10th ISCA Speech Synthesis Workshop
20-22 September 2019, Vienna, Austria

Sparse Approximation of Gram Matrices for GMMN-based Speech Synthesis

Tomoki Koriyama[1], *Shinnosuke Takamichi*[1], *Takao Kobayashi*[2]

[1]The University of Tokyo, Japan
[2]Tokyo Institute of Technology, Japan
[1]{tomoki_koriyama,shinnosuke_takamichi}@ipc.i.u-tokyo.ac.jp,
[2]takao.kobayashi@ip.titech.ac.jp

Abstract

This paper discusses a training method of speech synthesis framework using generative moment matching network (GMMN). GMMN is a deep generative model optimized by minimizing conditional maximum mean discrepancy (CMMD), and the GMMN-based speech synthesis system models the distribution of acoustic features. Although CMMD is computationally infeasible for a large amount of data, the reduction methods of computation complexity were not examined in the previous study. In this paper, we propose an approximation method based on random Fourier features (RFFs) and minibatch selection technique using K-means clustering. Experimental evaluations show that the proposed method outperformed the conventional one in the perception of inter-utterance variation.
Index Terms: Generative moment matching network, statistical speech synthesis, neural network, maximum mean discrepancy, kernel method

1. Introduction

Speech synthesis frameworks using deep neural networks (DNN) have been widely studied in recent years, and some studies report that synthetic speech and natural recording were perceptually indistinguishable from each other [1, 2]. However, since DNN-based speech synthesis generally models one-to-one mapping from contexts to acoustic features, synthetic speech does not change as long as the same sentence is input. This is different from human speech production, in which human utterances have variation even if the sentence is the same. Therefore, it is desirable to model such variation to construct more human-like speech synthesis system.

In this context, we have proposed a random sampling method of acoustic features based on generative moment matching network (GMMN) [3]. GMMN [4, 5] is a neural-network-based generative model which predicts the variation of output features, by minimizing the distance of distributions between training data and generated samples from the neural network. We can directly obtain the random sample of the predicted distribution by using random values of simple prior as the input variable of neural network. The GMMN-based speech synthesis models the distribution of speech parameters, such as mel-cepstrum and F0, instead of point estimation, and it achieves random sampling from the same context. Moreover, we have shown the effectiveness of GMMN in various applications such as speaker verification [6] and singing voice synthesis [7].

An issue of GMMN is its computational complexity in training. Specifically, a conditional maximum mean discrepancy (CMMD), a training criterion of GMMN, requires $\mathcal{O}(N^2)$ memory storage for Gram matrices and $\mathcal{O}(N^3)$ computation complexity for matrix inversion, where N is the number of training frames. Hence, it is unrealistic to directly incorporate CMMD into GMMN-based speech synthesis. In the previous study, we divide training data set into randomly selected minibatches and calculated CMMD for each minibatch. The CMMDs of respective minibatches are not equivalent to that of whole training data, and this is equivalent to block diagonal approximation of Gram matrices. However this is just one approximation method that is available for GMMN-based speech synthesis.

Therefore, we investigate the effect of the approximation method of CMMD for GMMN-based speech synthesis. In addition to the block diagonal approximation that utilizes local information of minibatch, we introduce a low-rank approximation based on random Fourier features (RFFs) [8] that uses global characteristics. Furthermore, we propose a method of minibatch selection based on K-means clustering instead of random selection so that we can utilize the information of similar frames. We perform a subjective evaluation to examine the effectiveness of approximation and minibatch selection methods. As a measure of subjective evaluation, we use not only the naturalness of synthetic speech but also whether the difference is perceived or not in randomly generated two speech samples of the same sentence.

2. Distance of distributions based on maximum mean discrepancy

2.1. Maximum mean discrepancy

In this paper, we first describe maximum mean discrepancy (MMD)[9], which measures the distance of two distributions. MMD can be used as the criterion for the training of a generative model that yield a distribution. We summarize the notations used in this paper in Table 1. Let P and $\tilde{P}$ be distributions on a space $\mathcal{Y}$, and the MMD of the distributions is defined by

$$\mathrm{MMD} = \sup_{\|f\|\leq 1, f\in\mathcal{F}} \left| \mathbb{E}_{Y\sim P}[f(Y)] - \mathbb{E}_{\tilde{Y}\sim\tilde{P}}[f(\tilde{Y})] \right| \quad (1)$$

where $\mathcal{F}$ is a reproducing kernel Hilbert space (RKHS) defined by a positive definite kernel function $k : \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$. Equation (1) expresses the distance of expectation values mapped by witness function $f : \mathcal{Y}\to\mathbb{R}$. We obtain the MMD by choosing the function that maximizes the distance.

