High-Order Sequence Modeling Using Speaker-Dependent Recurrent Temporal Restricted Boltzmann Machines for Voice Conversion

Toru Nakashika[1], *Tetsuya Takiguchi*[2], *Yasuo Ariki*[2]

[1]Graduate School of System Informatics, Kobe University, Japan
[2]Organization of Advanced Science and Technology, Kobe University, Japan
nakashika@me.cs.scitec.kobe-u.ac.jp, takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp

Abstract

This paper presents a voice conversion (VC) method that utilizes recently proposed recurrent temporal restricted Boltzmann machines (RTRBMs) for each speaker, with the goal of capturing high-order temporal dependencies in an acoustic sequence. Our algorithm starts from the separate training of two RTRBMs for a source and target speaker using speaker-dependent training data. Since each RTRBM attempts to discover abstractions at each time step, as well as the temporal dependencies in the training data, we expect that the models represent the speaker-specific latent features in the high-order spaces. In our approach, we run conversion from such speaker-specific-emphasized features of the source speaker to those of the target speaker using a neural network (NN), so that the entire network (the two RTRBMs ant the NN) forms a deep recurrent neural network and can be fine-tuned. Through VC experiments, we confirmed the high performance of our method especially in terms of objective criteria in comparison to conventional VC methods such as Gaussian mixture model (GMM)-based approaches.

Index Terms: voice conversion, recurrent temporal restricted Boltzmann machine, deep learning, recurrent neural network, speaker specific features

1. Introduction

In recent years, voice conversion (VC), a technique used to change specific information in the speech of a source speaker into that of a target speaker while retaining linguistic information, has been garnering much attention in speech signal processing. VC techniques have been applied to various tasks, such as speech enhancement [1], emotion conversion [2], speaking assistance [3], and other applications [4, 5]. Most of the related work in VC focuses not on f0 conversion but on the conversion of spectrum features, and we conform to that in this report as well.

Various statistical approaches to VC have been studied so far, for example those discussed in [6, 7]. Among these approaches, the Gaussian mixture model (GMM)-based mapping method [8] is widely used, and a number of improvements have been proposed. Toda *et al.* [9] introduced dynamic features and the global variance (GV) of the converted spectra over a time sequence. Helander *et al.* [10] proposed transforms based on Partial Least Squares (PLS) to prevent the over-fitting problem encountered in standard multivariate regression.

However, the GMM-based approaches rely on "shallow" voice conversion, a method based on piecewise-linear transformation. The shape of the vocal tract is generally non-linear, so non-linear voice conversion is more compatible with human speech. To capture the characteristics of speech more precisely, it is necessary to have a deeper non-linear architecture with more hidden layers. One example of deeper VC methods is proposed by Desai *et al.* [11] based on Neural Networks (NN). Nakashika *et al.* [12] also proposed a VC method using speaker-dependent restricted Boltzmann machines (RBMs) or deep belief networks (DBNs [13]) to achieve non-linear deep transformation. Chen *et al.* [14] models joint spectral distribution of a source and a target speaker using an RBM. Wu *et al.* [15] utilized a conditional restricted Boltzmann machine (CRBM [16]) to obtain latent non-linear relationships between the speech of a source and that of a target speaker. It was reported that these non-linear VC approaches achieved relatively higher performance than linear transformation approaches [11, 12, 15].

In this paper, we extend our earlier work in [12] to systematically capture time information as well as latent (deep) relationships between a source speaker's and a target speaker's features in a single network, accomplished by combining speaker-dependent recurrent temporal restricted Boltzmann machines (RTRBMs [17]) and a concatenating NN. An RTRBM is a non-linear probabilistic model used to capture temporal dependencies in time series data, similar to the before-mentioned CRBM and an RNN-RBM [18]. Despite its simplicity, this model does a good job at describing meaningful sequences such as video, music and speech. In our approach, we first train two exclusive RTRBMs for the source and the target speakers independently using segmented training data prepared for each speaker, then train a NN using the projected features, and finally fine-tune the networks as a single recurrent neural network. Because the training data for the source speaker RTRBM include various phonemes particular to the speaker, the speaker-dependent network tries to capture the abstractions to maximally express the training data that have abundant speaker individuality information and less phonological information. Furthermore, the network receives a collection of time-series feature vectors with the conditional models, enabling it to discover temporal correlations in the high-order space. Therefore, we expect that if feature conversion is conducted in such time-involving, individuality-emphasized, high-order spaces, it is much easier to convert voice features than if the original cepstrum-based space is used.

