PARALLEL-DATA-FREE, MANY-TO-MANY VOICE CONVERSION USING AN ADAPTIVE RESTRICTED BOLTZMANN MACHINE

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

[1]Graduate School of Information Systems, The University of Electro-Communications, Japan
[2]Organization of Advanced Science and Technology, Kobe University, Japan
nakashika@uec.ac.jp, takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp

ABSTRACT

Voice conversion (VC) is a technique where only speaker-specific information in source speech is converted while preserving the associated phonological information. Most of the existing VC methods rely on using parallel data—pairs of speech data from the source and target speakers uttering the same sentences—when training the models. However, the use of parallel data causes several problems; firstly, the data used for the training is limited to the pre-defined sentences. Secondly, the trained model is only applied to the speaker pair used in the training. In this paper, we propose a novel probabilistic model called an adaptive restricted Boltzmann machine (ARBM) for VC between arbitrary speakers without the need to use parallel data. An ARBM models a joint distribution of visible units (set as acoustic features), hidden units, and speaker-identity units with the speaker-dependent connections between the visible and hidden units. The visible-hidden connections are defined as the product of the speaker-independent matrix and speaker-adaptive matrices so that the speech signal can be decomposed into speaker-specific information and the remaining information (that is, phonological information). Voice conversion using an ARBM is achieved by switching the speaker-specific information of a source speaker into that of a target speaker with the phonological information unchanged.

Index Terms— Voice conversion, restricted Boltzmann machine, speaker adaptation, non-parallel training, many-to-many conversion

1. INTRODUCTION

In recent years, voice conversion (VC), which is a technique used to change speaker-specific information in the speech of a source speaker into that of a target speaker while retaining linguistic information, has been garnering much attention since the VC techniques can be applied to various tasks [1, 2, 3, 4, 5]. Various statistical approaches to VC have been studied so far as discussed in [6, 7]. Among these approaches, the Gaussian mixture model (GMM)-based mapping method [8] is most widely used, and a number of improvements have been proposed [9, 10, 11]. Other VC methods, such as approaches based on non-negative matrix factorization (NMF) [12, 13], neural networks (NNs) [14], restricted Boltzmann machines (RBMs) [15, 16], and deep learning [17, 18], have been also proposed.

However, the above-mentioned approaches require parallel data (aligned speech data from the source and the target speakers so that each frame of the source speaker's data corresponds to that of the target speaker) for training the models, which leads to several problems. First, the data is limited to pre-defined articles (both speakers must utter the same articles). Second, the trained model is only applied to the speaker pair used in the training, and it is difficult to reuse the model on the conversion of another speaker pair. Third, the training data (the parallel data) is not the original speech data anymore because the speech data is stretched and modified in the time axis when aligned, and it is not guaranteed that each frame is aligned perfectly.

Several other approaches have been proposed that do not use (or use minimally) parallel data of the source and the target speakers [19, 20, 21, 22]. In [19], for example, they model the spectral relationships between two arbitrary speakers (reference speakers) using GMMs, and convert the source speaker's speech using the matrix that projects the feature space of the source speaker into that of the target speaker through that of reference speakers. As a result, parallel data from the source and target speakers is not required (however, they use the parallel data of reference speakers). In [21], they first obtained codebooks (eigenvoice) using the parallel data of reference speakers, and achieved many-to-many VC by mapping the source speaker's speech into eigenvoice and the eigenvoice into target speaker's speech (however, this approach also needs parallel data among reference speakers when creating the eigenvoice).

In this paper, we propose a totally-parallel-data-free[1] VC method using a novel energy-based probabilistic model, which we call an "adaptive restricted Boltzmann machine" (ARBM). An ARBM is aimed at extracting latent, phonolog-

[1] It means that the method requires neither the parallel data of a source speaker and target speaker, nor the parallel data of reference speakers.

