INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Multi-speaker Text-to-speech Synthesis Using Deep Gaussian Processes

Kentaro Mitsui, Tomoki Koriyama, Hiroshi Saruwatari

The University of Tokyo, Japan

[kentaro_mitsui, tomoki_koriyama, hiroshi_saruwatari]@ipc.i.u-tokyo.ac.jp

Abstract

Multi-speaker speech synthesis is a technique for modeling multiple speakers' voices with a single model. Although many approaches using deep neural networks (DNNs) have been proposed, DNNs are prone to overfitting when the amount of training data is limited. We propose a framework for multi-speaker speech synthesis using deep Gaussian processes (DGPs); a DGP is a deep architecture of Bayesian kernel regressions and thus robust to overfitting. In this framework, speaker information is fed to duration/acoustic models using speaker codes. We also examine the use of deep Gaussian process latent variable models (DGPLVMs). In this approach, the representation of each speaker is learned simultaneously with other model parameters, and therefore the similarity or dissimilarity of speakers is considered efficiently. We experimentally evaluated two situations to investigate the effectiveness of the proposed methods. In one situation, the amount of data from each speaker is balanced (speaker-balanced), and in the other, the data from certain speakers are limited (speaker-imbalanced). Subjective and objective evaluation results showed that both the DGP and DGPLVM synthesize multi-speaker speech more effective than a DNN in the speaker-balanced situation. We also found that the DGPLVM outperforms the DGP significantly in the speaker-imbalanced situation.

Index Terms: deep Gaussian process, statistical speech synthesis, multi-speaker modeling, latent variable model

1. Introduction

With the development of machine learning in recent years, text-to-speech (TTS) synthesis has a greater variety of applications than ever before. Recent studies have shown that multi-speaker modeling, a technique that models the voices of multiple speakers with a single model, is effective for synthesizing multiple speakers' voices. Multi-speaker modeling can benefit from multi-task learning [1], which means this technique requires less training data to achieve high-quality speech synthesis.

Statistical parametric speech synthesis (SPSS) is one possible method for multi-speaker speech synthesis. Hidden Markov model (HMM)-based methods such as the average voice model [2] were widely used until the emergence of deep neural network (DNN)-based speech synthesis [3]. For multi-speaker modeling in DNN-based speech synthesis, Fan et al. introduced a shared hidden-layer structure, which shares the hidden-layer parameters of a DNN among different speakers, and reported that this structure improved the quality of synthetic speech relative to the speaker-dependent DNNs [4]. Another successful method for multi-speaker modeling is based on speaker codes, which are the representation of speakers in a form such as a one-hot vector or randomly assigned vector. Luong et al. investigated the optimal form for speaker codes [5]. The method proposed by Hojo et al. outperformed the shared hidden-layer structure by feeding one-hot speaker codes to the hidden layers of a DNN [6]. In addition, the method using speaker representation has recently been applied to end-to-end speech synthesis frameworks, and the method has achieved high speech quality [7, 8]. However, most of the DNN-based methods only consider data fitting while training, and thus overfitting often becomes a problem.

In this paper, we focus on the SPSS framework using deep Gaussian processes (DGPs) [9]. In this framework, the relationship between linguistic features and phoneme durations or acoustic features are modeled using DGPs [10]. A DGP is a deep architecture of Bayesian kernel regressions, so it can express complicated non-linear transformation with a small number of hyperparameters. Both data fitting and model complexity are considered in the training of a DGP, which makes the model less vulnerable to overfitting than a DNN. Previous work has shown that DGP-based TTS performs better than a feed-forward DNN for single-speaker modeling [9]. However, the DGP's effectiveness for multi-speaker TTS is yet to be verified.

Therefore, we propose multi-speaker TTS based on DGP. We introduce two methods: one method using a general DGP and feeding one-hot speaker codes to its hidden layers, similarly to the DNN-based method [6]; and the other based on learning latent representation of speakers using deep Gaussian process latent variable models (DGPLVMs) [10]. The second method incorporates a GPLVM [11], a Bayesian generative model shown to be effective in prosody modeling [12], into the general DGP to obtain speaker representation. The difference between DGPs and DGPLVMs is the representation of speaker similarity used for kernel regression. A DGPLVM can explicitly express the similarity using the latent representation whereas the speaker codes used in a general DGP cannot. In addition, the use of DGPLVM enables an analysis of speakers in the latent space.

