Odyssey 2018 The Speaker and Language Recognition Workshop
26-29 June 2018, Les Sables d'Olonne, France

Bidirectional Voice Conversion Based on Joint Training Using Gaussian-Gaussian Deep Relational Model

Kentaro Sone[1], *Shinji Takaki*[2], and Toru Nakashika[1]

[1]Graduate School of Informatics and Engineering,
The University of Electro-Communications, Japan
[2]National Institute of Informatics (NII), Japan

sone@sd.is.uec.ac.jp, nakashika@uec.ac.jp

Abstract

Statistical approaches to voice conversion based on Gaussian mixture models (GMMs) have been investigated in the last decade. These approaches attempt to model the joint distribution of source and target speakers utterances using GMMs. However, since GMMs do not have enough representation capability, they have been replaced by deep neural networks (DNNs). The DNN-based approaches attempt to represent feed-forward dependencies from source utterances into target utterances using DNNs. Owing to the high representation capability of DNNs, these approaches improved qualities of the converted speech. Although the performances are improved by DNNs, DNN-based approaches cannot convert target utterances into source utterances like GMM-based approaches can. Therefore, DNN-based approaches cost twice as much to train as GMM-based approaches. To classify and generate binary-valued images, a deep relational model (DRM) has been proposed. A DRM consists of two visible layers and multiple hidden layers the same as DNNs and can classify and generate images by modeling a bidirectional relationship between images and labels. In this paper, we define a Gaussian-Gaussian DRM (GGDRM), which is the Gaussian-Gaussian form of the traditional DRM, and propose a method to apply a GGDRM to voice conversion. Experimental results show that our GGDRM-based method outperforms GMM- and DNN-based methods.

1. Introduction

Voice conversion is a technique to transform an utterance of a source speaker so that it is recognized as if it were an utterance of a target speaker. Various voice conversion approaches [1, 2, 3] have been proposed since the first one, the code book-based method [4], was proposed. Among them, approaches based on Gaussian mixture models (GMMs) [5, 6] or deep neural networks (DNNs) [7, 8, 9] have been widely investigated.

In the approaches based on GMMs [5, 6], a GMM models the joint distribution of acoustic features of source and target speakers, and the parameters are estimated in the maximum likelihood criterion. Then the obtained parameters are used to convert the utterance of a source speaker into the utterance of a target speaker. Owing to the high flexibility of those, GMMs are widely used. However, when the joint distribution is modeled, vectors representing the acoustic features of the source and target speakers are combined and treated as one vector. Therefore, the feature spaces of the two visible variables (the source and target speaker) are not explicitly separated. As a result, since the GMM optimizes parameters by using vectors combining features of two visible variables, the dimensions of the feature space become large, and it can be said that GMMs are more susceptible to overfitting depending on the representation capability of the model.

On the other hand, in the approaches based on DNNs [7, 8, 9], a DNN, which is a neural network consisting of multiple hidden layers, represents the feedforward dependencies from acoustic features of source speakers into those of target speakers instead of modeling the joint distribution of acoustic features of the speaker pair like a GMM does. These approaches have been reported to improve qualities of the converted speech because 1) the representation capability of a DNN stacking non-linear transformations in multiple layers is higher than that of a GMM, and 2) the feature spaces of two visible variables can be explicitly separated owing to two visible layers (input and output layers).

In the domain of binary-valued image classification and generation, Nakashika [10] proposed a deep relational model (DRM), which can potentially classify and generate binary-valued images. The DRM models a joint distribution of the two variables and contains multiple hidden layers to capture their latent dependencies. Since DNNs improve the performances in voice conversion, they have recently replaced GMMs. However, since a GMM is the joint model, it can convert input-to-output and output-to-input bidirectionally after one training. Meanwhile, since a DNN represents only feedforward dependencies input-to-output, it cannot convert bidirectionally like a GMM can. To convert output-to-input using a DNN, another training of DNN is required. A DRM has high representation capability owing to multiple hidden layers the same as a DNN and can convert input-to-output and output-to-input bidirectionally the same as a GMM. In this paper, we focus on a DRM, which has the deep architecture and the capability to separate two visible variables explicitly the same as a DNN, and attempt to extract the bidirectional relationships between source and target speakers in voice conversion. However, since visible variables of the conventional DRM follow a Bernoulli distribution, it is not suitable for voice conversion, which involves acoustic features consisting of real-valued data. Thus, we define a Gaussian-Gaussian DRM (GGDRM), which is the Gaussian-Gaussian form of the DRM, and propose a GGDRM-based pre-training method for DNN-based voice conversion systems.

This paper is organized as follows. Section 2 describes the conventional DRM in the Bernoulli-Bernoulli form. Our proposed GGDRM-based method and its definition are described in Section 3. Experimental results are presented in Section 4. Concluding remarks are given in the final section.