We define an implicit feature mapping by $\phi(\mathbf{y}) = k(\mathbf{y},\cdot)$ and denote the kernel mean embeddings as $\mu_Y \triangleq \mathbb{E}_{Y\sim P}[k(Y,\cdot)] = \mathbb{E}_{Y\sim P}[\phi(Y)]$. The term of right-hand side in (1) is represented as follows:

$$\begin{aligned}\mathbb{E}_{Y\sim P}[f(Y)] &= \mathbb{E}_{Y\sim P}\langle f, \phi(Y)\rangle_{\mathcal{F}} \\ &= \langle f, \mathbb{E}_{Y\sim P}[\phi(Y)]\rangle_{\mathcal{F}} = \langle f, \mu_Y\rangle_{\mathcal{F}} \end{aligned} \quad (2)$$

where $\langle\cdot,\cdot\rangle_{\mathcal{F}}$ represents the inner product on RKHS $\mathcal{F}$. There-

10.21437/SSW.2019-27

Table 1: *Table of notation.*

random variable	X	Y
domain	$\mathcal{X}$	$\mathcal{Y}$
observation	$\mathbf{x}$	$\mathbf{y}$
kernel	$h(\mathbf{x}, \mathbf{x}')$	$k(\mathbf{y}, \mathbf{y}')$
Gram matrix	$\mathbf{H}$	$\mathbf{K}$
feature map	$\psi(\mathbf{x}) = h(\mathbf{x}, \cdot)$	$\phi(\mathbf{y}) = k(\mathbf{y}, \cdot)$
feature matrix	$\mathbf{\Upsilon}$	$\mathbf{\Phi}$
RKHS	$\mathcal{G}$	$\mathcal{F}$

fore, the MMD in (1) becomes

$$\begin{aligned}\mathrm{MMD} &= \sup_{\|f\| \leq 1, f \in \mathcal{F}} (\langle f, \mu_Y \rangle_{\mathcal{F}} - \langle f, \mu_{\tilde{Y}} \rangle_{\mathcal{F}}) \\ &= \sup_{\|f\| \leq 1, f \in \mathcal{F}} \langle f, \mu_Y - \mu_{\tilde{Y}} \rangle_{\mathcal{F}}. \end{aligned} \tag{3}$$

Since the witness function that maximize the inner product in (3) is given by

$$f = \frac{\mu_Y - \mu_{\tilde{Y}}}{\|\mu_Y - \mu_{\tilde{Y}}\|_{\mathcal{F}}} \tag{4}$$

the MMD can be expressed by the following equation:

$$\begin{aligned}\mathrm{MMD}^2 &= \|\mu_Y - \mu_{\tilde{Y}}\|_{\mathcal{F}}^2 \\ &= \langle \mu_Y, \mu_Y \rangle_{\mathcal{F}} + \langle \mu_{\tilde{Y}}, \mu_{\tilde{Y}} \rangle_{\mathcal{F}} - 2\langle \mu_Y, \mu_{\tilde{Y}} \rangle_{\mathcal{F}}. \end{aligned} \tag{5}$$

When the samples of distribution P are $\mathcal{D}_{\mathbf{Y}} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ and the mapped values are reresented by a matrix form as $\mathbf{\Phi} = [\phi(\mathbf{y}_1), \ldots, \phi(\mathbf{y}_N)]^\top$, the sample mean $\hat{\mu}_Y$ is given by

$$\hat{\mu}_Y = \frac{1}{N} \mathbf{\Phi}_{\mathbf{Y}}^\top \mathbf{1}_{N \times 1}. \tag{6}$$

where $\mathbf{1}_{A \times B}$ is a $(A \times B)$ matrix of ones. The inner product of sample mean is represented by

$$\begin{aligned}\langle \mu_Y, \mu_{\tilde{Y}} \rangle_{\mathcal{F}} &= \frac{1}{N\tilde{N}} \mathrm{Tr}\left[(\mathbf{\Phi}_{\mathbf{Y}}^\top \mathbf{1}_{N \times 1})^\top (\mathbf{\Phi}_{\tilde{\mathbf{Y}}}^\top \mathbf{1}_{\tilde{N} \times 1}) \right] \\ &= \frac{1}{N\tilde{N}} \mathrm{Tr}\left[\mathbf{\Phi}_{\mathbf{Y}} \mathbf{\Phi}_{\tilde{\mathbf{Y}}}^\top \mathbf{1}_{\tilde{N} \times N} \right] \\ &= \frac{1}{N\tilde{N}} \mathrm{Tr}\left[\mathbf{K}_{\mathbf{Y}\tilde{\mathbf{Y}}} \mathbf{1}_{\tilde{N} \times N} \right] \end{aligned} \tag{7}$$

where $\mathbf{K}_{\mathbf{Y}\tilde{\mathbf{Y}}} = \mathbf{\Phi}_{\mathbf{Y}} \mathbf{\Phi}_{\tilde{\mathbf{Y}}}^\top$ is a Gram matrix between $\mathcal{D}_{\mathbf{Y}}$ and $\mathcal{D}_{\tilde{\mathbf{Y}}}$ whose values are obtained from the kernel function $k(\cdot, \cdot)$. Since the elements of the Gram matrix are calculated using the kernel function, it is unnecessary to calculate infinite dimensional vector explicitly obtained from mapping $\phi(\cdot)$. By applying this equation to the inner products in (5), we can estimate MMD from data samples as follows:

$$\begin{aligned}\mathrm{MMD}^2 = &\frac{1}{N^2} \mathrm{Tr}\left[\mathbf{K}_{\mathbf{YY}} \mathbf{1}_{N \times N} \right] + \frac{1}{\tilde{N}^2} \mathrm{Tr}\left[\mathbf{K}_{\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}} \mathbf{1}_{\tilde{N} \times \tilde{N}} \right] \\ &- 2\frac{1}{N\tilde{N}} \mathrm{Tr}\left[\mathbf{K}_{\mathbf{Y}\tilde{\mathbf{Y}}} \mathbf{1}_{\tilde{N} \times N} \right]. \end{aligned} \tag{8}$$

MMD is regarded as a nonparametric method that does not have to assume the distribution P and $\tilde{P}$ to be parametric.