In the experimental evaluations, we investigate the performance of our methods in speaker-balanced and speaker-imbalanced situations. In the speaker-imbalanced situation, we first selected target speakers and used limited data for those speakers while training. We conducted objective and subjective evaluations in both situations to evaluate the effectiveness of the proposed methods. Experimental results showed that in the speaker-balanced situation, both proposed methods improved the speech quality relative to the DNN-based method; and in the speaker-imbalanced situation where only five training utterances were used for the target speakers, the DGPLVM improved naturalness and speaker similarity of synthetic speech.

2. Conventional methods

2.1. DNN-based multi-speaker TTS using speaker codes

We give an overview of DNN-based multi-speaker TTS using speaker codes [6], a simple yet highly effective method within the SPSS framework. Single-speaker models use only contextual factors as the inputs of duration/acoustic models, but this method uses speaker codes as auxiliary inputs to model speaker variation. Here, speaker code $\mathcal{S}$ is a one-hot vector represen-

http://dx.doi.org/10.21437/Interspeech.2020-3167

tation of the speaker ID. We apply linear transformation to this vector and add the result to hidden layers:

$$\mathbf{h}^{\ell+1} = \varphi(\mathbf{W}^{\ell+1}(\mathbf{h}^{\ell} + \mathbf{W}^{\ell}_{\mathcal{S}}\mathcal{S}) + \mathbf{b}^{\ell+1}) \quad (1)$$

where $\varphi(\cdot)$ is an activation function, $\mathbf{h}^{\ell}$ is the component of the ℓ-th hidden layer, $\mathbf{W}^{\ell}$ and $\mathbf{W}^{\ell}_{\mathcal{S}}$ are the connection weight of the hidden layers and speaker codes, respectively, and $\mathbf{b}^{\ell}$ is the bias. Training is conducted by minimizing the mean squared error between the natural and generated acoustic features.

2.2. DGP-based speech synthesis

In the DGP-based speech synthesis framework [9], a DGP model takes linguistic features as inputs and predicts phoneme durations or acoustic features. A DGP is a model defined as a cascade of Gaussian process regressions (GPRs).

GPRs model the relation between input $\mathbf{x}$ and output y as:

$$y = f(\mathbf{x}) + \epsilon \quad (2)$$

$$f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (3)$$

and infer the posterior distribution $p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y})$ against the new input $\mathbf{x}_*$ by using the training data $(\mathbf{X}, \mathbf{y})$. Here ϵ is random noise, and $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ are mean and kernel functions, respectively. We consider multiple GPRs when the output is multidimensional.

Although a single GPR can represent complicated nonlinear functions, its expressiveness is limited by the kernel function. A DGP overcomes this limitation by stacking multiple GPRs; this method is based on the assumption that the overall function f can be decomposed into multiple functions in the following manner:

$$f = f^{L+1} \circ f^{L} \circ \cdots \circ f^{1} \quad (4)$$

where L is the number of hidden layers, and each function f^{ℓ} is a sample of a Gaussian process. An approximation technique called doubly stochastic variational inference [13] is used in this framework, so training is conducted by maximizing the evidence lower bound (ELBO) of log marginal likelihood:

$$\log p(\mathbf{Y}) \geq \frac{1}{N_s}\sum_{j=1}^{N_s}\sum_{i=1}^{N}\left\{\sum_{d=1}^{D_{L+1}} \mathbb{E}_{q(\mathrm{f}^d_{i,j})}\left[\log p\left(y^d_i|\mathrm{f}^d_{i,j}\right)\right] - \frac{N_s}{N}\sum_{\ell=1}^{L+1}\mathrm{KL}\left[q(\mathbf{U}^{\ell})\|p(\mathbf{U}^{\ell}|\mathbf{Z}^{\ell})\right]\right\} \triangleq \mathcal{L}_1 \quad (5)$$

