On the Training of DNN-based Average Voice Model for Speech Synthesis

Shan Yang*, Zhizheng Wu† and Lei Xie*
* Shaanxi Provincial Key Lab of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
E-mail:{syang, lxie} @nwpu-aslp.org
† The Centre for Speech Technology Research, University of Edinburgh, UK
E-mail:zhizheng.wu@ed.ac.uk

***Abstract*—Adaptability and controllability are the major advantages of statistical parametric speech synthesis (SPSS) over unit-selection synthesis. Recently, deep neural networks (DNNs) have significantly improved the performance of SPSS. However, current studies are mainly focusing on the training of speaker-dependent DNNs, which generally requires a significant amount of data from a single speaker. In this work, we perform a systematic analysis of the training of multi-speaker average voice model (AVM), which is the foundation of adaptability and controllability of a DNN-based speech synthesis system. Specifically, we employ the i-vector framework to factorise the speaker specific information, which allows a variety of speakers to share all the hidden layers. And the speaker identity vector is augmented with linguistic features in the DNN input. We systematically analyse the impact of the implementations of i-vectors and speaker normalisation.**

I. Introduction

Recently, statistical parametric speech synthesis (SPSS) has attracted significant attentions because of its adaptability and controllability to the synthesised voice. By adapting/controlling the model parameters, voices with different characteristics can be easily generated. In the last decade, hidden Markov models (HMMs) were successfully applied to SPSS, in which the Gaussian mixture model (GMM) was used to model the observations of hidden states [1], [2]. Although the GMM-HMM framework can model the relationship between linguistic features and acoustic parameters, the decision tree based clustering cannot generalise well for unseen context and limits the naturalness of synthesised speech [3], [4]. Recent studies have shown that SPSS has been considerably advanced by deep neural networks (DNN) [3], [5], [6], [7], [8], [9], [4]. DNN and other neural network models can learn a direct, layered, non-linear model from linguistic features to acoustic parameters without decision tree clustering. However, current studies on SPSS are mainly speaker-dependent: a significant amount of data from a single speaker is required to build a stable acoustic model, and sometimes the quality of training data has a great influence on the naturalness of synthesised speech.

To explore the adaptability and controllability of SPSS, a significant amount of work has been done in the HMM-based framework. In [10], maximum likelihood linear regression (MLLR) was applied to the speaker adaptation model in order to transform voice characteristics to the target speaker. Then the multi-space probability distribution HMM (MSD-HMM) was used to simultaneously model and adapt excitation and spectral parameters [11]. Average-voice-based speech synthesis using hidden semi-Markov model (HSMM) also showed the adaptability of HMMs [12]. In [13], Yamagishi *et al.* provided a systematic analysis of HMM-based speaker adaptation techniques. They also proposed a constrained structural maximum a posterior linear regression (CSMAPLR) method for HMM-based adaptation. In [14], a speaker adaptive system was built, which employed several kinds of effective adaptation methods such as CSMAPLR+MAP and feature-space adaptive training. All these studies show promising adaptability of HMM with a small amount of adaptation data.

Recently, several studies have been conducted to assess the adaptability and controllability of DNN-based speech synthesis. In [15], Fan *et al.* proposed a multi-speaker DNN model, where the same hidden layers were shared among different speakers while the output layers were composed of speaker-dependent nodes explaining the target of each speaker. The hidden layers were further transferred for a new speaker with limited training data. In this approach, only a few speakers were considered for the shared DNN model and parallel data from multiple speakers were assumed in the model training. They further extended their multi-speaker DNN to a speaker and language factorization model [16]. Recently, Wu *et al.* [17] proposed three DNN-based speaker adaption methods at different levels: input, hidden layers and output. Specifically, at the DNN input level, they augmented a low-dimensional speaker-specific vector (i-vector) with linguistic features as input to represent speaker identity. An average voice model (AVM) with augmented i-vector was trained from multiple speakers. Different from the multi-speaker DNN model [15], the AVM+i-vector model was trained from a variety of speakers and the output layer was shared by all the speakers. At the adaptation phase, the target speaker's i-vector was first estimated by using the adaptation data, and then the i-vector was appended with linguistic features as input to generate the target speaker's voice. The advantages of this approach are obvious: (1) the AVM training does not need parallelled data from different speakers; (2) to synthesize the voice of a target speaker, the AVM model does not need to

be re-trained or fine-tuned using the target data. In [18], a prosodic controlling vector, which is similar to the idea of i-vectors, was introduced to DNN-based speech synthesis to control the global prosodic characteristics.