2.2. Conditional MMD [5]

We consider the case where two distributions P and $\tilde{P}$ are conditional ones given arbitrary input vectors $\mathbf{x}(\in \mathcal{X})$. In this case, the distance between the distributions is expressed by the following equation using the witness function $f \in \mathcal{F}$ in the same way as MMD:

$$\left| \mathbb{E}_{Y \sim P}[f(Y; \mathbf{x})] - \mathbb{E}_{\tilde{Y} \sim \tilde{P}}[f(\tilde{Y}; \mathbf{x})] \right|. \tag{9}$$

By using a conditional mean $\mu_{Y|\mathbf{x}}$, the expectation is represented by

$$\mathbb{E}_{Y \sim P}[f(Y; \mathbf{x})] = \langle f, \mu_{Y|\mathbf{x}} \rangle_{\mathcal{F}}. \tag{10}$$

Here, we consider another RHKS $\mathcal{G}$ whose kernel function is $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and mapping $\psi : \mathcal{X} \to \mathbb{R}$, $\psi(\mathbf{x}) \triangleq k(\mathbf{x}, \cdot)$. Moreover, we assume that the conditional mean embedding $\mu_{Y|\mathbf{x}}$ is given by a linear transformation as follows:

$$\mu_{Y|\mathbf{x}} = C_{Y|X} \psi(\mathbf{x}) \tag{11}$$

where $C_{Y|X}$ is a linear operator of tensor product Hilbert space $\mathcal{F} \otimes \mathcal{G}$. Hence, the inner product can be converted using tensor product $\otimes$ as follows:

$$\begin{aligned}\langle f, \mu_{Y|\mathbf{x}} \rangle_{\mathcal{F}} &= \langle f, C_{Y|X} \psi(\mathbf{x}) \rangle_{\mathcal{F}} &(12)\\ &= \langle f \otimes \psi(\mathbf{x}), C_{Y|X} \rangle_{\mathcal{F} \otimes \mathcal{G}}. &(13)\end{aligned}$$

Finally, we obtain conditional MMD (CMMD) given by

$$\mathrm{CMMD} = \sup_{\substack{\|g(\mathbf{x})\| \leq 1 \\ g(\mathbf{x}) \in \mathcal{F} \otimes \mathcal{G}}} \langle g(\mathbf{x}), C_{Y|X} - C_{\tilde{Y}|\tilde{X}} \rangle_{\mathcal{F} \otimes \mathcal{G}} \tag{14}$$

$$\mathrm{CMMD}^2 = \|C_{Y|X} - C_{\tilde{Y}|\tilde{X}}\|_{\mathcal{F} \otimes \mathcal{G}}^2. \tag{15}$$

When we have the data $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_1, \mathbf{y}_N)\}$, the estimate of the linear operator $C_{Y|X}$ is given by the following equation [10].

$$\hat{C}_{Y|X} = \mathbf{\Phi}_{\mathbf{Y}} (\mathbf{\Upsilon}_{\mathbf{X}} \mathbf{\Upsilon}_{\mathbf{X}}^\top + \lambda \mathbf{I})^{-1} \mathbf{\Upsilon}_{\mathbf{X}}^\top \tag{16}$$

where $\mathbf{\Phi}_{\mathbf{Y}} = [\phi(\mathbf{y}_1), \ldots, \phi(\mathbf{y}_N)]^\top$, and $\mathbf{\Upsilon}_{\mathbf{X}} = [\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_N)]^\top$. $\lambda (> 0)$ is a regularization constant. Then, the estimate of CMMD is derived as follows:

$$\begin{aligned}\mathrm{CMMD}^2 = &\Big\| \mathbf{\Phi}_{\mathbf{Y}} (\mathbf{\Upsilon}_{\mathbf{X}} \mathbf{\Upsilon}_{\mathbf{X}}^\top + \lambda \mathbf{I})^{-1} \mathbf{\Upsilon}_{\mathbf{X}}^\top \\ &- \mathbf{\Phi}_{\tilde{\mathbf{Y}}} (\mathbf{\Upsilon}_{\tilde{\mathbf{X}}} \mathbf{\Upsilon}_{\tilde{\mathbf{X}}}^\top + \lambda \mathbf{I})^{-1} \mathbf{\Upsilon}_{\tilde{\mathbf{X}}}^\top \Big\|_{\mathcal{F} \otimes \mathcal{G}}^2 \\ = &\mathrm{Tr}\left[\mathbf{K}_{\mathbf{Y},\mathbf{Y}} \bar{\mathbf{H}}_{\mathbf{X},\mathbf{X}}^{-1} \mathbf{H}_{\mathbf{X},\mathbf{X}} \bar{\mathbf{H}}_{\mathbf{X},\mathbf{X}}^{-1} \right] \\ &+ \mathrm{Tr}\left[\mathbf{K}_{\tilde{\mathbf{Y}},\tilde{\mathbf{Y}}} \bar{\mathbf{H}}_{\tilde{\mathbf{X}},\tilde{\mathbf{X}}}^{-1} \mathbf{H}_{\tilde{\mathbf{X}},\tilde{\mathbf{X}}} \bar{\mathbf{H}}_{\tilde{\mathbf{X}},\tilde{\mathbf{X}}}^{-1} \right] \\ &- 2\,\mathrm{Tr}\left[\mathbf{K}_{\mathbf{Y},\tilde{\mathbf{Y}}} \bar{\mathbf{H}}_{\tilde{\mathbf{X}},\tilde{\mathbf{X}}}^{-1} \mathbf{H}_{\tilde{\mathbf{X}},\mathbf{X}} \bar{\mathbf{H}}_{\mathbf{X},\mathbf{X}}^{-1} \right] \end{aligned} \tag{17}$$

where $\mathbf{H}_{\mathbf{X},\tilde{\mathbf{X}}} = \mathbf{\Upsilon}_{\mathbf{X}} \mathbf{\Upsilon}_{\tilde{\mathbf{X}}}^\top$ is the Gram matrix for input features and $\bar{\mathbf{H}}_{\mathbf{X},\tilde{\mathbf{X}}} = \mathbf{H}_{\mathbf{X},\tilde{\mathbf{X}}} + \lambda \mathbf{I}$.

For simplicity, we assume that two data have the same input, that is $\mathbf{X} = \tilde{\mathbf{X}}$, then we obtain the following equations.

$$\mathrm{CMMD}^2 = \mathrm{Tr}\left[\left(\mathbf{K}_{\mathbf{Y},\mathbf{Y}} + \mathbf{K}_{\tilde{\mathbf{Y}},\tilde{\mathbf{Y}}} - 2\mathbf{K}_{\mathbf{Y},\tilde{\mathbf{Y}}} \right) \mathbf{L} \right] \tag{18}$$

$$\mathbf{L} = (\mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H} (\mathbf{H} + \lambda \mathbf{I})^{-1}. \tag{19}$$

Compared with MMD, CMMD multiplies $\mathbf{L}$ instead of the matrix of ones $\mathbf{1}$. This means that the kernel values of output variables in CMMD are weighted by the kernel values of input variables.

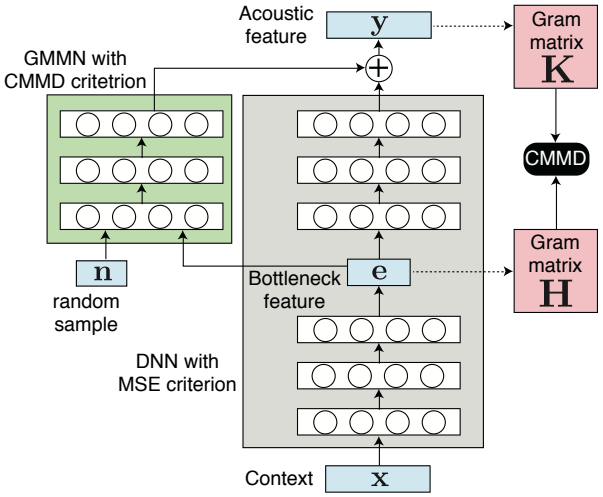

Figure 1: *Network of GMMN-based speech synthesis.*

3. GMNN-based speech synthesis

Generative moment matching network (GMMN) [4] is a generative neural network trained by MMD criterion obtained from the distributions of original and generated samples. In GMMN, we use random values sampled from a simple prior distribution, such as uniform and standard normal distributions, as an input of the neural network. It is supposed that GMMN yields a similar distribution to that of training data. Conditional GMMN [5] is a special case of GMMN in which a conditional distribution is predicted. For an input vector of conditional GMMN, condition vector and random values are concatenated, and CMMD is used as the criterion of conditional GMMN. We refer to both GMMN and conditional GMMN as GMMN in this paper. GMMN can represent various distribution because it is unnecessary to restrict probabilistic distribution function by a parametric one. Moreover, we can easily generate a sample of output distribution by inputting a random value to the neural network.

In GMMN-based speech synthesis [3], the conditional distribution of acoustic features given frame-level contexts is predicted by a neural network. Fig. 1 shows the model architecture used in this study. The model is composed of two neural networks: DNN that outputs center of acoustic features, and GMMN that outputs the variation from the center. We use low-dimensional bottleneck features obtained by DNN as an input of GMMN because it is hard to deal with high-dimensional context vector for kernel functions. To train the model, we first train of DNN with a mean squared error (MSE)-criterion in the same way as traditional DNN-based speech synthesis [11]. Next, we fix the DNN and train GMMN using bottleneck features obtained by the DNN. The parameters of GMNN are optimized to minimize CMMD between generated and original distributions of acoustic features by backpropagation. Note that we trained a GMMN that directly outputs the distribution of acoustic features in the previous study [3]. In this study, on the other hand, we train a GMMN that outputs the difference from DNN with MSE-criterion for training stability.