where N, N_s are the number of training data and Monte Carlo samples, respectively, and D_{ℓ} is the dimensionality of the output of the ℓ-th GPR. y^d_i is the d-th dimension of the i-th observed output $\mathbf{y}_i$, and $\mathrm{f}^d_{i,j}$ represents the corresponding latent function predicted from the j-th sample point. $\mathbf{Z}^{\ell}$ and $\mathbf{U}^{\ell}$ denote the inducing inputs and outputs, respectively, which are sparse representations of input and output data. While $\mathbf{Z}^{\ell}$ is a model parameter by itself, $\mathbf{U}^{\ell}$ itself is not a parameter but a random variable, in which we impose $q(\mathbf{U}^{\ell}) = \Pi_{d=1}^{D_{\ell}} q(\mathbf{u}^{\ell,d}) = \Pi_{d=1}^{D_{\ell}} \mathcal{N}(\mathbf{u}^{\ell,d}; \mathbf{m}^{\ell,d}, \mathbf{S}^{\ell,d})$ and regard mean $\mathbf{m}^{\ell,d}$ and variance $\mathbf{S}^{\ell,d}$ as model parameters for each layer ℓ and dimension d.

3. DGP-based multi-speaker TTS using speaker codes

We introduce the model architecture shown in Fig. 1 to apply the DGP-based speech synthesis framework [9] to multi-speaker TTS. In this architecture, speaker IDs are represented using one-hot speaker codes in a manner similar to the DNN-based method described in Section 2.1. We apply a single-layer GPR to these speaker codes before feeding them to the hidden

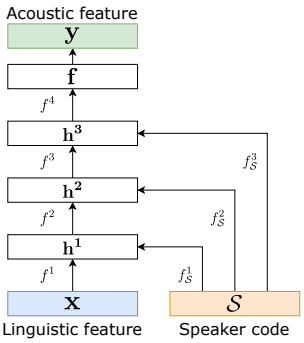

Figure 1: *Architecture of DGP-based acoustic model for multi-speaker TTS with three hidden layers.*

layers. Therefore, the values of the ℓ-th hidden layer $\mathbf{h}^{\ell}$ can be written as:

$$\mathbf{h}^{\ell} = f^{\ell}(\mathbf{h}^{\ell-1}) + f^{\ell}_{\mathcal{S}}(\mathcal{S}) \quad (6)$$

where $\mathcal{S}$ denotes the speaker code, f^{ℓ} is the ℓ-th GPR in the DGP (hereinafter called the hidden GP), and $f^{\ell}_{\mathcal{S}}$ is the ℓ-th GPR to transform speaker codes (hereinafter called the speaker GP). Speaker GPs have inducing inputs $\mathbf{Z}^{\ell}_{\mathcal{S}}$ and corresponding outputs $\mathbf{U}^{\ell}_{\mathcal{S}}$ as well as hidden GPs, so we must optimize these parameters jointly with other model parameters. This can be done by maximizing the new ELBO:

$$\mathcal{L}_2 = \mathcal{L}_1 - \sum_{\ell=1}^{L}\mathrm{KL}\left[q(\mathbf{U}^{\ell}_{\mathcal{S}})\|p(\mathbf{U}^{\ell}_{\mathcal{S}}|\mathbf{Z}^{\ell}_{\mathcal{S}})\right]. \quad (7)$$

4. DGPLVM for multi-speaker TTS

In this section, we propose another approach for multi-speaker TTS using a DGPLVM [10]. The DGP-based approach illustrated in the previous section is straightforward, but because one-hot speaker codes are orthogonal to each other between speakers, we cannot fully make use of the similarity or dissimilarity of speakers. In the DGPLVM-based approach, we aim to utilize speaker similarity for multi-speaker TTS.

We express K speakers by using latent variable $\mathbf{R} = (\mathbf{r}_1, ..., \mathbf{r}_K)$, and use the latent variable as the input of function f^{ℓ} as follows:

$$f^{\ell} \sim \mathcal{GP}(m(\mathbf{x}, \mathbf{r}_k), k([\mathbf{x}^{\top}, \mathbf{r}_k^{\top}]^{\top}, [\mathbf{x}'^{\top}, \mathbf{r}_{k'}^{\top}]^{\top})). \quad (8)$$

From Bayes' theorem, the distribution of $\mathbf{r}_k$ conditioned on input $\mathbf{x}$ and output $\mathbf{y}$ can be written as:

$$p(\mathbf{r}_k|\mathbf{x}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \mathbf{r}_k)p(\mathbf{r}_k). \quad (9)$$