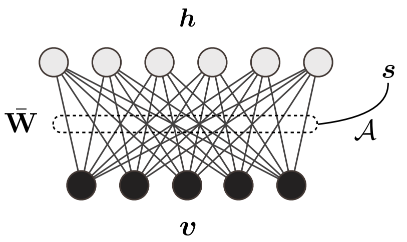


Fig. 1. Graphical representation of an ARBM.

ical features from the speech data uttered by several speakers while separating speaker-dependent information and speaker-independent information. This model consists of a visible layer and a hidden layer having connections between visible-hidden units like an RBM, but the weights of the connections vary with the speaker. Furthermore, we define the weights as a product of speaker-independent and speaker-dependent weight matrices, and these weights can be simultaneously optimized so as to maximize the likelihood of speech data that contains multiple speakers (not required to be parallel data). Many-to-many VC using an ARBM is conducted by replacing the speaker-dependent features of a source speaker with those of a target speaker while retaining the speaker-independent features.

2. ADAPTIVE RESTRICTED BOLTZMANN MACHINE

We define a graphical, probabilistic model called an adaptive restricted Boltzmann machine (ARBM) as shown in Figure 1. In addition to visible units $\boldsymbol{v} \in \mathbb{R}^I$ and hidden units $\boldsymbol{h} \in \{0,1\}^J$ appearing in conventional RBMs [23], we introduce identity units $\boldsymbol{s} \in \{0,1\}^K$ that represent which speaker utters the sentence (I, J, and K indicate the numbers of visible units, hidden units, and identity units, respectively). For example, the expression $s_k = 1, \forall s_{k'} = 0 \ (k' \neq k)$ means that the input vector $\boldsymbol{v}$ belongs to the kth speaker. In this model, connections exist that are controlled by $\boldsymbol{s}$ between visible units and hidden units. We define a connection-weight matrix $\mathbf{W}(\boldsymbol{s})$ as follows:

$$\mathbf{W}(\boldsymbol{s}) = \mathcal{A} \otimes_3 \boldsymbol{s}\bar{\mathbf{W}} + \mathcal{B} \otimes_3 \boldsymbol{s}, \tag{1}$$

where $\bar{\mathbf{W}} \in \mathbb{R}^{I \times J}$ is a speaker-independent weight matrix, and third-order tensors $\mathcal{A} \in \mathbb{R}^{I \times I \times K}$ and $\mathcal{B} \in \mathbb{R}^{I \times J \times K}$ adapt $\bar{\mathbf{W}}$ for the specific speaker (the kth matrices $\mathbf{A}_{:,:,k}$ and $\mathbf{B}_{:,:,k}$ of the tensors $\mathcal{A}$ and $\mathcal{B}$ indicate an adaptive matrix and a bias matrix for the kth speaker, respectively). $\mathcal{X} \otimes_d \boldsymbol{y}$ indicates an operator that takes an inner product of a third-order tensor $\mathcal{X}$ unfolded along with the dth mode and a vector $\boldsymbol{y}$ (i.e., $\mathcal{X} \otimes_d \boldsymbol{y} = \sum_k y_k \mathbf{X}_k$ where $\mathbf{X}_k$ are mode-d unfolded matrices of $\mathcal{X}$).

Using the matrix $\mathbf{W}(\boldsymbol{s})$ defined in Eq. (1), the joint probability $p(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s})$ is defined as follows:

$$p(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s}) = \frac{1}{Z} e^{-E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s})} \tag{2}$$

$$E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s}) = \left\| \frac{\boldsymbol{v} - \boldsymbol{b}}{2\boldsymbol{\sigma}} \right\|^2 - \boldsymbol{c}^{\mathrm{T}} \boldsymbol{h} - \left(\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2} \right)^{\mathrm{T}} \mathbf{W}(\boldsymbol{s}) \boldsymbol{h} \tag{3}$$

$$Z = \sum_{\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s}} e^{-E(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{s})}, \tag{4}$$

where $\|\cdot\|^2$ denotes L2 norm. $\boldsymbol{\sigma} \in \mathbb{R}^I$, $\boldsymbol{b} \in \mathbb{R}^I$, and $\boldsymbol{c} \in \mathbb{R}^J$ are other parameters of the ARBMs, indicating the standard deviations associated with the Gaussian visible units, a bias vector of the visible units, and a bias vector of the hidden units, respectively. The fraction bar in Eq. (3) denotes the element-wise division.