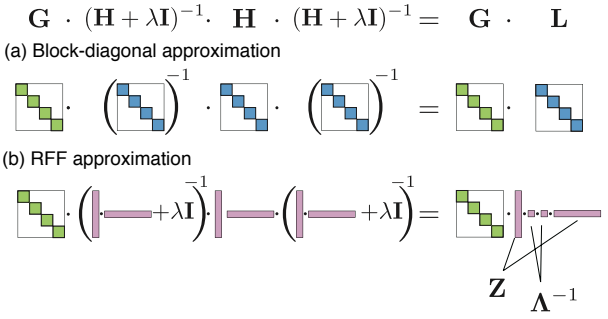

Figure 2: *Comparison of the approximation methods of Gram matrices for input variables.*

4. Approximated CMMD for GMNN training

The problem in GMMN is computational complexity. When the total number of training data points is N, (18) requires $\mathcal{O}(N^3)$[1] and $\mathcal{O}(N^2)$ for inverse matrix and trace operation, respectively. Therefore, the computation is infeasible for large N, which includes speech synthesis case. Moreover, a minibatch-wise loss function is required to perform minibatch-based gradient descent optimization, which is generally used in DNN training. In this study, we investigate approximation methods of CMMD for the training of GMMN-based speech synthesis.

4.1. Approximation of Gram matrices

First, we approximate the Gram matrix of output variables $\mathbf{G} \triangleq (\mathbf{K}_{\mathbf{Y}} + \mathbf{K}_{\tilde{\mathbf{Y}}} - 2\mathbf{K}_{\mathbf{Y},\tilde{\mathbf{Y}}})$ in (18) by block diagonal matrix $\mathrm{diag}[\mathbf{G}_1, \ldots, \mathbf{G}_S]$, where S is the number of minibatches. In the previous work [3], we also approximated the Gram matrix of $\mathbf{L}$ by the block diagonal ones $\mathbf{L}_i$ $(i = 1 \ldots S)$, the approximated CMMD becomes $\mathcal{L} = \sum_{i=1}^{S} \mathrm{Tr}[\mathbf{G}_i \mathbf{L}_i]$ as shown in Fig. 2 (a). Since the loss function is regarded as the sum of minibatch-wise CMMD, the minibatch-based optimization is available and the computation complexity for each minibatch requires $\mathcal{O}(B^3)$ when the number of data points of a minibatch is B.

In addition to the block-diagonal approximation, we propose a low-rank approximation method based on random Fourier features (RFFs) [8] for input features. The difference between these two methods of approximation is illustrated in Fig. 2. RFF approximates a kernel function by the inner product of finite-dimensional vectors. Specifically, a widely-used radial basis function (RBF) kernel defined by

$$h_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) \tag{20}$$

can be approximated by the following equation:

$$h_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') \approx \frac{1}{M} \sum_{r=1}^{M} \cos(\mathbf{x}^\top \boldsymbol{\omega}_r + b_r) \cos(\mathbf{x}'^\top \boldsymbol{\omega}_r + b_r) \tag{21}$$

where M is the dimensionality of RFF, and $\boldsymbol{\omega}_r$ and b_r are samples from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and a uniform distribution $\mathcal{U}[0, 2\pi)$, respectively.

[1] To be exact, a Strassen algorithm can reduce computation complexity to $\mathcal{O}(N^{\log_2 7}) \approx \mathcal{O}(N^{2.807})$. However, the matrix inversion still requires huge computational cost.

Let $\mathbf{Z}$ be the matrix form of $M(\ll N)$-dimensional RFF vectors given by

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^\top \tag{22}$$

$$\mathbf{z}_n = \left[\cos(\mathbf{x}_n^\top \boldsymbol{\omega}_1 + b_1), \ldots, \cos(\mathbf{x}_n^\top \boldsymbol{\omega}_M + b_M)\right]^\top \tag{23}$$

and we obtain an approximated Gram matrix of input feature as follows:

$$\mathbf{H} \approx \mathbf{Z}\mathbf{Z}^\top. \tag{24}$$

By using Woodbury identity [12], the matrix $\mathbf{L}$, which includes inverse of $\mathbf{H}$, is transformed as:

$$\begin{aligned}\mathbf{L} &\approx (\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I})^{-1}\mathbf{Z}\mathbf{Z}^\top(\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I})^{-1} \\ &= \mathbf{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}^{-1}\mathbf{Z}^\top \end{aligned} \tag{25}$$

$$\boldsymbol{\Lambda} = \mathbf{Z}^\top\mathbf{Z}/\lambda + \mathbf{I}. \tag{26}$$

$\boldsymbol{\Lambda}$ can be calculated in advance before the training of GMMN because it is independent of output features. The element of block diagonal matrix $\mathbf{L}_i$ becomes $\mathcal{O}(BM^2)$. The approximated $\mathbf{L}$ by RFF is supposed to have global characteristics of training data, while the block diagonal matrices have local features of minibatches. Moreover, since the RFF approximation does not require $\mathcal{O}(B^3)$ associated with the inverse matrix calculation as is required in the block diagonal approximation, RFF approximation can use larger minibatch size than the block diagonal approximation.