When we consider acoustic modeling, the left-hand side of (9) is conditioned not only on linguistic feature $\mathbf{x}$ but also on acoustic feature $\mathbf{y}$. Since the kernel function uses latent variable $\mathbf{r}_k$ as input, $\mathbf{r}_k$ is learned to express the similarity of acoustic features among different speakers. We assign a prior given by the standard normal distribution to $\mathbf{r}_k$:

$$p(\mathbf{r}_k) = \mathcal{N}(\mathbf{r}_k; \mathbf{0}, \mathbf{I}). \quad (10)$$

Also, we consider the latent variable for k-th speaker $\mathbf{r}_k$ to have a variational distribution

$$q(\mathbf{r}_k) = \mathcal{N}(\mathbf{r}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (11)$$

where $\boldsymbol{\mu}_k$ is a mean vector and $\boldsymbol{\Sigma}_k$ is a diagonal covariance matrix. This latent variable is fed to an arbitrary hidden layer of the DGP. In this case the ELBO of $\log \int p(\mathbf{Y}|\mathbf{R})p(\mathbf{R})d\mathbf{R}$ is written as:

$$\mathcal{L}_3 = \mathcal{L}_1 - \sum_{k=1}^{K}\mathrm{KL}\left[q(\mathbf{r}_k)\|p(\mathbf{r}_k)\right]. \quad (12)$$

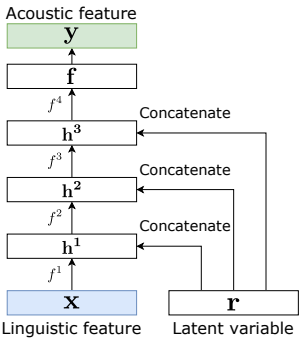

Figure 2: *Architecture of DGPLVM-based acoustic model for multi-speaker TTS with three hidden layers.*

5. Experiments

5.1. Experimental conditions

We used JVS corpus [14], which is comprised of speech data from 100 Japanese speakers, 49 males and 51 females. Speech waveforms were downsampled to 16 kHz. This corpus contained 100 parallel utterances (parallel100) and 30 non-parallel utterances (nonpara30) from each speaker. For the speaker-balanced situation, the training set consisted of all the non-parallel and 85 of the 100 parallel utterances from each speaker, and the test set consisted of the remaining 15 parallel utterances from each speaker. For the speaker-imbalanced situation, four speakers, two males and two females, were selected as target speakers; for these speakers, only five non-parallel utterances were used in training. To avoid low speech quality for the target speakers, we used an oversampling technique [15] and sampled each utterance of each target speaker 20 times. The target speakers were selected on the basis of subjective speaker similarity [16]. Specifically, we defined the speaker who had the largest median of similarity score between other speakers, in other words who had many similar speakers, as *male/female similar* (`MS/FS`), and the opposite ones as *male/female dissimilar* (`MD/FD`). The test set consisted of 15 parallel utterances from the four target speakers.

The input linguistic features of the duration model were 531-dimensional vectors containing contextual factors such as phoneme, accent, and part of speech, which were automatically estimated from texts using Open JTalk [17]. We added a four-dimensional frame index to these linguistic features and used them as the input of the acoustic model. The output of the duration model was a one-dimensional phoneme duration. The acoustic features, i.e. the output of the acoustic model, were 187-dimensional vectors comprised of 0–59th mel-cepstrum, log f_{o}, coded aperiodicity and their Δ, Δ^2, followed by voiced/unvoiced flags. These acoustic features were extracted every 5 ms using WORLD [18] (D4C edition [19]). We normalized input features to range $[0.01, 0.99]$ and output features to zero-mean and unit variance.

The DGP duration model had 2 hidden layers, with the dimensionality of each layer set to 32. The acoustic model had 5 hidden layers, and the dimensionality of each layer was 128. The number of inducing points was set to 1024 for hidden GPs and 8 for speaker GPs. We used ArcCos kernel [20] as a kernel function of GPs. The inducing inputs of each GP were initialized randomly with the standard normal distribution. The variational distributions of inducing outputs $q(\mathbf{u}^{\ell,d})$ of all GPs except the last hidden GP f^{L+1} were initialized with a Gaussian distribution with zero mean and variance 10^{-6}, while that of f^{L+1} had unit variance.