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

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

$$p(h_j = 1|\boldsymbol{v}, \boldsymbol{s}) = \mathcal{S}(c_j + \mathbf{W}(\boldsymbol{s})_{:,j}^{\mathrm{T}}(\frac{\boldsymbol{v}}{\boldsymbol{\sigma}^2})), \tag{6}$$

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

As for parameter estimation, the parameters of ARBMs $\boldsymbol{\Theta} = \{\bar{\mathbf{W}}, \mathcal{A}, \mathcal{B}, \boldsymbol{b}, \boldsymbol{\sigma}, \boldsymbol{c}\}$ can be simultaneously optimized so as to maximize the log-likelihood $\mathcal{L}(\boldsymbol{\Theta})$ using N training data $(\{\boldsymbol{v}_n, \boldsymbol{s}_n\}_{n=1}^N)$:

$$\mathcal{L}(\boldsymbol{\Theta}) = \log \prod_n p(\boldsymbol{v}_n, \boldsymbol{s}_n) = \sum_n \log \sum_{\boldsymbol{h}} p(\boldsymbol{v}_n, \boldsymbol{h}, \boldsymbol{s}_n). \tag{7}$$

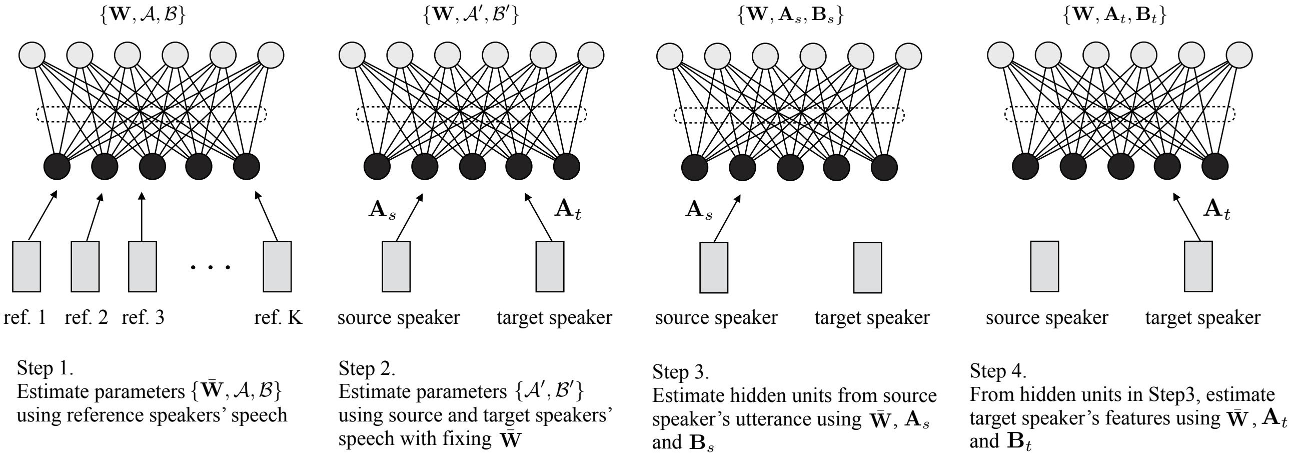
Differentiating partially with respect to each parameter, we obtain:

$$\frac{\partial \mathcal{L}}{\partial \bar{W}_{i,j}} = \langle \sum_{i',k} \frac{A_{i',i,k} v_{i'} h_j s_k}{\sigma_{i'}^2} \rangle_{\mathrm{data}} - \langle \sum_{i',k} \frac{A_{i',i,k} v_{i'} h_j s_k}{\sigma_{i'}^2} \rangle_{\mathrm{model}} \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial A_{i',i,k}} = \langle \sum_{j} \frac{\bar{W}_{i,j} v_{i'} h_j s_k}{\sigma_{i'}^2} \rangle_{\mathrm{data}} - \langle \sum_{j} \frac{\bar{W}_{i,j} v_{i'} h_j s_k}{\sigma_{i'}^2} \rangle_{\mathrm{model}} \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial B_{i,j,k}} = \langle v_i h_j s_k \rangle_{\mathrm{data}} - \langle v_i h_j s_k \rangle_{\mathrm{model}}, \tag{10}$$