10.21437/Odyssey.2018-37

2. Deep Relational Model

In this section, we introduce a traditional DRM. The same as a restricted Boltzmann machine (RBM) [11] and a deep Boltzmann machine (DBM) [12, 13], a DRM is an undirected graphical model with a set of visible and hidden units [10]. A DRM consists of two visible layers (the first visible variables $\boldsymbol{x} \in \{0,1\}^I$ and the second visible variables $\boldsymbol{y} \in \{0,1\}^K$) and multiple hidden variables $\boldsymbol{h}^{(l)} \in \{0,1\}^{J_l}$ $(l = 1, ..., L)$, where L is the number of hidden layers. A DRM has symmetric connections between the units in adjacent layers and no connections between the units in the same layer. A DRM is defined on the basis of the energy function to capture high-order relationships between two observable variables $\boldsymbol{x}$ and $\boldsymbol{y}$. The joint probability distribution using a DRM is defined as follows:

$$p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \sum_{\forall \boldsymbol{h}^{(l)}} p(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta}) \tag{1}$$

$$p(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\{-E(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta})\}, \tag{2}$$

where Z is the partition function. In a DRM, the energy function E is defined as:

$$E(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta}) = -\boldsymbol{b}^T\boldsymbol{x} - \sum_{l=1}^{L} \boldsymbol{c}^{(l)T}\boldsymbol{h}^{(l)} - \boldsymbol{d}^T\boldsymbol{y}$$
$$-\boldsymbol{x}^T\boldsymbol{W}^{(1)}\boldsymbol{h}^{(1)} - \sum_{l=2}^{L} \boldsymbol{h}^{(l-1)T}\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l)} - \boldsymbol{h}^{(L)T}\boldsymbol{W}^{(L+1)}\boldsymbol{y}, \tag{3}$$

where $\boldsymbol{b} \in \mathbb{R}^I$, $\boldsymbol{c}^{(l)} \in \mathbb{R}^{J_l}$ and $\boldsymbol{d} \in \mathbb{R}^K$ are the bias parameters corresponding to the units in the first visible layer, the l-th hidden layer and the second visible layer. $\boldsymbol{W}^{(1)} \in \mathbb{R}^{I \times J_1}$, $\boldsymbol{W}^{(l)} \in \mathbb{R}^{J_{l-1} \times J_l}$ and $\boldsymbol{W}^{(L+1)} \in \mathbb{R}^{J_L \times K}$ are the weight parameters of connections between the first visible layer and the first hidden layer, $(l-1)$-th and l-th hidden layer, and L-th hidden layer and the second visible layer, respectively.

Under the definition of the energy function, the conditional distributions for each visible and hidden unit given adjacent units are

$$p(x_i = 1|\boldsymbol{h}^{(1)}) = \sigma(b_i + \boldsymbol{W}^{(1)}_{i:}\boldsymbol{h}^{(1)}) \tag{4}$$

$$p(h^{(l)}_j = 1|\boldsymbol{h}^{(l-1)}, \boldsymbol{h}^{(l+1)}) = \sigma(c^{(l)}_j + \boldsymbol{W}^{(l)T}_{:j}\boldsymbol{h}^{(l-1)} + \boldsymbol{W}^{(l+1)}_{j:}\boldsymbol{h}^{(l+1)}) \tag{5}$$

$$p(y_k = 1|\boldsymbol{h}^{(L)}) = \sigma(d_k + \boldsymbol{W}^{(L)T}_{:k}\boldsymbol{h}^{(L)}), \tag{6}$$

where $\sigma(\cdot)$ denotes the logistic sigmoid function. Note that the hidden variables $\boldsymbol{h}^{(0)}$ and $\boldsymbol{h}^{(L+1)}$ are regarded as $\boldsymbol{h}^{(0)} = \boldsymbol{x}$ and $\boldsymbol{h}^{(L+1)} = \boldsymbol{y}$, respectively, in Eq. (5).

The parameters of a DRM $\boldsymbol{\theta} = \{\boldsymbol{b}, \boldsymbol{c}^{(l)}, \boldsymbol{d}, \boldsymbol{W}^{(1)}, \boldsymbol{W}^{(l)}, \boldsymbol{W}^{(L+1)}\}$ are optimized to maximize the joint log-likelihood $\mathcal{L} = \log \prod_t p(\boldsymbol{x}^t, \boldsymbol{y}^t; \boldsymbol{\theta})$. The partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{\theta}$ is computed as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \left\langle -\frac{\partial E}{\partial \boldsymbol{\theta}} \right\rangle_d - \left\langle -\frac{\partial E}{\partial \boldsymbol{\theta}} \right\rangle_m, \tag{7}$$

where shorthand notations $\langle \cdot \rangle_d$ and $\langle \cdot \rangle_m$ denote the expectations computed over the data and model distributions, respectively. The second term in Eq. (7) is computationally difficult. Therefore, the second term is approximated by using the mean-field update in the training stage of a DRM.