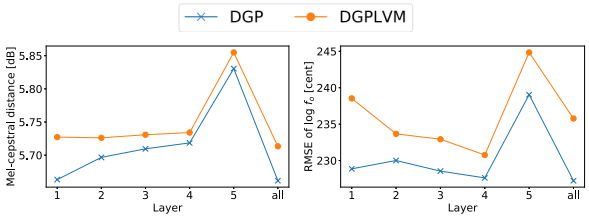

Figure 3: *Objective evaluation results for DGP and DGPLVM with different layers to feed speaker information.*

The DGPLVM had similar settings to the DGP model. However, it does not have speaker GPs and thus the total number of model parameters was reduced. The variational distributions of latent variables $q(\mathbf{r}_k)$ were initialized randomly with Gaussian distribution with zero mean and variance 10^{-4}.

We trained the models by mini-batch optimization with the batch size set to 1024, using Adam [21] whose learning rate was 0.01. For the conventional DNN model, we followed the previous work [6] and set the numbers of hidden layers to 2 and 5 for duration and acoustic models, respectively, the number of hidden units to 1024, and the learning rate of Adam to 10^{-4}. Training was conducted up to 50 epochs for the DGP/DGPLVM and 100 epochs for the DNN.

5.2. Objective evaluation

We compared the quality of synthetic speech in terms of distortions between the original and synthetic speech parameters. As evaluation metrics, we used the root mean squared error (RMSE) of phoneme durations (DUR) for duration models, and mel-cepstral distance (MCD) and RMSE of log f_{o} (F0) for acoustic models.

We first focused on the speaker-balanced situation and investigated the effect of model architecture on the performance of acoustic modeling. For the DGP, we fed the speaker code $\mathcal{S}$ to a certain layer (the first, second, third, fourth, or fifth layer) or all hidden layers of the acoustic model. In the same way, for the DGPLVM, we fed the latent speaker variable $\mathbf{r}_k$ to different layers. Here the dimensionality of $\mathbf{r}_k$ was set to three. The results are shown in Fig. 3. Although feeding speaker information only to the last hidden layer increased the acoustic distortion, the differences among other settings were relatively small. In the following experiments, we adopted the *all* settings for both the DGP and DGPLVM.

Next, we investigated the performance of the DGPLVM with different dimensionality of $\mathbf{r}_k$. We set the dimensionality of $\mathbf{r}_k$ to 2, 3, 16, and 64. The results are shown in Table 1. While higher dimensionality led to smaller distortions in the speaker-balanced situation, the results in the speaker-imbalanced situation were the opposite; lower dimensionality led to better results, and a dimensionality of three was optimal. This is possibly because latent speaker space becomes dense with low-dimensional speaker representation, and voice models of similar speakers are efficiently accounted for when synthesizing the target speaker's voice. We set the dimensionality of $\mathbf{r}_k$ to 64 for the speaker-balanced situation and 3 for the speaker-imbalanced situation in the following experiments.

Finally, we compared the performance of the conventional DNN, proposed DGP, and DGPLVM. In the speaker-balanced situation, all models yielded similar MCD, while the proposed DGP/DGPLVM showed better F0 and DUR than the DNN. In

Table 1: *Objective evaluation results for DGPLVM with different dimensionality of latent speaker variable* $\mathbf{r}_k$*. MCD: mel-cepstral distance [dB], F0: RMSE of log* f_0 *[cent].*

	Speaker-balanced		Speaker-imbalanced	
Dimensionality	MCD	F0	MCD	F0
2	5.72	235	6.24	280
3	5.71	236	**6.15**	**264**
16	**5.65**	233	6.28	285
64	**5.65**	**228**	6.31	282

Table 2: *Comparison of DNN, DGP and DGPLVM in terms of MCD: mel-cepstral distance [dB], F0: RMSE of log* f_0 *[cent], and DUR: RMSE of phoneme duration [ms].*

	Speaker-balanced			Speaker-imbalanced		
Method	MCD	F0	DUR	MCD	F0	DUR
DNN	5.66	239	25.6	**5.96**	271	28.0
DGP	5.66	**227**	25.4	6.29	280	27.7
DGPLVM	**5.65**	228	**24.9**	6.15	**264**	**27.6**

the speaker-imbalanced situation, DNN was the best in terms of MCD and DGPLVM was the best in terms of F0 and DUR.

