since we used a similar baseline setup.

Table 1. Word error rate of baseline systems.

Baseline	tst2013 (tst2014)
SI-DNN	16.2 (12.9)
SA-DNN	**15.1** (**12.4**)

5.2 Results from GMM Systems

Table 2 presents the results of our evaluated GMM systems. The first three columns show the possible techniques applied to make inputs to the GMMs. The techniques include *Early I-vector* for extracting speaker-adapted bottleneck features, followed by splicing and *LDA+STC* transformations, and *fMLLR* transformation at the last step. The last column presents word error rates (WER) on the both test sets.

By using discriminative bottleneck features, the GMM systems can achieve good recognition performance which is close to the DNN baseline. This also explains the smaller gains of applying *LDA+STC* and *fMLLR* transformations than performing on regular acoustic features such as MFCC. However, these techniques have still been effective when producing constant improvements over different test sets.

The results of the GMMs using SA-BNF are consistently better than using SI-BNF with identical constructions. The regular bottleneck GMM (with full transformation techniques) is 3-7 % less effective than the adapted bottleneck GMM. This shows that DBNF can explore the adapted input with the addition of i-vectors to provide better discriminative features.

Table 2. Comparison of word error rate for GMM systems using context-window of 462 bottleneck features.

Early I-vector	Splice+LDA+STC	fMLLR	tst2013 (tst2014)
✗	✗	✗	16.7 (13.1)
✗	✓	✗	15.9 (12.5)
✗	✓	✓	15.4 (12.3)
✓	✗	✗	15.7 (12.7)
✓	✓	✗	14.9 (12.4)
✓	✓	✓	**14.4** (**11.9**)

In Table 3, we present the performance of different GMMs that were used to estimate fMLLR features for DNN systems. Without using context-window of bottleneck features, the combination of *LDA+STC* transformations shows less effective. However, using *fMLLR* and *Early I-vector* still presents achievable improvements.

It is worth noting that while the trained GMM systems have good performance, the best speaker-adapted GMM is even better than SA-DNN baseline. This indicates that feeding their input features to DNNs may improve systems due to the better capacity of DNNs in classification task.

Table 3. Comparison of word error rate for GMM systems using 42 bottleneck features.

Early I-vector	LDA+STC	fMLLR	tst2013 (tst2014)
✗	✓	✗	16.5 (12.9)
✗	✗	✓	16.1 (12.5)
✗	✓	✓	15.8 (12.3)
✓	✓	✗	15.5 (12.7)
✓	✗	✓	15.0 (12.0)
✓	✓	✓	15.1 (12.2)

5.3 Results from DNN Systems

In Table 4, we compare the results of the examined DNNs using transformed SI-BNF with or without the addition of *Late I-vector*. Again, the last column shows the results in word error rates, while the other columns indicates the usage of our proposed adaptation techniques.

Table 4. Comparison of word error rate for DNN systems.

LDA+STC	fMLLR	Late I-vector	tst2013 (tst2014)
✗	✗	✗	15.3 (12.4)
✓	✗	✗	15.4 (12.5)
✗	✓	✗	14.5 (11.8)
✓	✓	✗	14.8 (12.1)
✗	✗	✓	14.1 (11.5)
✓	✗	✓	14.8 (12.7)
✗	✓	✓	**13.1** (**11.1**)
✓	✓	✓	13.7 (11.3)

Interestingly, *LDA+STC* transformations which usually produce better input to GMM modeling show a negative effect when applying to DNN inputs. However, performing *fMLLR* and *Late I-vector* adaptations on bottleneck features individually show effectiveness. When concatenating fMLLR transformed features with i-vectors, we found the best features combination. The best DNN

system with *fMLLR* and *Late I-vector* gives 15-19 % relative improvement over SI-DNN baseline and 11-13% over SA-DNN baseline.

Since SA-BNF features have been effective to GMM modeling, we also investigate to see if the DNNs can be also benefited from it. Table 5 compares the DNNs with SA-BNF against SI-BNF. Using *fMLLR* transformation on top of SA-BNF can improve the performance up to 8% relative. We could not however achieve further improvement with the DNNs by *Late I-vector* together with *Early I-vector* and *fMLLR*. That may be due to either fMLLR transformation not being able to completely remove speaker variability, or our used DNN architecture not being able to exploit this combined structure.

Table 5. Comparison of DNN systems with SA-BNF against SI-BNF.