To improve parameter optimization, a pre-training scheme is used similarly to the greedy layer-wise training in a deep belief network (DBN). First, RBMs are trained at the lowest and highest levels ($\boldsymbol{x}$ and $\boldsymbol{y}$). Second, RBMs are trained using the expected values of the hidden units from previously trained RBMs given $\boldsymbol{x}$ and $\boldsymbol{y}$. In this way, RBMs are trained from outer to inner in the pre-training stage of the DRM. The training is described in more detail by Nakashika [10].

3. Applying DRM Concepts to Voice Conversion

In this section, we introduce our model, a Gaussian-Gaussian DRM (GGDRM), whose visible layer consists of real-valued units unlike a traditional DRM which is described in Section 2. To distinguish a traditional DRM and a GGDRM explicitly, we refer to the former as a Bernoulli-Bernoulli DRM (BBDRM).

As we mentioned in Section 1, the advantage of the GGM-based approaches to voice conversion is that since a GMM models joint distributions of source and target speakers, it can convert source-to-target and target-to-source bidirectionally. However, since feature vectors of source and target speakers are concatenated in the training stage, a GMM cannot optimize the parameters considering dependencies from source to target or vice versa. On the other hand, the advantages of the DNN-based approaches are they have 1) better representation capability than GMM-based approaches owing to multiple hidden layers and 2) a model structure to separate inputs and outputs explicitly. However, a DNN cannot convert source-to-target and target-to-source bidirectionally like a GMM can. To improve the performances of voice conversion, we focus on a BBDRM that 1) models bidirectional relationships between source utterances and target utterances using deep architecture and 2) trains with a structure separating inputs and outputs explicitly. Since a BBDRM has been developed to model bidirectional relationships between two binary variables, it is not suitable to model bidirectional relationships between acoustic features of a source speaker and those of a target speaker. To address this issue, we define a GGDRM, which represents two Gaussian distributions. After that, we describe our proposed voice conversion method based on a GGDRM.

3.1. Gaussian-Gaussian DRM

A Gaussian-Bernoulli RBM (GBRBM) [14] was originally proposed to model real-valued data. Later, an improved GBRBM (IGBRBM) [15] was proposed to improve training of GBRBMs, which is difficult due to the variance parameters. Referring to an IGBRBM, we define the energy function of a GGDRM as follows:

$$E(\boldsymbol{x}, \boldsymbol{y}, \forall \boldsymbol{h}^{(l)}; \boldsymbol{\theta}) =$$
$$\frac{1}{2}\left(\frac{\boldsymbol{x}-\boldsymbol{b}}{\boldsymbol{\sigma}^{(x)}}\right)^T \left(\frac{\boldsymbol{x}-\boldsymbol{b}}{\boldsymbol{\sigma}^{(x)}}\right) - \left(\frac{\boldsymbol{x}}{\boldsymbol{\sigma}^{(x)} \circ \boldsymbol{\sigma}^{(x)}}\right)^T \boldsymbol{W}^{(1)}\boldsymbol{h}^{(1)}$$
$$-\sum_{l=1}^{L} \boldsymbol{c}^{(l)T}\boldsymbol{h}^{(l)} - \sum_{l=2}^{L} \boldsymbol{h}^{(l-1)T}\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l)}$$
$$+\frac{1}{2}\left(\frac{\boldsymbol{y}-\boldsymbol{d}}{\boldsymbol{\sigma}^{(\boldsymbol{y})}}\right)^T \left(\frac{\boldsymbol{y}-\boldsymbol{d}}{\boldsymbol{\sigma}^{(\boldsymbol{y})}}\right) - \boldsymbol{h}^{(L)T}\boldsymbol{W}^{(L+1)}\left(\frac{\boldsymbol{y}}{\boldsymbol{\sigma}^{(y)} \circ \boldsymbol{\sigma}^{(y)}}\right), \tag{8}$$