As we know, average voice model (AVM) is the basis and the key to the success of speaker adaptation. However, to the best of our knowledge, a systematic analysis of the training of AVM is still missing. To bridge the gap, we present a systematic analysis on the training of AVM in this study. We follow the i-vector framework presented in [17] rather than that in [15], as the i-vector framework allows us to model a large number of speaker without assuming parallel data. We aim to answer the following questions through the analysis:

- **How to do normalisation for acoustic features?** As the AVM training involves acoustic features from multiple speakers, it might be important to know how to effectively normalise the acoustic features.
- **Is i-vector effective to improve the performance of AVM?** I-vector is a low-dimensional vector representing speaker individuality and has been widely used in speaker recognition [19]. We expect that with the help of this speaker identity vector, the speech synthesis performance might be improved.
- **How to extract i-vectors, and how many dimensions are enough.** To make the i-vector more robust and compact, linear discriminant analysis (LDA) [20] is usually adopted. We are interested in the impacts from different i-vector dimensions after LDA.

The answers to these questions will help us to understand how to train a better AVM, and how to maximise the adaptability and controllability in the future studies of DNN-based speech synthesis.

II. Average Voice Model Training

In this work, we follow the i-vector framework proposed in [17], in which all the speakers share all the DNN layers, including the output layer. The i-vector, which represents speaker identity [21], is used to control the network to produce the speaker's voice. The framework is presented in Figure 1. In the framework, an i-vector and a gender code are appended with the speaker-independent linguistic features as the network input. The i-vector and gender code are used as speaker-dependent features to discriminate among different speakers. We will analyse the impact of acoustic feature normalisation and the implementation of i-vector to the performance of an average voice model (AVM).

A. Speaker normalisation

As each speaker has its own characteristics, it is useful to remove those speaker variations when training an AVM. In [15], a so-called multi-task learning was applied to give each speaker a private output layer, and the bottom layers were shared by all the speakers. However, to maximise the flexibility and controllability, it is important to share all the layers for all the speakers. For example, to generate arbitrary

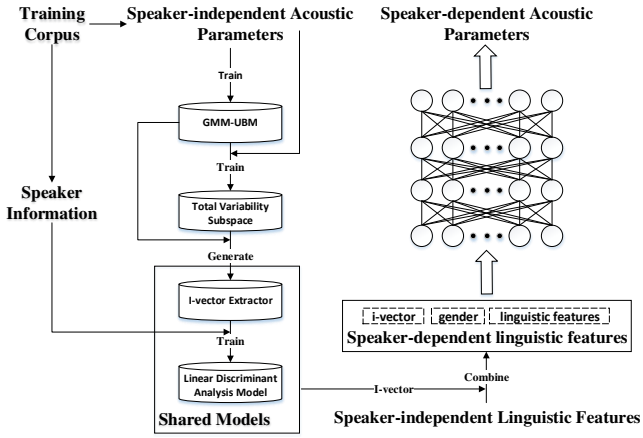

Fig. 1. The average voice model (AVM) framework used in this work, which takes i-vector, gender code and linguistic features as system input to predict vocoder parameters which are also called acoustic features.