2. Models

Our voice conversion system uses recurrent temporal restricted Boltzmann machines (RTRBMs) to capture high-order conversion-friendly features. We briefly review the RTRBM and its fundamental model, the restricted Boltzmann machine (RBM), in this section.

2.1. RBM

An RBM was originally introduced as an undirected graphical model that defines the distribution of binary visible variables with binary hidden (latent) variables [19]. Later, this model was extended to deal with real-valued data, a so-called Gaussian-Bernoulli RBM (GBRBM) [13], and became a popular tool for representing complicated distributions of actual data, such as audio and images. In the literature of an improved GBRBM [20], the joint probability $p(\boldsymbol{v}, \boldsymbol{h})$ of real-valued visible units $\boldsymbol{v} = [v_1, \cdots, v_I]^T, v_i \in \mathbb{R}$ and binary-valued hidden units $\boldsymbol{h} = [h_1, \cdots, h_J]^T, h_j \in \{0, 1\}$ is defined with an energy function $E(\boldsymbol{v}, \boldsymbol{h})$ as follows:

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v}, \boldsymbol{h})} \tag{1}$$

$$E(\boldsymbol{v}, \boldsymbol{h}) = \left\| \frac{\boldsymbol{v} - \boldsymbol{b}}{2\boldsymbol{\sigma}} \right\|^2 - \boldsymbol{c}^T \boldsymbol{h} - \left(\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2} \right)^T \boldsymbol{W} \boldsymbol{h} \tag{2}$$

$$Z = \sum_{\boldsymbol{v}, \boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})}, \tag{3}$$

where $\| \cdot \|^2$ denotes L2 norm. $\boldsymbol{W} \in \mathbb{R}^{I \times J}$, $\boldsymbol{\sigma} \in \mathbb{R}^{I \times 1}$, $\boldsymbol{b} \in \mathbb{R}^{I \times 1}$, and $\boldsymbol{c} \in \mathbb{R}^{J \times 1}$ are model parameters of the GBRBM, indicating the weight matrix between visible units and hidden units, the standard deviations associated with Gaussian visible units, a bias vector of the visible units, and a bias vector of hidden units, respectively. The fraction bar in Eq. (2) denotes the element-wise division.

Because there are no connections between visible units or between hidden units, the conditional probabilities $p(\boldsymbol{h}|\boldsymbol{v})$ and $p(\boldsymbol{v}|\boldsymbol{h})$ form simple equations as follows:

$$p(h_j = 1|\boldsymbol{v}) = \mathcal{S}(c_j + \boldsymbol{W}_{:j}^T (\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2})) \tag{4}$$

$$p(v_i = v|\boldsymbol{h}) = \mathcal{N}(v \mid b_i + \boldsymbol{W}_{i:} \boldsymbol{h}, \sigma_i^2), \tag{5}$$

where $\boldsymbol{W}_{:j}$ and $\boldsymbol{W}_{i:}$ denote the jth column vector and the ith row vector, respectively. $\mathcal{S}(\cdot)$ and $\mathcal{N}(\cdot|\mu, \sigma^2)$ indicate an element-wise sigmoid function and Gaussian probability density function with the mean μ and variance σ^2.

For parameter estimation, the log-likelihood of a collection of visible units $\mathcal{L} = \log \prod_n p(\boldsymbol{v}_n)$ is used as an evaluation function. Differentiating partially with respect to each parameter, we obtain:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}_{ij}} = \langle \frac{v_i h_j}{\sigma_i^2} \rangle_{data} - \langle \frac{v_i h_j}{\sigma_i^2} \rangle_{model} \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial b_i} = \langle \frac{v_i}{\sigma_i^2} \rangle_{data} - \langle \frac{v_i}{\sigma_i^2} \rangle_{model} \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial c_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model}, \tag{8}$$

where $\langle \cdot \rangle_{data}$ and $\langle \cdot \rangle_{model}$ indicate expectations of input data and the inner model, respectively. However, it is generally difficult to compute the second term, so typically, expectation of the reconstructed data $\langle \cdot \rangle_{recon}$ computed by Eqs. (4) and (5) is alternatively used [13]. Using Eqs. (6), (7), and (8), each parameter can be updated using stochastic gradient descent.