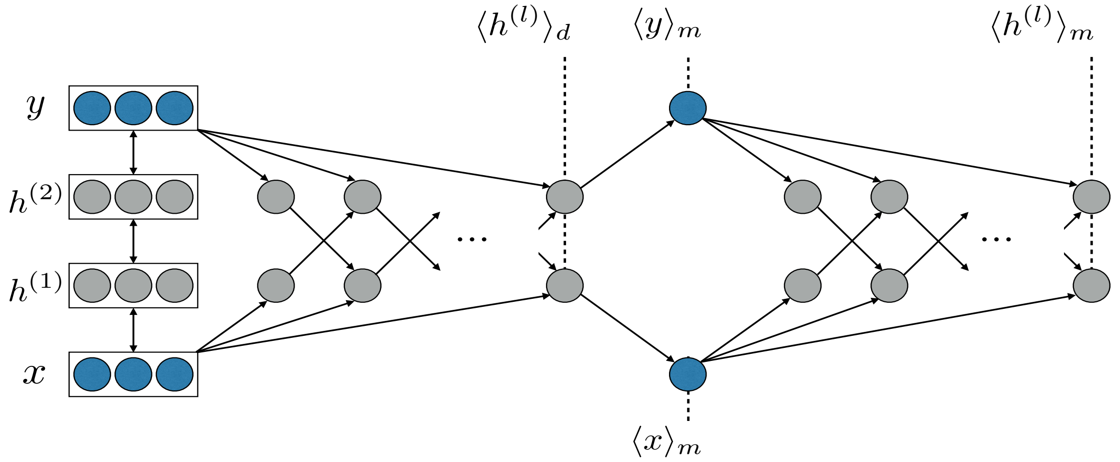


Figure 1: Calculating expectations using mean field update in the training of a DRM.

where $\boldsymbol{x} \in \mathbb{R}^I$ and $\boldsymbol{y} \in \mathbb{R}^K$ are the Gaussian variables in the first and second visible layers, $\boldsymbol{\sigma}^{(x)} \in \mathbb{R}^I$ and $\boldsymbol{\sigma}^{(y)} \in \mathbb{R}^K$ are the deviation parameters of the visible Gaussian units $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. The parameters $\boldsymbol{b}$, $\boldsymbol{c}^{(l)}$, $\boldsymbol{d}$, and $\boldsymbol{W}^{(l)}$ have the same definitions as those of a BBDRM. Each is the parameter to optimize in the training stage. Note that the binary operator $\circ$ and each division in the energy function denote element-wise product and division.

Under the definition of the energy function of a GGDRM, the conditional probabilities for each visible unit given the adjacent hidden units are computed as:

$$p(x_i = x|\boldsymbol{h}^{(1)}) = \mathcal{N}\Big(x|b_i + \boldsymbol{W}^{(1)}_{i:}\boldsymbol{h}^{(1)}, \sigma_i^{(x)2}\Big) \tag{9}$$

$$p(y_k = y|\boldsymbol{h}^{(L)}) = \mathcal{N}\Big(y|d_k + \boldsymbol{W}^{(L+1)T}_{:k}\boldsymbol{h}^{(L)}, \sigma_k^{(y)2}\Big), \tag{10}$$

where $\mathcal{N}(\cdot|\mu, \sigma^2)$ denotes the Gaussian probability density function with mean μ and variance σ^2. The conditional probabilities for hidden variables $\boldsymbol{h}^{(1)}$ and $\boldsymbol{h}^{(L)}$ given adjacent hidden units and visible units are computed as:

$$p(h_j^{(1)} = 1|\boldsymbol{x}, \boldsymbol{h}^{(2)}) = \sigma\Big(c_j^{(1)} + \boldsymbol{W}^{(1)T}_{:j}\frac{\boldsymbol{x}}{\boldsymbol{\sigma}^{(x)2}} + \boldsymbol{W}^{(2)}_{j:}\boldsymbol{h}^{(2)}\Big) \tag{11}$$

$$p(h_j^{(L)} = 1|\boldsymbol{y}, \boldsymbol{h}^{(L-1)}) = \sigma\Big(c_j^{(L)} + \boldsymbol{W}^{(L)T}_{:j}\boldsymbol{h}^{(L-1)} + \boldsymbol{W}^{(L+1)}_{j:}\frac{\boldsymbol{y}}{\boldsymbol{\sigma}^{(y)2}}\Big). \tag{12}$$

As with a BBDRM, the conditional probabilities for hidden variables at the 2nd, ..., $(L-1)$-th hidden layers are calculated as:

$$p(h_j^{(l)} = 1|\boldsymbol{h}^{(l-1)}, \boldsymbol{h}^{(l+1)}) = \sigma(c_j^{(l)} + \boldsymbol{W}^{(l)T}_{:j}\boldsymbol{h}^{(l-1)} + \boldsymbol{W}^{(l+1)}_{j:}\boldsymbol{h}^{(l+1)}). \tag{13}$$

In the same fashion as a BBDRM, the parameters $\boldsymbol{\theta} = \{\boldsymbol{b}, \boldsymbol{c}^{(l)}, \boldsymbol{d}, \boldsymbol{W}^{(1)}, \boldsymbol{W}^{(l)}, \boldsymbol{W}^{(L+1)}, \boldsymbol{\sigma}^{(x)}, \boldsymbol{\sigma}^{(y)}\}$ are estimated to maximize the joint log-likelihood $\mathcal{L}$ in the training stage of a GGDRM. The gradients for each parameter are computed as:

$$\frac{\partial \mathcal{L}}{\partial b_i} = \frac{1}{\sigma_i^{(x)2}}\Big(\langle x_i\rangle_d - \langle x_i\rangle_m\Big) \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial c_j^{(l)}} = \langle h_j^{(l)}\rangle_d - \langle h_j^{(l)}\rangle_m \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial d_k} = \frac{1}{\sigma_k^{(y)2}}\Big(\langle y_k\rangle_d - \langle y_k\rangle_m\Big) \tag{16}$$