where $\langle \cdot \rangle_{\mathrm{data}}$ and $\langle \cdot \rangle_{\mathrm{model}}$ indicate expectations of the training data and the inner model, respectively. The update rules for the other parameters $\boldsymbol{b}$, $\boldsymbol{\sigma}$, and $\boldsymbol{c}$ are the same as in [23]. It is generally difficult to compute the expectations of the inner model $\langle \cdot \rangle_{\mathrm{model}}$ in Eqs. (8), (9), and (10); however, we can still use contrastive divergence [24] and efficiently approximate them with the expectations of the reconstructed data $\langle \cdot \rangle_{\mathrm{recon.}}$. Using the gradients in Eqs. (8), (9), and (10), each parameter can be updated using stochastic gradient descent.


$\{\mathbf{W}, \mathcal{A}, \mathcal{B}\}$
$\{\mathbf{W}, \mathcal{A}', \mathcal{B}'\}$
$\{\mathbf{W}, \mathbf{A}_s, \mathbf{B}_s\}$
$\{\mathbf{W}, \mathbf{A}_t, \mathbf{B}_t\}$
$\mathbf{A}_s$
$\mathbf{A}_t$
$\mathbf{A}_s$
$\mathbf{A}_t$
. . .
ref. 1
ref. 2
ref. 3
ref. K
source speaker
target speaker
source speaker
target speaker
source speaker
target speaker

Step 1.
Estimate parameters $\{\bar{\mathbf{W}}, \mathcal{A}, \mathcal{B}\}$ using reference speakers' speech

Step 2.
Estimate parameters $\{\mathcal{A}', \mathcal{B}'\}$ using source and target speakers' speech with fixing $\bar{\mathbf{W}}$

Step 3.
Estimate hidden units from source speaker's utterance using $\bar{\mathbf{W}}$, $\mathbf{A}_s$ and $\mathbf{B}_s$

Step 4.
From hidden units in Step3, estimate target speaker's features using $\bar{\mathbf{W}}$, $\mathbf{A}_t$ and $\mathbf{B}_t$

Fig. 2. Procedure of voice conversion using an ARBM.

3. APPLICATION TO MANY-TO-MANY VC

In this section, we describe how an ARBM is applied to VC tasks. As shown in Figure 2, each parameter of an ARBM is simultaneously estimated using training data that contains the speech uttered by K reference speakers (Step 1). Then, using a small amount of speech data of the source speaker and the target speaker, we estimate the additional adaptive parameters $\mathcal{A}' = \mathbf{A}_s \cup_3 \mathbf{A}_t$, $\mathcal{B}' = \mathbf{B}_s \cup_3 \mathbf{B}_t$ using Eqs. (9) and (10), where $\mathbf{A}_s$, $\mathbf{A}_t$, $\mathbf{B}_s$, and $\mathbf{B}_t$ are an adaptive matrix for the source speaker, an adaptive matrix for the target speaker, a bias matrix for the source speaker, and a bias matrix for the target speaker, respectively, and $\cup_d$ indicates a concatenate operation along with mode-d, while fixing the other parameters (Step 2). Here, we extend the identity variable $\boldsymbol{s}$ to have the length of $K+2$, and update the parameters $\mathcal{A}$ and $\mathcal{B}$ as $\mathcal{A} \leftarrow \mathcal{A} \cup_3 \mathcal{A}'$ and $\mathcal{B} \leftarrow \mathcal{B} \cup_3 \mathcal{B}'$, respectively. In Step 3, we calculate the latent features (hidden units) from the input features (acoustic features such as MFCCs) of the source speaker $\boldsymbol{v}_s$ as follows:

$$\begin{aligned}\hat{\boldsymbol{h}} &\triangleq \mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v}_s,\boldsymbol{s}_s)}[\boldsymbol{h}]\\ &= \mathcal{S}(\boldsymbol{c} + \mathbf{W}(\boldsymbol{s}_s)^{\mathrm{T}}(\frac{\boldsymbol{v}_s}{\boldsymbol{\sigma}^2}))\\ &= \mathcal{S}(\boldsymbol{c} + (\mathbf{A}_s\bar{\mathbf{W}} + \mathbf{B}_s)^{\mathrm{T}}(\frac{\boldsymbol{v}_s}{\boldsymbol{\sigma}^2})),\end{aligned} \tag{11}$$

where $\boldsymbol{s}_s$ is the vector where the $(S+1)$th element is set to be 1, and the other elements are set to be 0. As Eq. (11) indicates, the latent features are obtained using the weight matrix that is adapted to the source speaker from the speaker-independent weights $\bar{\mathbf{W}}$. Because the column vectors of the adapted weight matrix are similar to the patterns appearing in the source speaker's acoustic features, the obtained latent features $\hat{\boldsymbol{h}}$ represent speaker-independent, possibly phonological, information. Therefore, when we want to convert the speech so that it is as if the target speaker spoke, without changing the phonological information, we just calculate the visible units from the $\hat{\boldsymbol{h}}$ using the identity units $\boldsymbol{s}_t$ (a vector whose $(S+2)$th element is 1 and the others are 0), indicating that the target speaker spoke the speech (Step 4). The converted acoustic features for the target speaker are obtained as follows:

$$\begin{aligned}\hat{\boldsymbol{v}}_t &\triangleq \mathbb{E}_{p(\boldsymbol{v}|\hat{\boldsymbol{h}},\boldsymbol{s}_t)}[\boldsymbol{v}]\\ &= \boldsymbol{b} + (\mathbf{A}_t\bar{\mathbf{W}} + \mathbf{B}_t)\hat{\boldsymbol{h}},\end{aligned} \tag{12}$$

showing that the converted speech is generated from the phonological information $\hat{\boldsymbol{h}}$ and the weight matrix (acoustic-feature patterns) that is adapted to the target speaker. In addition, as Eqs. (11) and (12) indicate, our VC method is based on a non-linear function that maps the acoustic features of the source speaker $\boldsymbol{v}_s$ to those of the target speaker $\boldsymbol{v}_t$.

In practice, we can consider that there are many cases where we have a sufficient number of speakers but only minimal speech data for each speaker. In such cases, although training data for estimating $\bar{\mathbf{W}}$ is quantitatively sufficient, that for estimating $\mathcal{A}$ and $\mathcal{B}$ is so scarce that it may cause errors in the estimation or over-fitting. Therefore, we practically reduce the number of parameters by approximating each matrix $\mathbf{A}_{:,:,k}$ in $\mathcal{A}$ and $\mathbf{B}_{:,:,k}$ in $\mathcal{B}$ with a diagonal matrix and a matrix whose column vectors are the same, respectively, and will report the results in our experiments.

4. EXPERIMENTS

4.1. Conditions

In our VC experiments, we evaluated the performance of our model, an ARBM, using the TIMIT[2] speech corpus that contains speech data uttered by American English speakers of

[2] https://catalog.ldc.upenn.edu/LDC93S1

Table 1. Performance of our method (SDIR [dB]).

# of hidden units	128	192	256	512
female-to-female	7.18	7.26	7.30	7.14
female-to-male	7.64	7.81	7.81	7.82
male-to-female	7.50	7.54	7.61	7.48
male-to-male	7.86	8.00	8.03	8.06
avg.	7.54	7.65	**7.69**	7.63

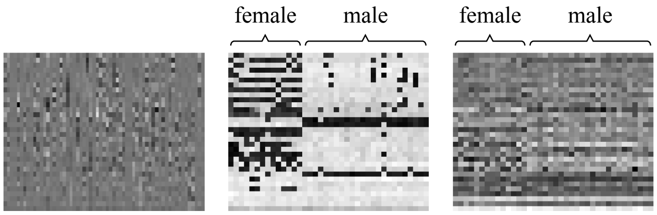


Fig. 3. Left to right: estimated $\bar{\mathbf{W}}$, $\mathcal{A}$, and $\mathcal{B}$.