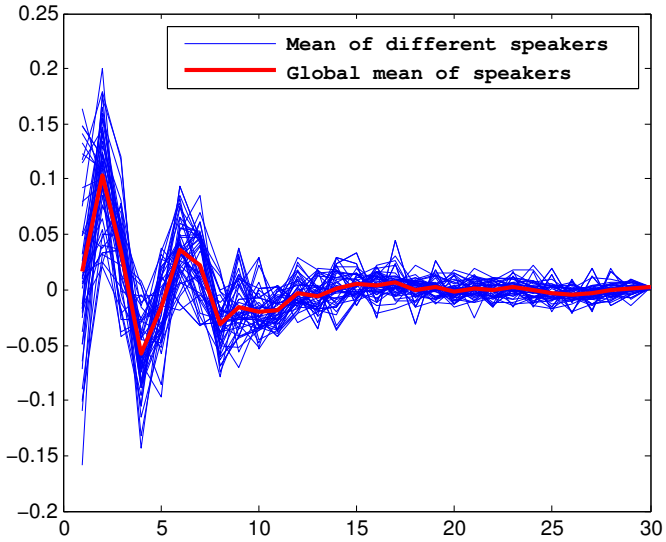

Fig. 2. Variation of the average Mel-Cepstral Coefficients (first 30 coefficients) from 44 male speakers in the VCTK corpus.

speaker's voice without his/her training data, it is impossible to train its private layer.

We propose to do speaker-dependent normalisation, rather than global normalisation. Figure 2 presents the cepstral means of 44 male speakers in the VCTK corpus, as well as the global mean of those speakers. It demonstrates that doing global mean-variance normalisation cannot catch the speaker variations when training an AVM.

B. I-vector extraction

I-vector is a low-dimensional representation of speaker identity, which is realised by projecting a speaker supervector (i.e. GMM supervector) into a low-dimensional subspace via factor analysis technique [21]. A speaker-dependent GMM supervector can be modeled as,

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

where $\mathbf{s}$ is the speaker supervector, $\mathbf{m}$ is the mean supervector of speaker-independent universal background model (UBM), $\mathbf{T}$ is the total variability matrix representing speaker space. The speaker space is also called total variability subspace (TVS). $\mathbf{i}$ is the speaker identity vector, i.e., i-vector, which is assumed to follow a standard normal distribution.

In practice, the total variability matrix $\mathbf{T}$ and the UBM model are gender-dependent, and maximum a posterior is used to estimate the speaker supervector from the speaker-independent UBM model.

C. Linear discriminant analysis

To make the i-vector more robust and compact, linear discriminant analysis (LDA) is usually adopted. LDA is a discriminant analysis method, which finds the best identification vector space to represent the high-dimensional samples, and tends to make samples have the minimum inter-class variance and the maximum intra-class variance [22]. Suppose there are N different classes with mean μ_i, then the scatter between class variability can be defined as

$$\sum\nolimits_b = \frac{1}{N}\sum_{i}^{N}(\mu_i - \mu)(\mu_i - \mu)^T \tag{2}$$

where μ is the mean of all classes. Assuming all classes have the same covariance, the Fisher criterion for multi-class LDA can be maximized in the form [22], [23]:

$$S = \frac{\sigma^2_{between}}{\sigma^2_{within}} = \frac{V^T \sum\nolimits_b V}{V^T \sum V} \tag{3}$$

where σ defines the variance between the classes and within classes, S is the separation between these two variances, and $\sum$ is the covariance of all samples. So the main purpose of LDA is to find a suitable V that maximizes S. Since it is a Rayleigh quotient, when V is the generalized eigenvectors of $\sum^{-1}\sum_b$, S will be equal to the corresponding eigenvalue [23].

Given the generated i-vectors and their corresponding speaker labels, we do LDA under the fisher criterion with the assumption that each speaker has the same covariance. In this way, we take the transformation matrix V to reduce the dimension of i-vector using different number of components.

III. Experiments

A. Experimental setups

In the experiments, the VCTK corpus[1] was used, which contains speech data from 103 speakers, including 44 male and 59 female speakers. Each speaker has around 400 sentences, and in total 41,294 sentences. We took 40,294 randomly selected sentences for AVM training, 500 sentences as development set and another 500 sentences as testing set. The sampling rate of speech files was 48 kHz. the STRAIGHT vocoder [24] was used to extract 60-dimensional Mel-Cepstral Coefficients (MCCs), 25-dimensional band aperiodicities(BAPs) and F_0 in log-scale at a 5-ms step.