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} = \begin{cases} \frac{1}{\sigma_i^{(x)2}}\Big(\langle x_i h_j^{(1)}\rangle_d - \langle x_i h_j^{(1)}\rangle_m\Big) & (l = 1) \\ \langle h_i^{(l-1)} h_j^{(l)}\rangle_d - \langle h_i^{(l-1)} h_j^{(l)}\rangle_m & (l = 2, ..., L) \\ \frac{1}{\sigma_k^{(y)2}}\Big(\langle h_i^{(L)} y_j\rangle_d - \langle h_i^{(L)} y_j\rangle_m\Big) & (l = L+1) \end{cases}. \tag{17}$$

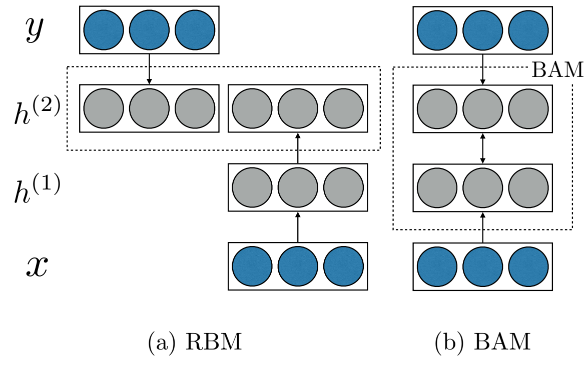
Additionally, we learn log-variances $z_i^{(x)} = \log\sigma_i^{(x)2}$ and $z_k^{(y)} = \log\sigma_k^{(y)2}$ to keep the variances positive following the training of an IGBRBM. Therefore, the gradients for log-variances are calculated as:

$$\frac{\partial \mathcal{L}}{\partial z_i^{(x)}} = e^{-z_i^{(x)}}\Bigg(\Big\langle\frac{1}{2}(x_i - b_i)^2 - x_i\boldsymbol{W}^{(1)}_{i:}\boldsymbol{h}^{(1)}\Big\rangle_d - \Big\langle\frac{1}{2}(x_i - b_i)^2 - x_i\boldsymbol{W}^{(1)}_{i:}\boldsymbol{h}^{(1)}\Big\rangle_m\Bigg) \tag{18}$$

$$\frac{\partial \mathcal{L}}{\partial z_k^{(y)}} = e^{-z_k^{(y)}}\Bigg(\Big\langle\frac{1}{2}(y_k - d_k)^2 - y_k\boldsymbol{W}^{(L+1)T}_{:k}\boldsymbol{h}^{(L)}\Big\rangle_d - \Big\langle\frac{1}{2}(y_k - d_k)^2 - y_k\boldsymbol{W}^{(L+1)T}_{:k}\boldsymbol{h}^{(L)}\Big\rangle_m\Bigg). \tag{19}$$

In the training stage of a GGDRM, each parameter is updated iteratively using Eqs. (14) to (19).

The expectations over the data distribution of the visible variables $\langle\boldsymbol{x}\rangle_d$ and $\langle\boldsymbol{y}\rangle_d$ are obtained by calculating the mean value of the observed data, and those of the hidden variables $\langle\boldsymbol{h}^{(l)}\rangle_d$ are obtained by iterative inference from observed visible variables using Eqs. (5), (11) and (12) (see Fig. 1). On



(a) RBM (b) BAM

Figure 2: Pre-training methods of using (a) only RBMs, and (b) RBMs and BAM.

the other hand, as with a BBDRM, the expectations over the model distributions $\langle\cdot\rangle_m$ are approximated by iterative inference to avoid the computational difficulties. The expectations over the model distribution of the visible variables $\langle\boldsymbol{x}\rangle_m$ and $\langle\boldsymbol{y}\rangle_m$ are calculated given $\langle\boldsymbol{h}^{(l)}\rangle_d$ using Eqs. (9) and (10), and $\langle\boldsymbol{h}^{(l)}\rangle_m$ is computed given $\langle\boldsymbol{x}\rangle_m$ and $\langle\boldsymbol{y}\rangle_m$ using Eqs. (5), (11) and (12).

3.2. Pre-training using BAM

As mentioned in Section 2, the greedy-wise training using RBMs is performed from the outer to the inner in the pre-training of a BBDRM. When RBMs train parameters between two hidden layers, pseudo visible variables for RBMs are the expected values of the hidden units inferred from previously trained RBMs. Therefore, two RBMs represent different hidden variables (see Fig. 2 (a)). To avoid this, we use a bidirectional associative memory (BAM) [16], which can optimize the weight parameters of connections between two visible layers, to pre-train a GGDRM (see Fig. 2 (b)).