4.2. Minibatch selection using clustering

In this study, we examine the method to choose minibatch. When the amount of training data is large and the output features are diverse, the Gram matrix of output feature, $\mathbf{K}$, tends to be sparse. For example, the kernel function values between different phonemes become very small. Due to the sparsity, only a small amount of information is used to optimize GMMN parameters.

To overcome this problem, we propose a clustering-based minibatch selection to get together similar training data points. For the clustering, we employed K-means clustering with bottleneck features. In order to limit the maximum size of each cluster, we recursively divide the training data by 2-class K-means clustering until the size of each cluster becomes lower than a given value.

5. Experiments

5.1. Experimental conditions

We used a database of Japanese speech recorded by a female speaker for the experimental evaluations. The database included 203 sentences, and each sentence was recorded five times to take the inter-utterance diversity into account. 100 sentences were included in JSUT corpus [13] and the rest 103 sentences were from ATR phonetically balanced Japanese sentences [14]. The training sentences were 750 utterances which consisted of five recordings for every 150 sentence. 26 and 27 sentences were used for validation and test data, respectively.

We extracted F0, spectral envelope, and aperiodicity from STRAIGHT [15] every 5 ms from the speech signal downsampled by a sampling rate of 16 kHz, and obtained 0-39th mel-cepstrum, log F0, and 5-band aperiodicity. We used a 139-dimensional vector consisting of Δ, Δ^2, and V/UV flags as

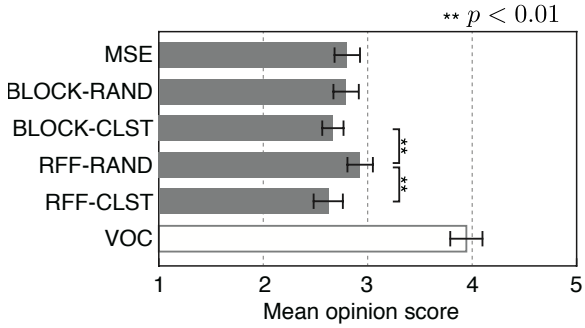

Figure 3: *Subjective evaluation on naturalness (1: too bad, 5: very good).*

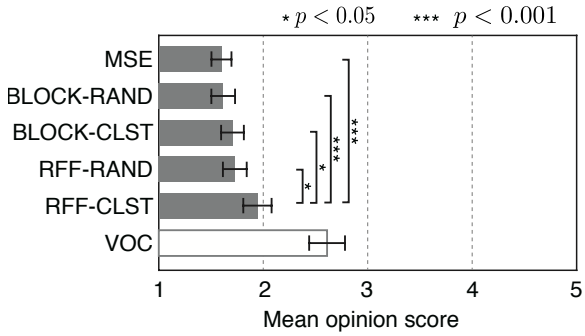

Figure 4: *Subjective evaluation on the inter-utterance variation of two synthetic speech samples (1: completely equivalent, 5: very different).*

acoustic features. The acoustic features were normalized to a range $[-1, 1]$. We used a 556-dimensional contextual vector as an input vector obtained from questions about context and frame position information, The input vector was normalized to zero mean and unit variance. The dimensionalities of bottleneck features and random samples were 128 and 3, respectively,

We used model architecture shown in Fig. 1. The DNN trained by MSE criterion consists of three encoder layers and three decoder layers. The number of hidden layers of GMMN was three. We set the number of hidden units of the DNN and GMMN to 512. We used ReLU activation function for hidden layers and $\tanh$ for bottleneck and output layers.

The kernel functions of CMMD were RBF kernels defined by

$$k_{\mathrm{RBF}}(\mathbf{y}, \mathbf{y}') = \exp\left(-\frac{\|\mathbf{y} - \mathbf{y}'\|^2}{2l_{\mathbf{y}}^2}\right) \tag{27}$$

$$h_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l_{\mathbf{x}}^2}\right) \tag{28}$$

where $l_{\mathbf{y}}$ and $l_{\mathbf{x}}$ are scale parameters for input and output features, respectively. We adopted the statistic of Euclidean distance between two input/output feature vectors to determine the scale parameters. Specifically, the median distance was used for $l_{\mathbf{y}}$ and the half value of maximum distance was used for $l_{\mathbf{x}}$. The dimensionality of RFF was set to 1024, and the regularization constant λ was 0.01. For the K-means-based method, the maximum number of frames of each cluster was 1024.

The minibatch size was 1024 for DNN training and 10000 for GMMN training. We employed parameter optimization based on Adam [16] with learning rate 0.001. A dropout rate was 20% and weight decay with coefficient 10^{-6} was used for

Table 2: *Average of the frame-level standard deviations of synthetic speech parameters.*

	Mel-cepstrum		Log F0	Duration
	0th	1st	[cent]	[msec]
BLOCK-RAND	0.0230	0.0121	15.80	2.46
BLOCK-CLST	0.0528	0.0216	18.23	3.50
RFF-RAND	0.0208	0.0066	1.54	3.77
RFF-CLST	0.0493	0.0266	13.97	5.47

regularization, and batch normalization was adopted to avoid gradient vanishing. Early stopping was performed using the validation set and the maximum number of training epochs was 300.