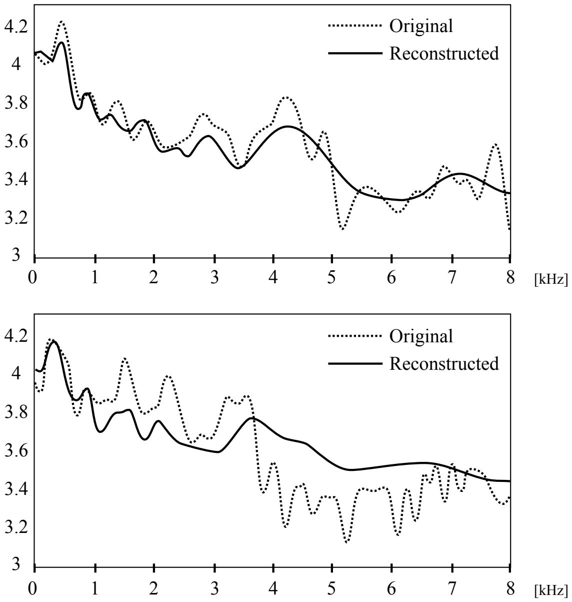


Fig. 4. Log-spectrum from a source speaker and the reconstructed spectrum (above), and log-spectrum from a target speaker and the converted spectrum from the source speech to the target speech (below).

various dialects. From the corpus, we randomly selected 38 speakers (14 females and 24 males) and used the speech of five sentences from each speaker for the parameter estimation (approximately 270,000 frames in total). As an input vector (set to visible units), we used 32-dimensional mel-frequency cepstral coefficients (MFCCs) that were calculated from 513-dimensional STRAIGHT[25] spectra (i.e., $I = 32$). As for the number of hidden units, we compared the performance by changing the number as $J = 128$, 192, 256, and 512. In the training of the ARBM, we used a learning rate of 0.005, a momentum of 0.9, and a batch-size of 50, and set the number of iterations as 500.

For the evaluation of the proposed method, we used pairs of speech data of four female and four male speakers selected from the reference speakers[3] (28 combinations in total), converted the MFCCs for each pair, decoded the MFCCs back to STRAIGHT spectra using filter-theory [26], and compared the performance of cross/same-sex VC in the spectrum space. Here, the speech data used for the evaluation (for calculation of SDIR) were not included in the training data, and had the same contents (two sentences) spoken by each of the speakers. We evaluated each configuration (female-to-female, female-to-male, male-to-female, and male-to-male) by taking an average of SDIR (spectral distortion improvement ratio), which is calculated as follows:

$$\mathrm{SDIR[dB]} = 10 \log_{10} \frac{\sum_d |\mathbf{V}_t(d) - \mathbf{V}_s(d)|^2}{\sum_d |\mathbf{V}_t(d) - \hat{\mathbf{V}}_t(d)|^2}, \qquad (13)$$

where $\mathbf{V}_s$, $\mathbf{V}_t$ and $\hat{\mathbf{V}}_t$ are spectrograms (time-frequency STRAIGHT-spectral matrices) of the source speaker's speech, target speaker's speech, and converted speech, respectively. The higher the value of SDIR is, the better the performance of the VC is. $\mathbf{V}_s$ and $\mathbf{V}_t$ were spectra decoded from MFCC of parallel data of the two speakers, which was created using dynamic programming.

4.2. Results of parameter estimation

Figure 3 shows the actually-estimated parameters of an ARBM, $\bar{\mathbf{W}}$ (partially), $\mathcal{A}$, and $\mathcal{B}$. The vertical axis indicates the dimension of the MFCCs (the first dimension at the top). Since we approximate the adaptive matrix $\mathbf{A}_{:,:,k}$ as a diagonal matrix, we plot only the diagonal elements in columns for $\mathcal{A}$ in Figure 3 (Similarly, we plot the representative vectors for each speaker for $\mathcal{B}$.) For both $\mathcal{A}$ and $\mathcal{B}$ in Figure 3, the 14 column vectors on the left side and the 24 column vectors on the right side correspond to the female speakers and male speakers, respectively.