The energy function of a BAM is defined as:

$$E_{BAM}(\boldsymbol{x},\boldsymbol{y}) = -\boldsymbol{b}^T\boldsymbol{x} - \boldsymbol{d}^T\boldsymbol{y} - \boldsymbol{x}^T\boldsymbol{W}\boldsymbol{y}, \tag{20}$$

where $\boldsymbol{b}$ and $\boldsymbol{d}$ are the biases corresponding to first and second visible variables, and $\boldsymbol{W}$ is the weight connections between two visible layers, respectively. Chen *et al.* [17] adopted the CD (Contrastive Divergence) algorithm to estimate the parameters of a BAM as with RBMs regarding BAM as the probability density function. The probability density function of a BAM is:

$$p(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{Z_{BAM}}\exp(-E_{BAM}(\boldsymbol{x},\boldsymbol{y})), \tag{21}$$

where $Z_{BAM} = \sum_{\boldsymbol{x},\boldsymbol{y}}\exp(-E_{BAM}(\boldsymbol{x},\boldsymbol{y}))$ is the partition function.

3.3. Voice Conversion Based on GGDRM

When a GGDRM is applied to voice conversion, acoustic features of source and target speakers are assigned to the first and second visible variables $\boldsymbol{x}$, $\boldsymbol{y}$. After the training stage of the GGDRM, the obtained parameters are set as the initial parameters of a DNN and fine-tuned using back propagation as a feedforward DNN. To construct the source-to-target converter, acoustic features of source and target speakers are assigned to the input and output variables. The biases $\boldsymbol{b}$, $\boldsymbol{c}^{(1)}$, …, $\boldsymbol{c}^{(L)}$, $\boldsymbol{d}$ and the weight connections $\boldsymbol{W}^{(1)}$, …, $\boldsymbol{W}^{(L+1)}$ of the GGDRM are assigned to the DNN from the input layer into the output layer. On the other hand, to construct the target-to-source converter, input and output variables are replaced. The biases $\boldsymbol{d}$, $\boldsymbol{c}^{(L)}$, …, $\boldsymbol{c}^{(1)}$, $\boldsymbol{b}$ and the transposed weight connections $\boldsymbol{W}^{(L+1)T}$, …, $\boldsymbol{W}^{(1)T}$ of the GGDRM are assigned to the DNN from the input layer into the output layer. Since a GGDRM models the joint distribution of source and target speakers, the same parameters are assigned to the DNNs in either case.

Table 1: Comparison of MCD [dB] obtained by each method. For example, f2m indicates female-to-male conversion.

	MCD [dB]			
	f2f	m2m	f2m	m2f
GMM	6.21	6.16	6.41	6.37
DNN	5.53	5.48	5.59	5.62
GGDRM	**5.43**	**5.36**	**5.48**	**5.41**

4. Experiments

In this section, we evaluate our GGDRM-based voice conversion method in objective and subjective experiments using the speech data provided in Voice Conversion Challenge 2018[1]. The experiments were conducted for HUB tasks using parallel data.

4.1. Experimental Conditions

We evaluate our method in objective and subjective experiments using the speech data provided in Voice Conversion Challenge 2018 from eight native English speakers. The speech data includes 81 parallel utterances from 4 female and 4 male speakers. We used 50 utterances as the training data for each experiment. The raw audio was upsampled from 22,050 Hz to 24 kHz sampling and transformed into 40 dimensional mel-cepstral coefficients [18] with dynamic features (deltas and delta-deltas) [19], which results in 120 dimensional features as acoustic features using WORLD[2]. Acoustic features are normalized to have zero-mean and unit-variance over the training data.

To evaluate the performances of our GGDRM-based method in each experiment, we compared it with the GMM- and DNN-based methods. The DNN and the GGDRM consist of 3 hidden layers including 600 hidden units. The weights of the DNN were initialized randomly, and those of the GGDRM were fine-tuned using back propagation the same as a DNN after the training stage of the GGDRM. Note that the GGDRM provides the same parameters as the initial parameters of the DNN in source-to-target and target-to-source conversion. The number of mixture components of the GMM was 64.

Fig. 3 plots the trajectories of 3-th mel-cepstral coefficients of natural speech and those generated by the DNN and the GGDRM-based systems. It can be seen that the both systems can generate reasonable acoustic features and our method reproduced natural speech better than the DNN-based system.