5.3. Subjective evaluation

We conducted listening tests to subjectively evaluate the speech quality in terms of naturalness and speaker similarity[1]. The naturalness of synthetic speech was evaluated by preference A/B test, and speaker similarity was evaluated by XAB test. We compared two pairs: DNN–DGP and DGP–DGPLVM in the speaker-balanced/imbalanced situations. Thirty crowdsourced listeners participated in each of the evaluations, and each listener evaluated ten speech samples. The original speech of the target speaker was used as the reference X in the XAB tests.

The results are shown in Figs. 4 and 5. In the speaker-balanced situation, the scores of both naturalness and speaker similarity were higher for all speakers for the DGP than for the DNN. Although both scores of `FS` were lower in the DGPLVM than in the DGP due to duration errors, the scores of the remaining three speakers were comparable in DGP–DGPLVM. Collating these results with those of the objective evaluation, f_0 seems to have the greatest effect on naturalness and speaker similarity.

In the speaker-imbalanced situation, there was no significant difference between the DNN and DGP in total, though we observed larger acoustic feature distortions for the DGP in the objective evaluation. The naturalness of the DGPLVM for `MS` and `FS` were significantly higher than those of the DGP. In addition, the speaker similarity of those speakers were slightly higher than those of the other speakers in the DGPLVM. From these results, we infer that the DGPLVM can beneficially utilize similar speakers using the learned latent speaker representation.

5.4. Latent speaker representation learned by DGPLVM

The latent speaker representation after training the DGPLVM is shown in Fig. 6. Here, the dimensionality of $\mathbf{r}_k$ is set to two for ease of visualization. We found that male and female speakers were clearly separated, *similar* speakers (`MS`: 022 and `FS`: 063) were embedded inside of the cluster while *dissimilar* speakers (`MD`: 006 and `FD`: 010) were embedded outside, and speakers embedded closely in the speaker-balanced situation were also closely embedded in the speaker-imbalanced situation. These results indicate that the learned latent speaker representation expresses the similarity or dissimilarity of speakers as expected.

[1]Synthetic speech samples are available at `https://kentaro321.github.io/demo_DGP_MS_TTS/`.

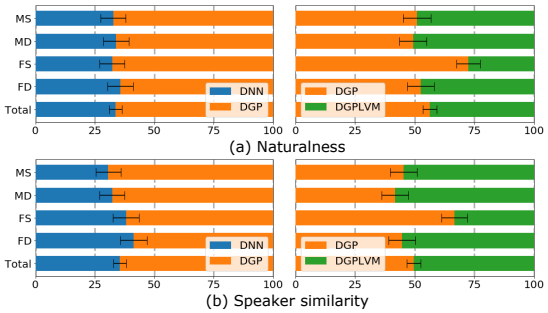

Figure 4: *Subjective evaluation results with 95% confidence intervals in speaker-balanced situation.*

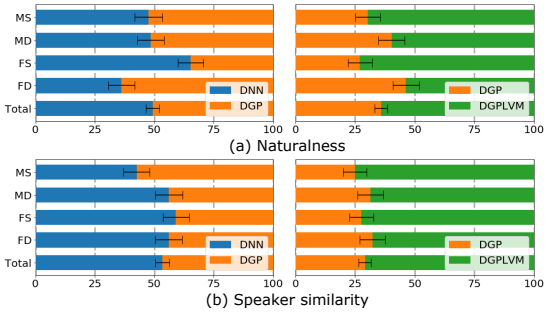

Figure 5: *Subjective evaluation results with 95% confidence intervals in speaker-imbalanced situation.*