2.2. RTRBM

An RTRBM is an extended version of RBM proposed by Sutskever *et al.* [17], and is suitable for capturing and modeling temporal dependencies in sequence data. In addition to the use of an undirected model as in an RBM, RTRBM also employs directed models from previous hidden units $\boldsymbol{h}^{(t-1)} = [h_1^{(t-1)}, \cdots, h_J^{(t-1)}]^T, h_j^{(t-1)} \in \{0, 1\}$ to the current hidden units $\boldsymbol{h}^{(t)} = [h_1^{(t)}, \cdots, h_J^{(t)}]^T, h_j^{(t)} \in \{0, 1\}$ and the current visible units $\boldsymbol{v}^{(t)} = [v_1^{(t)}, \cdots, v_I^{(t)}]^T, v_i^{(t)} \in \mathbb{R}$ at the current frame t. In this model, there are three types of parameters to be estimated: $\boldsymbol{B} \in \mathbb{R}^{I \times J}$ (a directed weight matrix from $\boldsymbol{h}^{(t-1)}$ to $\boldsymbol{v}^{(t)}$), $\boldsymbol{C} \in \mathbb{R}^{J \times J}$ (a directed weight matrix from $\boldsymbol{h}^{(t-1)}$ to $\boldsymbol{h}^{(t)}$), and $\boldsymbol{W} \in \mathbb{R}^{I \times J}$ (an undirected weight matrix between $\boldsymbol{v}^{(t)}$ and $\boldsymbol{h}^{(t)}$). These weights are estimated using contrastive divergence in a similar manner to RBM by maximizing the log-likelihood $\mathcal{L} = \log \prod_t p(\boldsymbol{v}^{(t)}|\mathcal{A}^{(t)})$ denoted by $\mathcal{A}^{(t)} = \{\boldsymbol{v}^{(\tau)}, \boldsymbol{h}^{(\tau)} | \tau < t\}$, where

$$p(\boldsymbol{v}^{(t)}|\mathcal{A}^{(t)}) = \frac{1}{Z} \sum_{\boldsymbol{h}^{(t)}} e^{-E(\boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t)} | \boldsymbol{h}^{(t-1)})}. \tag{9}$$

In our RTRBM model, the energy function E becomes:

$$E(\boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t)} | \boldsymbol{h}^{(t-1)}) = \left\| \frac{\boldsymbol{v}^{(t)} - \boldsymbol{b}^{(t)}}{2\boldsymbol{\sigma}} \right\|^2 - \boldsymbol{c}^{(t)^T} \boldsymbol{h}^{(t)} - \left(\frac{\boldsymbol{v}^{(t)}}{\boldsymbol{\sigma}^2} \right)^T \boldsymbol{W} \boldsymbol{h}^{(t)} \tag{10}$$

$$\boldsymbol{b}^{(t)} = \boldsymbol{b} + \boldsymbol{B} \boldsymbol{h}^{(t-1)} \tag{11}$$

$$\boldsymbol{c}^{(t)} = \boldsymbol{c} + \boldsymbol{C} \boldsymbol{h}^{(t-1)}. \tag{12}$$

The previous hidden units $\boldsymbol{h}^{(t-1)}$ in Eqs. (11) and (12) are replaced with the mean-field values $\hat{\boldsymbol{h}}^{(t-1)}$ as follows:

$$\hat{\boldsymbol{h}}^{(t-1)} = \mathcal{S}(\boldsymbol{c}^{(t-1)} + \boldsymbol{W}^T (\frac{\boldsymbol{v}^{(t-1)}}{\boldsymbol{\sigma}^2})) \tag{13}$$

since this approach improves the efficiency of training [17]. For the initial values $\boldsymbol{h}^{(0)}$, we use a zero vector in this paper.

We obtain the following partial differential equations to the log-likelihood:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{B}_{ij}} = \langle \frac{v_i^{(t)} \hat{h}_j^{(t-1)}}{\sigma_i^2} \rangle_{data} - \langle \frac{v_i^{(t)} \hat{h}_j^{(t-1)}}{\sigma_i^2} \rangle_{model} \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{C}_{j'j}} = \langle \hat{h}_{j'}^{(t-1)} \hat{h}_j^{(t)} \rangle_{data} - \langle \hat{h}_{j'}^{(t-1)} \hat{h}_j^{(t)} \rangle_{model}. \tag{15}$$

The other parameters related to the undirected model ($\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{c}$) are also calculated from equations (6), (7) and (8) by proper substitution of variables. The second terms in Eqs. (14) and (15) are computed as the reconstructed values similarly to RBM.