As shown in Figure 3, we see that each column in $\bar{\mathbf{W}}$ may indicate the phonological pattern of MFCCs. The most interesting point is that each column in $\mathcal{A}$ (and $\mathcal{B}$ as well) that corresponds to the female speakers differs largely from that of male speakers, and the columns corresponding to the same sex differ slightly from each other. This agrees with the intuition that when we try to recognize the identities of speakers, we feel the differences between the sexes larger than the differences between people.

[3] Essentially, it is possible to estimate the speaker-specific parameters for the speakers used in evaluation as in Step 2 in Fig. 2; however, in our experiments, these parameters were estimated simultaneously in Step 1 in Fig. 2 due to the limited number of speakers in the corpus.

4.3. VC performance

First, we list the results of the proposed method in Table 1, showing the SDIRs of each configuration (female-to-female VC, female-to-male VC, male-to-female VC, and male-to-male VC) with their overall averages ("avg.") when changing the number of hidden units of an ARBM. For example, for "female-to-female" we converted the speech of each of four female speakers into that of the other three female speakers, and took the average of any combinations. As shown in Table 1, the more hidden units we give, the more the VC performance was improved, with some exceptions. Comparing the results of 512 hidden units to that of 256 hidden units, on the VC to male (female-to-male and male-to-male) the results were better with 512 hidden units, whereas on the VC to female (female-to-female and male-to-female) the results were better with 256 hidden units. (Consequently, we got better performance with the case of 256 than the case of 512 in the average SDIR). This is because as the number of parameters was increased, the model became overfit. (The ratio of female to male speakers used for the training was 14 to 28, and the fact that the model with 512 hidden units strongly responded to male speaker's speech implies overfitting.)

Next, we show an example of the converted speech from a female speaker (identified with "FCJF0" in the corpus) to a male speaker ("MWAR0") using our method in Figure 4. In this example, we calculated MFCCs from the log-spectrum (dotted line in upper half of Figure 4) at a certain frame of FCJF0's speech, estimated $\hat{\boldsymbol{h}}$ using the weight matrix of the ARBM that adapted to FCJF0, and reconstructed log-spectrum (solid line in lower half of Figure 4) from the converted MFCCs using the weight matrix that adapted to MWAR0. For reference, we also plotted the reconstructed log-spectrum using the weight matrix adapted to FCJF0 after estimating $\hat{\boldsymbol{h}}$ (solid line in upper half of Figure 4), and original log-spectrum of MWAR0 that should be the target (dotted line in lower half of Figure 4). As shown in Figure 4, we can say that the converted spectrum more or less captures the characteristics of the target speaker's spectrum; e.g., the frequencies of spectral peaks (formant information) in low frequencies are similar to those of the target speaker (as for the source speaker's speech, the reconstructed spectrum for FCJF0 is also close to the original spectrum of FCJF0). For both speakers, the reconstructed high-frequency spectra differ greatly from the original spectra. This is due to the reconstruction from MFCCs to spectra. The information for the high frequencies is missing when reconstructed. As shown in Figure 4, our proposed method has a great advantage in that even though we did not train a model of the direct conversion from FCJF0 to MWAR0 and never used parallel data during the training, the FCJF0's speech was converted into that of MWAR0.

5. CONCLUSIONS

In this paper, we proposed a many-to-many voice conversion method that does not require any parallel data during training, by separating speaker-dependent information from the phonological information in speech data. To do this, we also proposed an extension model of an RBM, namely an ARBM, which may be applicable to various other tasks such as controlling emotions in speech, speaker identification, and object recognition. For example, in speaker identification, it would be possible to identify the speaker by estimating the identity units $\boldsymbol{s}$ after training the ARBM in the same manner. Furthermore, the ARBM can separate the phonological information and speaker-related information from speech; therefore, the speaker-independent phonological information may improve speech recognition accuracy. In the future, we will examine a simultaneous estimation method for speaker identity and speech recognition.

6. REFERENCES

[1] Alexander Kain and Michael W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *ICASSP*, 1998, pp. 285–288.