Early I-vector	fMLLR	Late I-vector	tst2013 (tst2014)
✗	✗	✗	15.3 (12.4)
✓	✗	✗	14.6 (12.6)
✗	✓	✗	14.8 (12.1)
✓	✓	✗	14.1 (11.5)
✗	✓	✓	13.1 (11.1)
✓	✓	✓	13.7 (11.2)

6 Conclusion and Future Work

We have presented an effective way of combining deep bottleneck network with i-vectors and fMLLR to produce speaker-adapted features for ASR systems. In our experiments, a GMM system with speaker-adapted bottleneck features outperforms a regular bottleneck GMM system with 3-7 % relative improvement, while a DNN system even achieves higher improvements of 15-19 % over a strong DNN baseline. Since the used deep bottlenecks network is open to modeling a variety of different input features, the replacement of *Late I-vector* or *Late I-vector* with other speaker codes, or FBANK with other single or multiple regular features can be done without changing the feature extraction pipeline. A further study can go in this direction to better explore speaker-adapted bottlenecks features.

References

1. Cardinal, P., Dehak, N., Zhang, Y., Glass, J.: Speaker adaptation using the i-vector technique for bottleneck features. In: Proceedings of Interspeech. vol. 2015 (2015)
2. Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Frederico, M.: Report on the 10th iwslt evaluation campaign. In: The International Workshop on Spoken Language Translation (IWSLT) 2013 (2013)
3. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19(4), 788–798 (2011)

4. Finke, M., Geutner, P., Hild, H., Kemp, T., es, K.R., Westphal, M.: The Karlsruhe VERBMOBIL speech recognition engine. In: Proc. of ICASSP (1997)
5. Gales, M.J.: Maximum likelihood linear transformations for HMM-based speech recognition. Computer speech & language 12(2), 75–98 (1998)
6. Gales, M.J.: Semi-tied covariance matrices for hidden markov models. IEEE Transactions on speech and audio processing 7(3), 272–281 (1999)
7. Gauvain, J.L., Lee, C.H.: Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE transactions on speech and audio processing 2(2), 291–298 (1994)
8. Gehring, J., Miao, Y., Metze, F., Waibel, A.: Extracting deep bottleneck features using stacked auto-encoders. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 3377–3381 (2013)
9. Graff, D.: The 1996 broadcast news speech and language-model corpus. In: Proceedings of the DARPA Workshop on Spoken Language technology. (1997)
10. Kaukoranta, T., Franti, P., Nevalainen, O.: A new iterative algorithm for VQ codebook generation. In: Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on. vol. 2, pp. 589–593 (1998)
11. Liao, H.: Speaker adaptation of context dependent deep neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 7947–7951. IEEE (2013)
12. Miao, Y., Zhang, H., Metze, F.: Speaker adaptive training of deep neural network acoustic models using i-vectors. IEEE/ACM Transactions on Audio, Speech, and Language Processing 23(11), 1938–1949 (2015)
13. Mohamed, A.r., Hinton, G., Penn, G.: Understanding how deep belief networks perform acoustic modelling. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4273–4276 (2012)
14. Parthasarathi, S.H.K., Hoffmeister, B., Matsoukas, S., Mandal, A., Strom, N., Garimella, S.: fmllr based feature-space speaker adaptation of DNN acoustic models. In: Proceedings of Interspeech. vol. 2015 (2015)
15. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al.: The Kaldi speech recognition toolkit. In: IEEE 2011 workshop on automatic speech recognition and understanding. No. EPFL-CONF-192584 (2011)
16. Rath, P.S., Povey, D., Veselý, K., Černocký, J.: Improved feature processing for Deep Neural Networks. In: Proceedings of Interspeech 2013. pp. 109–113. No. 8 (2013)
17. Rousseau, A., Deléglise, P., Estève, Y.: Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks. In: LREC (2014)
18. Saon, G., Soltau, H., Nahamoo, D., Picheny, M.: Speaker adaptation of neural network acoustic models using i-vectors. In: ASRU. pp. 55–59 (2013)
19. Seide, F., Li, G., Chen, X., Yu, D.: Feature engineering in context-dependent deep neural networks for conversational speech transcription. In: Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on. pp. 24–29. IEEE (2011)
20. Senior, A., Lopez-Moreno, I.: Improving DNN speaker independence with i-vector inputs. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2014)
21. Stüker, S., Kilgour, K., Kraft, F.: Quaero 2010 speech-to-text evaluation systems. In: High Performance Computing in Science and Engineering'11, pp. 607–618. Springer (2012)

22. Tan, T., Qian, Y., Yu, D., Kundu, S., Lu, L., Sim, K.C., Xiao, X., Zhang, Y.: Speaker-aware training of LSTM-RNNS for acoustic modelling. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5280–5284. IEEE (2016)
23. Tomashenko, N., Khokhlov, Y., Esteve, Y.: On the use of gaussian mixture model framework to improve speaker adaptation of deep neural network acoustic models. In: Proceedings of INTERSPEECH (2016)
24. Williams, W., Prasad, N., Mrva, D., Ash, T., Robinson, T.: Scaling recurrent neural network language models. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5391–5395 (2015)
25. Yu, D., Seltzer, M.L.: Improved bottleneck features using pretrained deep neural networks. In: Interspeech. vol. 237, p. 240 (2011)