[1] http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html

TABLE I

Comparison between global and speaker-dependent (SD) normalisation with and without i-vectors in AVM training. MCD = Mel-Cepstral Distortions. BAP is the distortion for band aperiodiciaties. RMSE = Root Mean Squared Error. V/UV is the percentage of voicing/unvoicing decision error.

Methods	MCD (dB)	BAP (dB)	F_0 RMSE (Hz)	V/UV (%)
global (w/o i-vec)	3.254	4.375	47.152	14.842
global (with i-vec)	2.743	4.069	17.540	12.042
SD (w/o i-vec)	3.070	4.221	17.577	14.166
SD (with i-vec)	2.754	4.078	16.765	12.182

For the DNN AVM model, there were 6 feed-forward hidden layers, each of which had 1,536 hidden units. The hyper-parameters, such as learning rate and momentum, were tuned on the development set. The input features to the DNN included three parts: i-vector, gender code and speaker-independent linguistic features. The linguistic features were extracted by Festival [25], and they were converted to 501-dimensional binary and/or numerical features, similar to that in [8]. The output acoustic parameters included 60-dimensional MCCs, 25-dimensional BAPs and linearly interpolated F_0 in log-scale with their delta, delta-delta features, plused a voicing/unvoicing (V/UV) flag, totally 259 dimensions. The input features were normalised to a fixed range [0.01 0.99], and the output acoustic parameters were normalised by either speaker-dependent mean-variance normalisation (MVN) or speaker-independent global MVN. At the generation time, maximum likelihood parameter generation (MLPG) and spectral enhancement to MCCs were applied to improve the naturalness of synthesised speech. The speech waveforms were reconstructed by the STRAIGHT vocoder. In practice, Merlin [26] was used to train the AVM.

For i-vector extraction, WSJ0, WSJ0, WSJ-CAM and VCTK databases were used to train UBM, TVS and LDA models. All the databases were downsampled to 16 kHz, as i-vectors were used as the input to the DNN model, which is independent from the sampling rate for speech synthesis, and in practice it is much easier to obtain 16 kHz data. I-vectors were extracted from gender-dependent models. In the gender-dependent GMM-UBM training, we extracted 19-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) and log-energy with corresponding delta and delta-delta coefficients, and the size of window is 25ms with a 10ms shift. Voice activity detection (VAD) was performed to remove the silence frames. A set of 512 Gaussian components were used, and we calculated the sufficient statistics from UBM for every 10 sentences, which were used to extract one 400-dimensional i-vector. All the individual i-vectors of a target speaker were averaged to get a single i-vector to represent each speaker. In this study, the MSR identity toolbox [27] was used to extract i-vectors.

B. Objective evaluation

We first analysed the impact of speaker-dependent (SD) normalisation. Objective results with and without i-vectors are presented in Table I. It is observed that without i-vectors in the AVM training, SD normalisation achieves considerably lower distortions than global normalisation for all the objective