Once the parameters are estimated, forward inference (the conditional probability of $\boldsymbol{h}^{(t)}$ given $\boldsymbol{v}^{(t)}$ and $\boldsymbol{h}^{(t-1)}$) and backward inference (the conditional probability of $\boldsymbol{v}^{(t)}$ given $\boldsymbol{h}^{(t)}$ and $\boldsymbol{h}^{(t-1)}$) are respectively written as:

$$p(h_j^{(t)} = 1|\boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t-1)}) = \mathcal{S}(c_j^{(t)} + \boldsymbol{W}_{:j}^T (\frac{\boldsymbol{v}^{(t)}}{\boldsymbol{\sigma}^2})) \tag{16}$$

$$p(v_i^{(t)} = v|\boldsymbol{h}^{(t)}, \boldsymbol{h}^{(t-1)}) = \mathcal{N}(v|b_i^{(t)} + \boldsymbol{W}_{i:} \boldsymbol{h}^{(t)}, \sigma_i^2). \tag{17}$$

3. Voice conversion using SD-RTRBMs

In general, the less phonological and the more individuality-emphasized features a source input includes for a speaker, the easier it is to convert the source features to target features. This paper proposes a voice conversion method using such features obtained from speaker-dependent restricted Boltzmann machines (SD-RTRBMs).

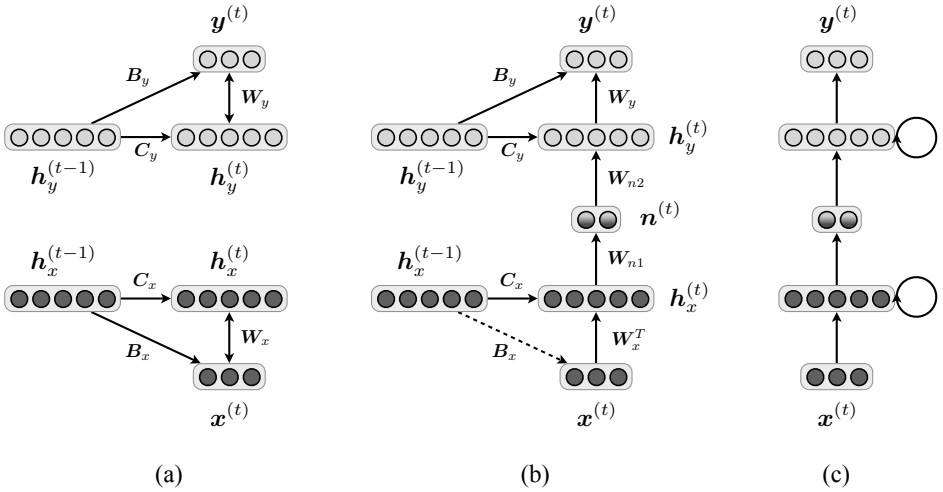

Figure 1: (a) RTRBMs for a source speaker (below) and a target speaker (above), (b) our proposed voice conversion architecture combining two speaker-dependent RTRBMs with a NN, (c) an alternative representation of (b) that can be regarded as a recurrent neural network.

Figure 1 shows an overview of our proposed voice conversion system. In our approach, we independently train RTRBMs for each speaker beforehand as shown in Figure 1(a). Variables $\boldsymbol{x}^{(t)}$ and $\boldsymbol{y}^{(t)}$ ($\boldsymbol{x}^{(t-1)}$ and $\boldsymbol{y}^{(t-1)}$) are acoustic feature vectors (e.g., visible units in RTRBM), such as MFCC, at frame t (at frame $t-1$) for a source speaker (and a target speaker).