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

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

[4] Li Deng, Alex Acero, Li Jiang, Jasha Droppo, and Xuedong Huang, "High-performance robust speech recognition using stereo training data," in *ICASSP*, 2001, pp. 301–304.

[5] Aki Kunikoshi, Yu Qiao, Nobuaki Minematsu, and Keikichi Hirose, "Speech generation from hand gestures based on space mapping," in *INTERSPEECH*, 2009, pp. 308–311.

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

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

[8] Yannis Stylianou, Olivier Cappé, and Eric Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech and Audio Process.*, vol. 6, no. 2, pp. 131–142, 1998.

[9] Tomoki Toda, Alan W. Black, and Keiichi Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, 2007.

[10] Elina Helander, Tuomas Virtanen, Jani Nurminen, and Moncef Gabbouj, "Voice conversion using partial least squares regression," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 18, no. 5, pp. 912–921, 2010.

[11] Nobuaki Minematsu Daisuke Saito, Hidenobu Doi and Keikichi Hirose, "Application of matrix variate Gaussian mixture model to statistical voice conversion," in *INTERSPEECH*, 2014, pp. 2504–2508.

[12] Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki, "Exemplar-based voice conversion in noisy environment," in *SLT*, 2012, pp. 313–317.

[13] Ryoichi Takashima, Ryo Aihara, Tetsuya Takiguchi, and Yasuo Ariki, "Noise-robust voice conversion based on spectral mapping on sparse space," in *SSW8*, 2013, pp. 71–75.

[14] Srinivas Desai, E. Veera Raghavendra, B. Yegnanarayana, Alan W. Black, and Kishore Prahallad, "Voice conversion using artificial neural networks," in *ICASSP*, 2009, pp. 3893–3896.

[15] L. H. Chen, Z. H. Ling, Yan Song, and L. R. Dai, "Joint spectral distribution modeling using restricted Boltzmann machines for voice conversion," in *INTERSPEECH*, 2013, pp. 3052–3056.

[16] Zhizheng Wu, Eng Siong Chng, and Haizhou Li, "Conditional restricted Boltzmann machine for voice conversion," in *ChinaSIP*, 2013.

[17] Toru Nakashika, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki, "Voice conversion in high-order eigen space using deep belief nets," in *INTERSPEECH*, 2013, pp. 369–372.

[18] Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki, "Voice conversion using RNN pre-trained by recurrent temporal restricted Boltzmann machines," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 23, no. 3, pp. 580–587, 2015.

[19] Athanasios Mouchtaris, Jan Van der Spiegel, and Paul Mueller, "Nonparallel training for voice conversion based on a parameter adaptation approach," *IEEE Trans. Audio, Speech, and Lang. Processs*, vol. 14, no. 3, pp. 952–963, 2006.

[20] Chung-Han Lee and Chung-Hsien Wu, "Map-based adaptation for speech conversion using adaptation data selection and non-parallel training," in *INTERSPEECH*, 2006, pp. 2254–2257.

[21] Tomoki Toda, Yamato Ohtani, and Kiyohiro Shikano, "Eigenvoice conversion based on Gaussian mixture model," in *INTERSPEECH*, 2006, pp. 2446–2449.

[22] Daisuke Saito, Keisuke Yamamoto, Nobuaki Minematsu, and Keikichi Hirose, "One-to-many voice conversion based on tensor representation of speaker space," in *INTERSPEECH*, 2011, pp. 653–656.

[23] KyungHyun Cho, Alexander Ilin, and Tapani Raiko, "Improved learning of Gaussian-Bernoulli restricted Boltzmann machines," in *ICANN*, pp. 10–17. Springer, 2011.

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

[25] Hideki Kawahara, Masanori Morise, Toru Takahashi, Ryuichi Nisimura, Toshio Irino, and Hideki Banno, "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *ICASSP*, 2008, pp. 3933–3936.

[26] Ben Milner and Xu Shao, "Speech reconstruction from mel-frequency cepstral coefficients using a source-filter model," in *INTERSPEECH*, 2002, pp. 2421–2424.