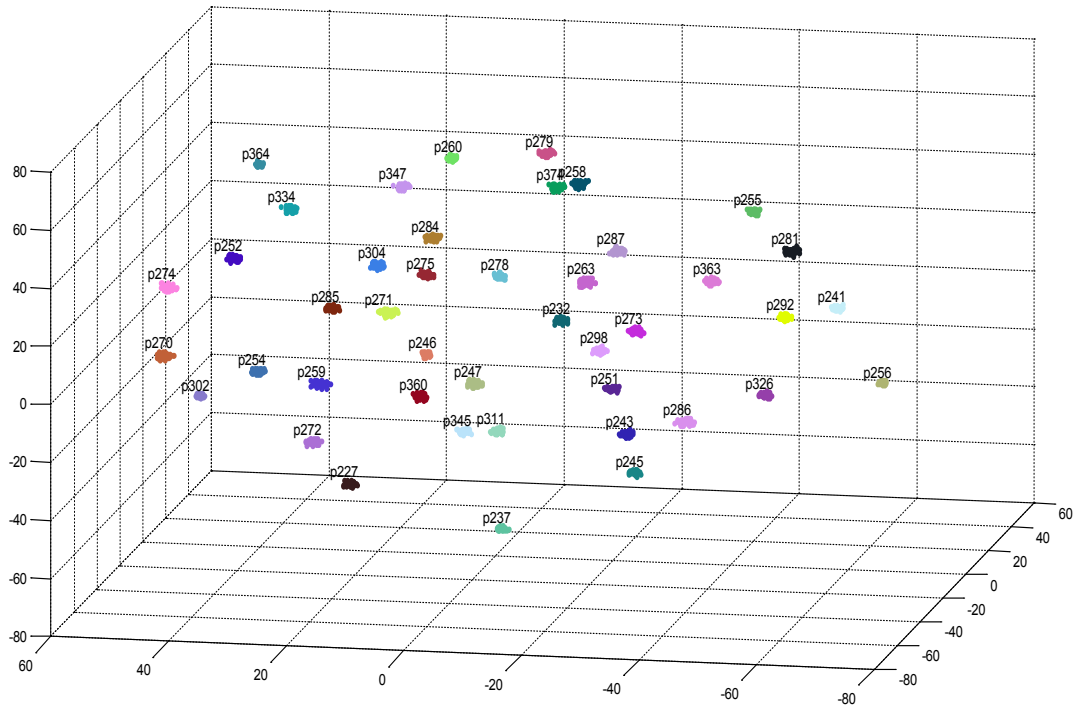

Fig. 3. The 3D distribution of 16-dimensional i-vectors. The t-SNE tool [28] is used for visualization. Each point in the space represents an average i-vector extracted from a group of sentences from the speaker. It is seen that i-vectors from the same speaker are generally in the same cluster, and there are clear discrimination across speakers.

measures, specially F_0, as global normalisation cannot tell speaker variations. However, when i-vectors are included in the AVM training, global and SD normalisation methods result in similar objective results. It might because i-vectors have considered speaker variations already.

We then analysed the impact of i-vector extractions. We first extracted 400-dimensional i-vectors and then applied LDA to obtain low-dimensional i-vectors. The dimensionality was varied from 2 to 128 to evaluate its effects on the AVM performance. To visually show the relations of i-vectors across speakers, we used t-SNE [28] to visualise 16-dimensional i-vectors from male speakers in the AVM training data, as presented in Fig. 3. Each data point within a speaker's "cloud" is an average i-vector extracted from a group of sentences from the speaker. It is seen that i-vectors from the same speaker are generally in the same "cloud" or cluster, and there are clear discrimination across speakers. The distance across i-vectors might reflect the speaker distance or similarity and the AVM with i-vectors might be able to learn these relationships.

Objective results with varied dimensions of i-vectors are presented in Table II. It is observed that even with a 2-dimensional i-vector, we achieved 4.7% relative MCD degradation and 7.1% relative V/UV error degradation, in comparison to that without i-vectors in the AVM training. As the dimension of i-vectors increases, there is a considerable drop in all distortion measures from 2-dimensional to 16-dimensional i-vectors, and the distortions converge when the dimension of i-vectors is 16 or 32. It suggests that 16-dimensional or 32-dimensional i-vectors are enough to capture speaker identity information for speech synthesis which generally uses clean speech. This is different from that in speaker verification which generally uses 200-dimensional i-vectors or even higher.

TABLE II
The impact of i-vector dimensions to the performance of AVMs.

Dimension	MCD (dB)	BAP (dB)	F_0 RMSE (Hz)	V/UV (%)
w/o i-vector	3.070	4.221	17.577	14.166
2-D	2.927	4.167	17.352	13.165
4-D	2.812	4.107	16.876	12.486
8-D	2.771	4.078	16.803	12.252
16-D	2.754	4.078	16.765	12.182
32-D	2.745	4.071	16.848	12.142
64-D	2.745	4.070	16.979	12.161
128-D	2.742	4.067	16.909	12.058

In general, objective results confirm the important of speaker-dependent normalisation and the use of i-vectors for AVM training.

C. Subjective evaluation

Although objective measures provide a good way to tune AVMs, they might not always correlate with human perception. To this end, we performed subjective evaluation in terms of naturalness and speaker similarity to assess the performance of AVMs.