For the source speaker, for instance, the parameter matrices $\boldsymbol{W}_x$, $\boldsymbol{B}_x$, and $\boldsymbol{C}_x$ are estimated so as to maximize the probability of a T-time sequence $p(\boldsymbol{x}) = \prod_t^T p(\boldsymbol{x}^{(t)}|\mathcal{A}^{(t)})$. Because each unit in the hidden vector $\boldsymbol{h}_x^{(t)}$ is independent from the others, it captures the *common* characteristics in the visible units. The training data usually include various phonemes and unvarying speaker-specific features; thus, we expect that the extracted features in $\boldsymbol{h}_x^{(t)}$ emphasize speaker-individual information. Furthermore, since we estimate the time-related matrices $\boldsymbol{B}_x$and $\boldsymbol{C}_x$ jointly with the static term $\boldsymbol{W}_x$ as shown in Eq. (10) using the training data, the matrices try to capture time-related information. This means that the obtained features in the hidden units $\boldsymbol{h}_x^{(t)}$ also help to capture time-related speaker-individualities. An input vector $\boldsymbol{x}^{(t)}$ at frame t is projected into such the speaker-dependent latent space that captures speaker-individualities. In this paper, the latent features $\boldsymbol{h}_x^{(t)}$ are obtained using mean-field approximation as in Eq. (16). The above discussion applies to the target speaker, and the hidden vector for the target $\boldsymbol{h}_y^{(t)}$ is obtained in the same manner. In our approach, we convert such individuality-emphasized features (from $\boldsymbol{h}_x^{(t)}$ to $\boldsymbol{h}_y^{(t)}$) using a neural network (NN) that has $L+2$ layers (L is the number of hidden layers; typically, L is 0 or 1) as shown in Figure 1(b). To train the NN, we use the parallel training set $\{\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}\}_{t=0}^{T'}$ where T' is the number of frames of the parallel data[1]. During the training stage of the NN, the projected vectors of the source speaker's acoustic features $\boldsymbol{h}_x^{(t)}$ are the inputs, and the projected vectors of the corresponding target speaker's features $\boldsymbol{h}_y^{(t)}$ are outputs. Weight parameters of the NN $\{\boldsymbol{W}_l, \boldsymbol{d}_l\}_{l=0}^{L}$ are estimated to minimize the error between the output $\eta(\boldsymbol{h}_x^{(t)})$ and the target vector $\boldsymbol{h}_y^{(t)}$ as is typical for a NN. Once the weight parameters are estimated, an input vector $\boldsymbol{h}_x^{(t)}$ is converted as follows:

$$\eta(\boldsymbol{h}_x^{(t)}) = \bigodot_{l=0}^{L} \eta_l(\boldsymbol{h}_x^{(t)}) \tag{18}$$

$$\eta_l(\boldsymbol{h}_x^{(t)}) = \mathcal{S}(\boldsymbol{W}_l \boldsymbol{h}_x^{(t)} + \boldsymbol{d}_l) \tag{19}$$

where $\bigodot_{l=0}^{L}$ denotes the composition of $L+1$ functions. For instance, $\bigodot_{l=0}^{1} \eta_l(\boldsymbol{z}) = \mathcal{S}(\boldsymbol{W}_1 \mathcal{S}(\boldsymbol{W}_0 \boldsymbol{z} + \boldsymbol{d}_0) + \boldsymbol{d}_1)$ for a NN with one hidden layer. To convert the output of the NN to the acoustic features of the target speaker, we simply use backward inference of an RTRBM using Eq. (17).

Summarizing the above discussion, a voice conversion function of our method from a source acoustic vector $\boldsymbol{x}^{(t)}$ to a target vector $\boldsymbol{y}^{(t)}$ at frame t is written as:

$$\boldsymbol{y}^{(t)} = \underset{\boldsymbol{y}^{(t)}}{\operatorname{argmax}}\, p(\boldsymbol{y}^{(t)}|\boldsymbol{x}^{(t)}, \boldsymbol{h}_x^{(t-1)}, \boldsymbol{h}_y^{(t-1)}) \tag{20}$$

$$= \boldsymbol{a}_{L+2}^{(t)} + \boldsymbol{W}_{L+2} \bigodot_{k=0}^{L+1} \mathcal{S}(\boldsymbol{a}_k^{(t)} + \boldsymbol{W}_k \boldsymbol{x}^{(t)}) \tag{21}$$

where $\boldsymbol{a}_k^{(t)}$ and $\boldsymbol{W}_k$ denote elements of a set of dynamic parameters $\Theta^{(t)} = \{\boldsymbol{a}^{(t),\boldsymbol{W}}\}$:

$$\boldsymbol{a}^{(t)} = \{\boldsymbol{a}_k^{(t)}\}_{k=0}^{L+2} = \{\boldsymbol{c}_x^{(t)}, \boldsymbol{d}_0, \cdots, \boldsymbol{d}_L, \boldsymbol{b}_y^{(t)}\} \tag{22}$$

$$\boldsymbol{W} = \{\boldsymbol{W}_k\}_{k=0}^{L+2} = \{\boldsymbol{W}_x^T, \boldsymbol{W}_0, \cdots, \boldsymbol{W}_L, \boldsymbol{W}_y\}, \tag{23}$$

where $\boldsymbol{c}_x^{(t)}$ and $\boldsymbol{b}_y^{(t)}$ are a forward-inference bias vector in a source speaker's RTRBM and a backward-inference bias vector in the target speaker's RTRBM obtained from Eqs. (12) and (11), respectively. $\boldsymbol{h}_x^{(0)}$ and $\boldsymbol{h}_y^{(0)}$ are zero vectors.