4.2. Objective Evaluation

First, we compared our GGDRM-based method to the conventional GMM- and DNN-based methods objectively. Mel-

[1] http://www.vc-challenge.org/

[2] http://www.kki.yamanashi.ac.jp/ mmorise/world/english/

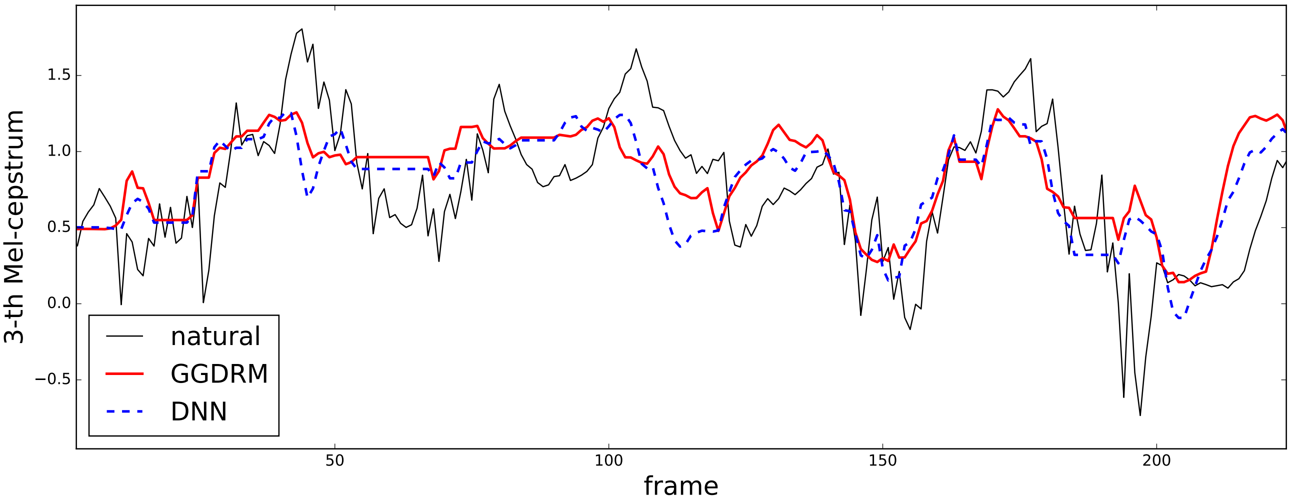


Figure 3: Trajectories of 3-th mel-cepstral coefficients of natural speech and those generated by the DNN-based and proposed GGDRM-based methods.

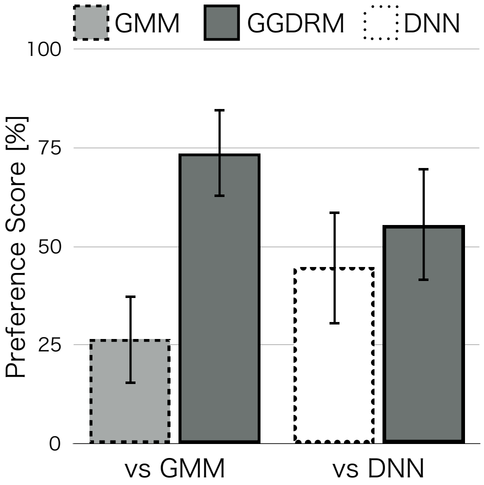


Figure 4: Subjective preference scores [%] for similarity of speech samples obtained by each method.

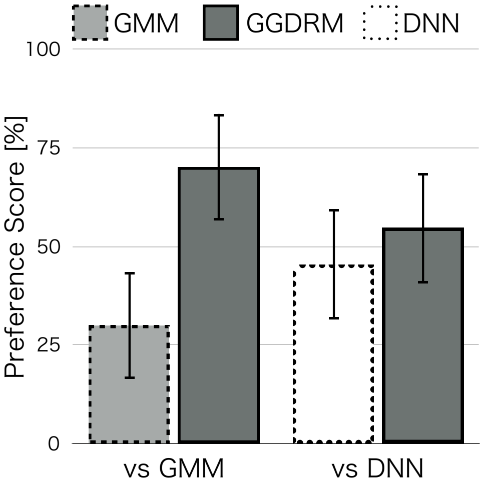


Figure 5: Subjective preference scores [%] for quality of speech samples obtained by each method.

cepstral coefficients for evaluation were generated from the acoustic features obtained from the models using the speech parameter generation algorithm [20]. The objective measure to evaluate the naturalness of the converted speech, mel-cepstral distortion (MCD) [7] (Eq. (22)), was used.

$$MCD = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{40}(mc_d^t - mc_d^e)^2}, \qquad (22)$$

where mc_d^t and mc_d^e denote d-th mel-cepstral coefficients of natural and generated speech at a frame, respectively. MCD indicates Euclidean distance between mel-cepstral coefficients of natural and generated speech.

As shown in Table 1, our GGDRM-based method performed the best in each pair. It is assumed that our method represents deep bidirectional representation between source and target speakers owing to considering not only feedforward dependencies from a source into a target but also backward dependencies from a target into a source using the GGDRM.