For experimental evaluations, we synthesized five speech samples using random values for each 27 test sentence[2]

5.2. Subjective evaluation

We conducted crowd-sourcing-based subjective evaluations by mean opinion score (MOS) tests to examine the effectiveness of approximation methods. We employed two tests which measured naturalness and inter-utterance variation, respectively. In the naturalness test, the participants listened to speech samples and rate the naturalness on them on a five-point scale (1: too bad, 5: very good). In the inter-utterance variation test, we chose pairs of synthetic speech samples of the same sentence, which were generated using different random values. The participants listened to the pair and the similarity of two speech samples was graded by a five-point scale (1: completely different, 5: very different). The number of participants on a crowd-sourcing service was 60 for respective tests and 3 sentences were randomly selected individually for each participant from the 27 test sentences.

The results of naturalness and inter-utterance variation are shown in 3 and 4, respectively. MSE in the figure is the result of DNN trained by MSE criterion, which did not perform random sampling. VOC denotes vocoded speech samples, which were re-synthesized from extracted acoustic features. BLOCK and RFF correspond to block diagonal and RFF approximation methods. For the minibatch selection methods, RAND represents a random selection and CLST means K-means clustering.

When comparing naturalness, we find that RFF-RAND gave the highest score among the GMMN-based methods and clustering-based minibatch selection degraded the naturalness of synthetic speech. The scores of GMM-based methods were comparable with MSE. As for inter-utterance variation, RFF-CLST yielded higher scores than the other GMMN-based methods and MSE. We should note that VOC was only 2.61 even though the original recordings were not equivalent to each other. Moreover, although the synthetic samples of MSE had no difference, the score was not 1 but 1.60. A possible reason of the narrow score range is that the variation of original recording was not large enough for the listeners to perceive because the recording samples were reading-style speech not including expressiveness.

5.3. Inter-utterance variation

To explore the detail of subjective evaluation results, we examined the variance of generated speech parameters. We calculated the average of standard deviations of respective frames obtained from randomly generated 5 samples. The results for 0th and 1st mel-cepstral coefficients, log F0, and phone duration are shown in Table 2. When we calculated the standard deviation of log F0s, unvoiced frames were removed. From the table, it is seen that the standard deviations of mel-cepstral coefficients of BLOCK-CLST and RFF-CLST, which performed clustering-based minibatch selection, were larger than the other methods. On the other hand, RFF-CLST had the largest standard deviation in phone duration and was followed by RFF-RAND. When comparing the standard deviation of log F0s, we see that RFF-RAND yielded very small variation. Moreover, the scores of inter-utterance variation increased with the increase of the standard deviation of generated phone durations. Therefore, the phone duration variation might have been a dominant factor in the subjective evaluation of inter-utterance variation.

5.4. Random sample example

We plot the generated speech parameters in Fig. 5 as an example. These figures shows the speech parameter using 5 different random values. The mel-cepstral coefficients and log F0s were generated using phone durations of original speech to make the difference clear. The 5 contours with different colors were corresponding to speech parameters of respective random samples. It is seen that the variation in 1st mel-cepstral coefficients was too small to affect the perception of synthetic speech for all methods. When comparing log F0s, we see that the contours of BLOCK-CLST and RFF-CLST varied dependent on random values. The phone durations had variation in all methods in Fig. 5(c), which means that the random sampling was effective in duration models.

6. Conclusions

In this paper, we investigated the approximation methods of CMMD for GMMN-based speech synthesis. In addition to conventional block diagonal approximation, we proposed RFF-based approximation of Gram matrices for input features. Moreover, we introduced the minibatch selection using K-means clustering with bottleneck features to avoid sparse Gram matrices. Through the experimental evaluations, it is found that the method using RFF approximation and clustering yielded larger variance of generated phone durations, and this could have enhanced the perceptual score of inter-utterance variation. For future work, we will compare the performance with other generative models such as variational auto-encoder and generative adversarial network. Otherwise, giving random noise and bias for speech parameters can yield inter-utterance variation. Therefore, future work includes the evaluation of trade-off between variation and naturalness with the simple methods of random generation. Moreover, we should use a large speech dataset and a dataset with larger speech variation to enhance the score of VOC.

7. Acknowledgements

Part of this work was supported by SECOM Science and Technology Foundation.

8. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS

[2] Synthetic speech samples are available at `https://hyama5.github.io/demo_GMMN_TTS`.