The conversion function shown in Eq. (21) implies an $(L+4)$-layer recurrent neural network with sigmoid activated functions as shown in Figure 1(c). Therefore, we can fine-tune each parameter of the entire network consisting of the two RTRBMs and the NN by back-propagation using the acoustic parallel data. Eq. (21) also shows that our VC method is based on the composite function of multiple different non-linear functions considering time-series data. Therefore, it is expected that our composite model can represent more complex relationships than the conventional GMM-based method and other static network approaches [11, 12] do.

[1]For sake of simplicity, we used the same parallel data for both training of the RTRBMs and the NN in our experiments ($T' = T$).

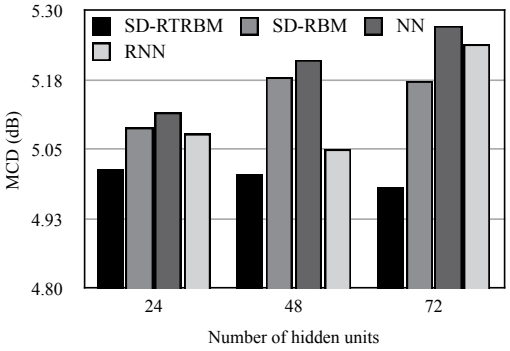

Figure 2: Averaged MCD with changing the architectures.

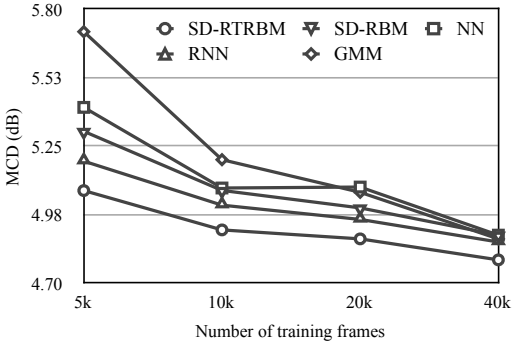

Figure 3: Averaged MCD for each method.

4. Experiments

4.1. Conditions

In our voice conversion experiments, we compared our method ("SD-RTRBM") with three conventional methods: the well-known GMM-based approach ("GMM"), the NN-based approach ("NN") and our previous work [12] that utilized speaker-dependent RBMs for the pre-training of the NN ("SD-RBM"). In [12], deeper architectures using DBNs were reported, but we used a single-layer DBN (i.e., an RBM) for each speaker for a comparison with our method. All of the network-based methods (RTRBMs, a NN, and RBMs) contained four layers ($L = 0$) with various numbers of hidden units as discussed in the following section. We trained the network-based methods with a learning rate of 0.01 and momentum of 0.9, with the number of epochs being 400, using acoustic features from the ATR Japanese speech database [21]. The parameters of our method and "SD-RBM" were fine-tuned after the training of the RTRBMs and RBMs, respectively. From the database, we selected a male speaker (identified with "MMY" in the database) for the source, and a female speaker ("FTK") for the target. 24-dimensional MFCC features were used as an input vector, calculated from STRAIGHT spectra [22] using filter-theory [23] to decode the MFCC back to STRAIGHT spectra in the synthesis stage. The parallel data of the source/target speakers processed by Dynamic Programming were created from 216 word utterances in the dataset, and were used for training. Note that the parallel data were prepared for the NN and GMM methods, and two speaker-wise RTRBMs were trained independently. For the objective test, 15 sentences that were not included in the training data were arbitrarily selected from the database. For the objective evaluation, we used MCD (mel-cepstral distortion) to measure how close the converted vector is to the target vector in mel-cepstral space.

4.2. Determination of hyper parameters

Determining the number of hidden units in the network-based approaches and the number of mixtures in the GMM-based approach is important for a fair comparison. We also compared with a recurrent neural network ("RNN") whose parameters were randomly initialized with the same architecture as our method for a reference. In the first experiments, we changed the number of hidden units for the network-based approaches as 24, 48, and 72, trained each method using $T = 20,000$ frames, and checked the performance of each method using a development set that contains five sentences different from the test set.