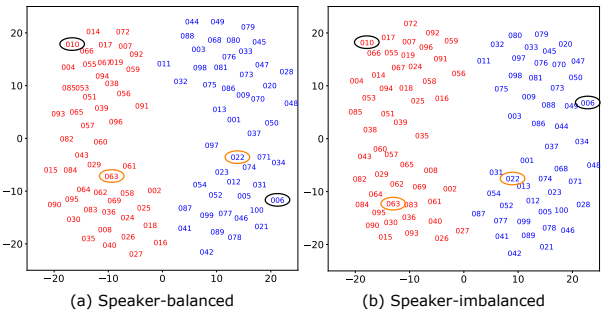

Figure 6: *Latent speaker representation learned by DGPLVM in (a) speaker-balanced situation and (b) speaker-imbalanced situation. Red and blue numbers indicate female and male speakers, respectively. Orange and black circles indicate the* similar *and* dissimilar *speakers, respectively.*

6. Conclusions

We have proposed multi-speaker TTS based on the DGP. We found that with one-hot speaker codes, the use of the DGP can improve naturalness and speaker similarity of multi-speaker speech relative to the DNN. We also introduced the DGPLVM-based multi-speaker TTS framework, in which speaker representation is treated as a latent variable and jointly learned with other model parameters. The experimental results showed that the DGPLVM-based approach is especially effective when the amount of training data from a certain speaker is highly limited. For future work, we will compare our DGPLVM-based method with other latent-space-based methods such as variational autoencoder [22]. We also plan to compare the performance of the proposed methods with recent end-to-end approaches.

7. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP19K20292.

8. References

[1] S. Ruder, "An overview of multi-task learning in deep neural networks," in *arXiv preprint arXiv:1706.05098*, 2017.

[2] 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.

[3] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, May 2013, pp. 7962–7966.

[4] Y. Fan, Y. Qian, F. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *Proc. ICASSP*, Brisbane, Australia, Apr. 2015, pp. 4475–4479.

[5] H.-T. Luong, S. Takaki, G. Henter, and J. Yamagishi, "Adapting and controlling DNN-based speech synthesis using input codes," in *Proc. ICASSP*, New Orleans, U.S.A., Mar. 2017, pp. 4905–4909.

[6] N. Hojo, Y. Ijima, and H. Mizuno, "DNN-based speech synthesis using speaker codes," *IEICE Transactions on Information and Systems*, vol. 101, no. 2, pp. 462–472, 2018.

[7] W. Ping, K. Peng, A. Gibiansky, S. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, Vancouver, Canada, May 2018.

[8] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. NIPS*, Montreal, Canada, Dec. 2018, pp. 4480–4490.

[9] T. Koriyama and T. Kobayashi, "Statistical parametric speech synthesis using deep Gaussian processes," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 5, pp. 948–959, 2019.

[10] A. Damianou and N. Lawrence, "Deep Gaussian processes," in *Proc. AISTATS*, Scottsdale, U.S.A., Apr. 2013, pp. 207–215.

[11] M. Titsias and N. Lawrence, "Bayesian Gaussian process latent variable model," in *Proc. AISTATS*, Sardinia, Italy, May 2010, pp. 844–851.

[12] T. Koriyama and T. Kobayashi, "Semi-supervised prosody modeling using deep Gaussian process latent variable model," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 4450–4454.

[13] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep Gaussian processes," in *Proc. NIPS*, California, U.S.A., Dec. 2017, pp. 4588–4599.

[14] S. Takamichi, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "JVS corpus: free Japanese multi-speaker voice corpus," in *arXiv preprint arXiv:1908.06248*, 2019.

[15] H.-T. Luong, X. Wang, J. Yamagishi, and N. Nishizawa, "Training multi-speaker neural text-to-speech systems using speaker-imbalanced speech corpora," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 1303–1307.

[16] Y. Saito, S. Takamichi, and H. Saruwatari, "DNN-based speaker embedding using subjective inter-speaker similarity for multi-speaker modeling in speech synthesis," in *Proc. 10th ISCA Speech Synthesis Workshop*, Vienna, Austria, Sep. 2019, pp. 51–56.

[17] Open JTalk, http://open-jtalk.sourceforge.net/.

[18] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[19] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.

[20] Y. Cho and L. Saul, "Kernel methods for deep learning," in *Proc. NIPS*, Vancouver, Canada, Dec. 2009, pp. 342–350.

[21] D. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, U.S.A, May 2015.

[22] D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, Banff, Canada, Apr. 2014.