4.3. Subjective Evaluation

Second, we conducted listening tests to evaluate the performance of our method subjectively. In this experiment, 11 subjects each evaluated 10 pairs that were randomly chosen from 31 test utterances. Each pair was evaluated on the basis of 1) the similarity: which one is more like the natural speech of the target speaker, and 2) the quality: which one has better quality speech. Speech waveforms were synthesized from generated mel-cepstral coefficients. Fundamental frequencies F0 are linearly converted as:

$$\hat{y}_t = \frac{\rho^{(y)}}{\rho^{(x)}}(x_t - \mu^{(x)}) + \mu^{(y)}, \qquad (23)$$

where x_t and $\hat{y}_t$ are log-scaled F0 of source and converted utterances at the frame t, $\mu^{(x)}$ and $\rho^{(x)}$ are the mean and standard deviation of source utterances in the training data, $\mu^{(y)}$ and $\rho^{(y)}$ are those of target utterances, respectively.

Experimental results for the similarity are shown in Fig. 4. The error ranges in the figure indicate 95 % confidence intervals. It can be seen that our GGDRM-based method was preferred significantly to the GMM-based method. In comparison with DNN-based method, our method provided the same initial parameters as and performed comparably to the DNN-based method despite the DNN being constructed from only the feed-forward dependencies from source speakers to target speakers.

Fig. 5 shows that the experimental results for the quality. In quality comparisons, subjective preference scores similar to similarity comparisons were obtained. Therefore, our GGDRM-based method can be said that to be effective for voice conversion.

5. Conclusion

In this paper, we define a Gaussian-Gaussian deep relational model (GGDRM), which is the Gaussian-Gaussian form of the Bernoulli-Bernoulli DRM (BBDRM) and proposed a GGDRM-based voice conversion method. In the objective and subjective experiments, our GGDRM-based method outperformed conventional Gaussian mixture model (GMM)- and deep neural network (DNN)-based methods owing to its representation capability by multi-layer construction and the joint training. In the future, we will investigate its potential without the fine-tuning scheme.

6. Acknowledgements

This work was partially supported by the Telecommunications Advancement Foundation Grants, and by JST ACT-I.

7. References

[1] Alexander Kain and Michael W. Macon, "Design and evaluation of a voice conversion algorithm based on spectral envelope mapping and residual prediction," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001, pp. 813–816.

[2] Hélène Valbret, Eric Moulines, and Jean-Pierre Tubach, "Voice transformation using PSOLA technique," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1992, pp. 145–148.

[3] Li-Juan Liu, Ling-Hui Chen, Zhen-Hua Ling, and Li-Rong Dai, "Using bidirectional associative memories for joint spectral envelope modeling in voice conversion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 7884–7888.

[4] Masanobu Abe, Satoshi Nakamura, Kiyohiro Shikano, and Hisao Kuwabara, "Voice conversion through vector quantization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1988, pp. 655–658.

[5] Yannis Stylianou, Olivier Cappè, and Eric Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[6] Arthur R. Toth and Alan W Black, "Using articulatory position data in voice transformation," in *Proceedings of the ISCA Workshop on Speech Synthesis*, 2007, pp. 182–187.

[7] Srinivas Desai, E. Veera Raghavendra, B. Yegnanarayana, Alan W Black, and Kishore Prahallad, "Voice conversion using artificial neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3893–3896.

[8] Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki, "Voice conversion using RNN pre-trained recurrent temporal restricted boltzmann machines," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 580–587, 2015.

[9] Seyed H. Mohammadi and Alexander Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *Proceedings of the IEEE Spoken Language Technology Workshop*, 2014, pp. 19–23.

[10] Toru Nakashika, "Deep relational model: A joint probabilistic model with a hierarchical structure for bidirectional estimation of image and labels," *IEICE Transactions on Information and Systems*, vol. E101-D, no. 2, pp. 428–436, 2018.

[11] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] Ruslan Salakhutdinov and Geoffrey Hinton, "Deep Boltzmann machines," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.

[13] Ruslan Salakhutdinov and Hugo Larochelle, "Efficient learning of deep Boltzmann machines," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010, pp. 693–700.

[14] Geoffrey E. Hinton and Ruslan Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[15] KyungHyun Cho, Alexander Ilin, and Tapani Raiko, "Improved learning of Gaussian-Bernoulli restricted Boltzmann machines," in *Proceedings of the International Conference on Artificial Neural Networks*, 2011, pp. 10–17.

[16] Bart Kosko, "Bidirectional associative memories," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 49–60, 1988.

[17] Ling-Hui Chen, Tuomo Raitio, Cassia Valentini-Botinhao, Zhen-Hua Ling, and Junichi Yamagishi, "A deep generative architecture for postfiltering in statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 2003–2014, 2015.

[18] Toshiaki Fukuda, Keiichi Tokuda, Takao Kobayash, and Satoshi Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1992, pp. 137–140.

[19] Takashi Masuko, Keiichi Tokuda, Takao Kobayashi, and Satoshi Imai, "Speech synthesis from HMMs using dynamic features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1996, pp. 389–392.

[20] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009, pp. 1315–1318.