Table 1: Averaged MOS w.r.t. similarity for each method.

SD-RTRBM	SD-RBM	NN	GMM
2.86	2.80	2.77	2.14

Each network-based method has a four-layer architecture; for example, the 48-unit "NN" has 24, 48, 48, and 24 units from the input layer to the output layer. Figure 2 depicts the averaged MCD obtained from each method, showing that the wider architecture (such as "72") does not always provide better results than narrower architectures expect for our method. For the GMM-based approach, we tested GMMs with 8, 16, 32, 64, and 128 mixtures and obtained the best performance from 64 mixtures. The best architectures for each method were used in the remaining experiments (24 units for "SD-RBM" and "NN", 48 units for "RNN", 72 units for "SD-RTRBM", and 64 mixtures for "GMM").

4.3. Results and discussion

Figure 3 and Table 1 summarize the experimental results, comparing each method with respect to objective and subjective criteria, respectively. Figure 3 also shows the "RNN" results for reference. For the subjective evaluation, MOS (mean opinion score) listening tests were conducted, where 7 participants listened to pairs of an original target speech signal (generated from analysis-by-synthesis) and the converted speech signals for each method, and then selected how close the converted speech sounded to the original speech on a 5-point scale (5: excellent, 4: good, 3: fair, 2: poor, and 1: bad). As shown in Figure 3 and Table 1, our approach outperformed other conventional methods in both criteria. The reason for the improvement is attributed to the fact that our time-involving high-order conversion system using RTRBMs is able to capture and convert the abstractions of speaker individualities better than the other methods. In particular, as shown in Figure 3, our approach achieved high performance in MCD criteria. This is because the RTRBMs modeled and captured sequence data more appropriately than the other methods and alleviated estimation errors.

5. Conclusions

In this paper, we presented a voice conversion method that combines speaker-dependent RTRBMs and a NN to extract time-involving speaker-individual information from sequence data. Through experiments, we confirmed that our approach is effective especially in terms of MCD.

6. References

[1] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998, pp. 285–288.

[2] C. Veaux and X. Robet, "Intonation conversion from neutral to expressive speech," in *Proc. Interspeech*, 2011, pp. 2765–2768.

[3] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech Communication*, vol. 54, no. 1, pp. 134–146, 2012.

[4] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, pp. 301–304.

[5] A. Kunikoshi, Y. Qiao, N. Minematsu, and K. Hirose, "Speech generation from hand gestures based on space mapping," in *Proc. Interspeech*, 2009, pp. 308–311.

[6] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, 1984.

[7] H. Valbret, E. Moulines, and J.-P. Tubach, "Voice transformation using PSOLA technique," *Speech Communication*, vol. 11, no. 2, pp. 175–187, 1992.

[8] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[9] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[10] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 912–921, 2010.

[11] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2009, pp. 3893–3896.

[12] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets," in *Proc. Interspeech*, 2013, pp. 369–372.

[13] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[14] L.-H. Chen, Z.-H. Ling, Y. Song, and L.-R. Dai, "Joint spectral distribution modeling using restricted boltzmann machines for voice conversion," in *Proc. Interspeech*, 2013, pp. 3052–3056.

[15] Z. Wu, E. S. Chng, and H. Li, "Conditional restricted boltzmann machine for voice conversion," in *IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, 2013.

[16] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Advances in neural information processing systems*, 2006, pp. 1345–1352.

[17] I. Sutskever, G. Hinton, and G. Taylor, "The recurrent temporal restricted boltzmann machine." in *NIPS*, vol. 19, 2008, pp. 1601–1608.

[18] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *International Conference on Machine Learning*, 2012.

[19] Y. Freund and D. Haussler, *Unsupervised learning of distributions of binary vectors using two layer networks*. Computer Research Laboratory, 1994.

[20] K. Cho, A. Ilin, and T. Raiko, "Improved learning of gaussian-bernoulli restricted boltzmann machines," in *Artificial Neural Networks and Machine Learning*, 2011, pp. 10–17.

[21] 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, 1990.

[22] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, f0, and aperiodicity estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2008, pp. 3933–3936.

[23] B. Milner and X. Shao, "Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model," in *Proc. Interspeech*, 2002, pp. 2421–2424.