AB preference tests were conducted for both naturalness and similarity. For the listening test, 20 sentences were randomly selected from the testing data, and 25 listeners participated in each test. In the naturalness test, two samples were presented

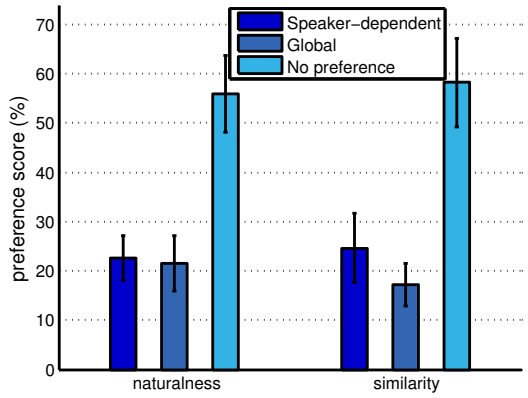

Fig. 4. AB preference results between speaker-dependent and speaker-independent (also called "global") normalisation with 95% confidence intervals. Note that 16-dimensional i-vectors were expanded as suggested by the objective results.

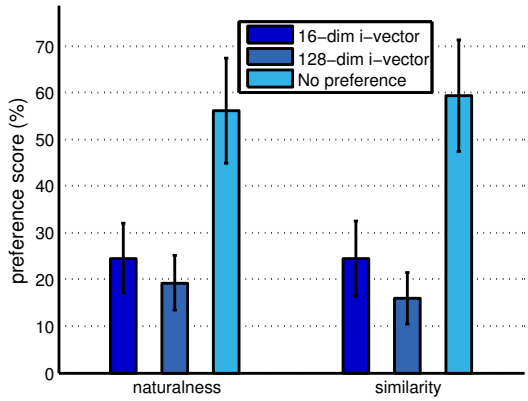

Fig. 5. AB preference results between AVMs that used 16-dimensional and 128-dimensional i-vectors with 95% confidence intervals.

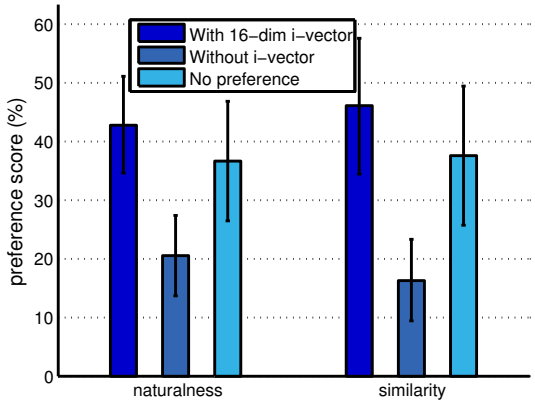

Fig. 6. AB preference results with and without i-vectors for AVM training. The error line are 95% confidence intervals.

to the listeners, and the listeners were asked to decided which one was more natural. If they could not hear the difference, they were guided to choose the "neutral" option. As for the similarity test, a reference sample from each target speaker's was first presented, and then two samples from two different AVM models were presented to the listeners. They were asked to choose the one which sounded more like the reference sample, or choose the "neutral" option if there was no difference.

We first analysed the impact of speaker normalisation to the performance of AVM. The preference results are presented in Figure 4. Reviewing Table I, the objective results can be considerably reduced by introducing i-vectors in AVM training. Hence, we used AVMs with 16-dimensional i-vector as input in this test. It is observed that even though the normalisation methods do not affect the naturalness, speaker-dependent normalisation achieves relatively higher similarity scores than the global normalisation which does not take speaker variations into account. The results suggest that speaker-dependent normalisation can improve the performance of AVM.

We then analysed the impact of the dimensionality of i-vectors to the performance of AVM. As suggested by the objective results, 16-dimensional i-vectors can achieve almost the same objective results as that by 128-dimensional i-vectors. Hence, we only performed subjective evaluation between AVMs that used 16-dimensional and 128-dimensional i-vectors, and employed speaker-dependent normalisation. The preference results are shown in Figure 5. It demonstrates that 16-dimensional i-vectors achieve slightly higher naturalness and similarity scores than that of 128-dimensional i-vectors, but the differences are not significant. We note that since the dimensionality of the linguistic features is only 501, 128-dimensional i-vectors might be slightly redundant and might degrade the performance of AVMs.