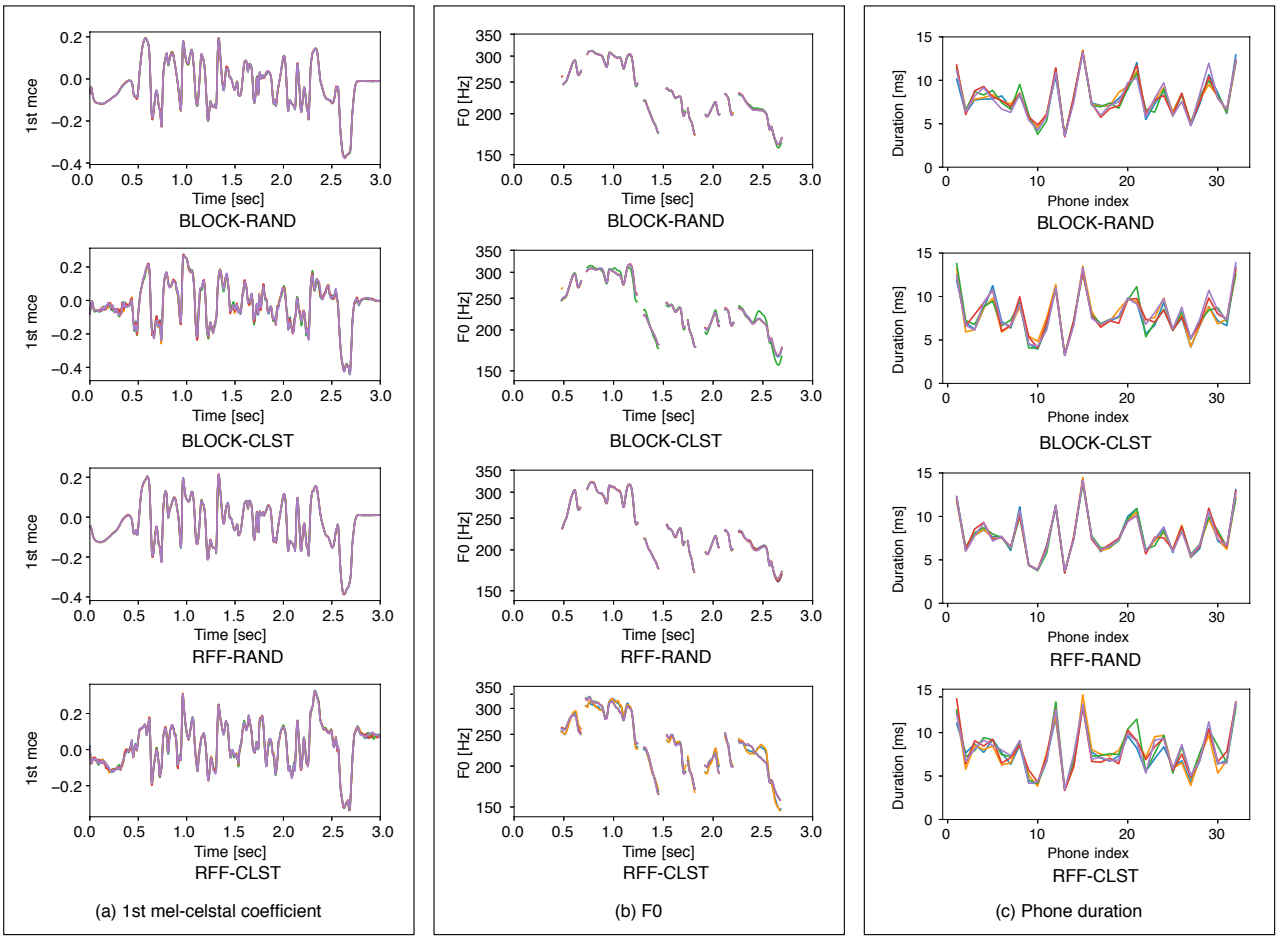

Figure 5: *Randomly generated speech parameters for sentence "平均倍率を下げた形跡がある (heekin bairitsuwo sageta keesekiga aru)"*

synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[2] H.-T. Luong, X. Wang, J. Yamagishi, and N. Nishizawa, "Investigating accuracy of pitch-accent annotations in neural network-based speech synthesis and denoising effects," in *Proc. INTERSPEECH*, 2018, pp. 37–41.

[3] S. Takamichi, T. Koriyama, and H. Saruwatari, "Sampling-based speech parameter generation using moment-matching networks," in *Proc. INTERSPEECH*, 2017, pp. 3961–3965.

[4] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Proc. ICML*, 2015, pp. 1718–1727.

[5] Y. Ren, J. Zhu, J. Li, and Y. Luo, "Conditional generative moment-matching networks," in *Proc. NIPS*, 2016, pp. 2928–2936.

[6] S. Shiota, S. Takamichi, , and T. Matsui, "Data augmentation with moment-matching networks for i-vector based speaker verification," in *Proc. APSIPA*, 2018, pp. 345–349.

[7] H. Tamaru, Y. Saito, S. Takamichi, T. Koriyama, and H. Saruwatari, "Generative moment matching network-based random modulation post-filter for DNN-based singing voice synthesis and neural double-tracking," in *Proc. ICASSP*, 2019, pp. 7070–7074.

[8] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. NIPS*, 2008, pp. 1177–1184.

[9] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, pp. 723–773, 2012.

[10] L. Song, J. Huang, A. Smola, and K. Fukumizu, "Hilbert space embeddings of conditional distributions with applications to dynamical systems," in *Proc. ICML*, 2009, pp. 961–968.

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

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Univ. Press, 1992.

[13] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis," *arXiv preprint arXiv:1711.00354*, 2017.

[14] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "ATR Japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, Aug. 1990.

[15] 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. 3-4, pp. 187–207, 1999.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.