After that, we analysed the effectiveness of i-vectors to the performance improvement of AVMs. We compared AVMs without i-vectors and that with 16-dimensional i-vectors. The preference results are presented in Figure 6. It is observed that by introducing i-vectors in the AVM training, both naturalness and similarity are signficantly increased, and the subjective results are consistent with the objective results.

IV. Conclusions

In this work, we performed a systematic analysis of the training of multi-speaker average voice model for DNN-based speech synthesis. We have the following findings:

- Speaker-dependent normalisation on acoustic features achieves better performance than global normalisation both objectively and subjectively.
- I-vector is an effective technique to improve the performance of AVMs, in both naturalness and speaker similarity.
- Even though in speaker verification, i-vectors at a higher dimension is generally used, in our experiments, we found that 16-dimensional i-vectors can already capture speaker identity information for the speech synthesis task.

As i-vectors are used at the input level, it is flexible to combine with other speaker adaptive training (SAT) techniques. In

future work, we plan to investigate the combination of i-vectors with other techniques, such as learning hidden unit contributions (LHUC). We will also investigate the impact of selection of speakers for AVM training.

REFERENCES

[1] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[3] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7962–7966.

[4] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, "From hmms to dnns: where do the improvements come from?" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[5] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural network (dnn) for parametric tts synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3829–3833.

[6] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 10, pp. 2129–2139, 2013.

[7] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3844–3848.

[8] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4460–4464.

[9] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4455–4459.

[10] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Speaker adaptation for hmm-based speech synthesis system using mllr," in *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998.

[11] M. TAMURA, T. MASUKO, K. TOKUDA, and T. KOBAYASHI, "Speaker adaptation of pitch and spectrum for hmm-based speech synthesis," *IEICE transactions on information and systems*, vol. 85, no. 4, p. 793, 2002.

[12] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using hsmm-based speaker adaptation and adaptive training," *IEICE TRANSACTIONS on Information and Systems*, vol. 90, no. 2, pp. 533–543, 2007.

[13] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for hmm-based speech synthesis and a constrained smaplr adaptation algorithm," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 66–83, 2009.

[14] J. Yamagishi, T. Nose, H. Zen, Z.-H. Ling, T. Toda, K. Tokuda, S. King, and S. Renals, "Robust speaker-adaptive hmm-based text-to-speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 6, pp. 1208–1230, 2009.

[15] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for dnn-based tts synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4475–4479.

[16] ——, "Speaker and language factorization in dnn-based tts synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2016, pp. 5540–5544.

[17] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King, "A study of speaker adaptation for dnn-based speech synthesis," in *Proceedings interspeech*, 2015.

[18] O. Watts, Z. Wu, and S. King, "Sentence-level control vectors for deep neural network speech synthesis," in *Interspeech*, 2015.

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

[20] B. Scholkopft and K.-R. Mullert, "Fisher discriminant analysis with kernels," *Neural networks for signal processing IX*, vol. 1, no. 1, p. 1, 1999.

[21] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[22] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[23] C. R. Rao, "The utilization of multiple measurements in problems of biological classification," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 10, no. 2, pp. 159–203, 1948.

[24] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[25] A. Black, P. Taylor, R. Caley, R. Clark, K. Richmond, S. King, V. Strom, and H. Zen, "The festival speech synthesis system version 1.4.2," Software, Jun 2001. [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival/

[26] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," in *9th ISCA Speech Synthesis Workshop (SSW9)*, Sunnyvale, CA, USA, September 2016.

[27] S. O. Sadjadi, M. Slaney, and L. Heck, "Msr identity toolbox v1. 0: A matlab toolbox for speaker-recognition research," *Speech and Language Processing Technical Committee Newsletter*, 2013.

[28] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.