Dictionary Update for NMF-based Voice Conversion Using an Encoder-Decoder Network

Chin-Cheng Hsu[1], *Hsin-Te Hwang*[1], *Yi-Chiao Wu*[1], *Yu Tsao*[2], *and Hsin-Min Wang*[1]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
{jeremycchsu, hwanght, tedwu, whm}@iis.sinica.edu.tw, yu.tsao@citi.sinica.edu.tw

Abstract

In this paper, we propose a dictionary update method for Non-negative Matrix Factorization (NMF) with high dimensional data in a spectral conversion (SC) task. Voice conversion has been widely studied due to its potential applications such as personalized speech synthesis and speech enhancement. Exemplar-based NMF (ENMF) emerges as an effective and probably the simplest choice among all techniques for SC, as long as a source-target parallel speech corpus is given. ENMF-based SC systems usually need a large amount of bases (exemplars) to ensure the quality of the converted speech. However, a small and effective dictionary is desirable but hard to obtain via dictionary update, in particular when high-dimensional features such as STRAIGHT spectra are used. Therefore, we propose a dictionary update framework for NMF by means of an encoder-decoder reformulation. Regarding NMF as an encoder-decoder network makes it possible to exploit the whole parallel corpus more effectively and efficiently when applied to SC. Our experiments demonstrate significant gains of the proposed system with small dictionaries over conventional ENMF-based systems with dictionaries of same or much larger size.

Index Terms: voice conversion, autoencoder, NMF, dictionary update

1. Introduction

The purpose of voice conversion is to transform spectral and prosodic characteristics of an utterance from a source speaker so that the perceived speaker identity matches a target speaker, with other information, such as the linguistic contents, unaltered. In this study, we focus on spectral conversion (SC), whereas inspection on prosody conversion is beyond the scope of this paper.

A wide variety of techniques have been applied to SC, including Gaussian mixture models (GMMs) [1–3], frequency warping [4, 5], deep neural networks (DNNs) [6–9], and exemplar-based approaches [10–13]. Among them, exemplar-based non-negative matrix factorization (ENMF), a confluence of exemplar-based approaches and NMFs [14], is considered exceptionally suitable for the task of SC [12]. An ENMF-based SC system has a meaningful set of basis frames (a.k.a. dictionary) that reconstructs a given input. Conversion is achieved simply via applying the activation weights of the source dictionary to the target dictionary. The paired source and target dictionaries are constructed beforehand from a parallel corpus. Systems of this kind can be applied on the fly (without training procedures). Figure 1 depicts how ENMF-based SC is conducted.

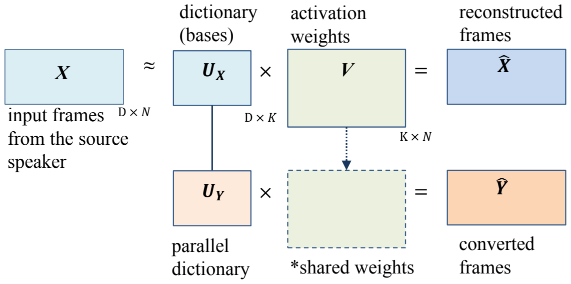

Figure 1: *Basic idea of ENMF-based spectral conversion.*

Aside from the advantages, ENMF-based SC approaches have several practical shortcomings. First, it is unclear how dictionary update could be applied to improve the pair of source and target dictionaries. Second, multiplicative update becomes costly when the dimensionality and number of samples become huge. Third, one still needs a mechanism for dictionary selection in order to reduce the conversion time, although the robustness of ENMF-based SC increases with the dictionary size.

Attempts have been made to tackle the above problems. For example, the authors of [15] relaxed the non-negativity constraint for the dictionaries and adopted features with lower dimension (mel-cepstra coefficients) to make computation manageable. In addition, they dropped the assumption of weight sharing during dictionary update but instead posed a similarity constraint on the individual source and target activation weights so that the paired dictionaries could be learned separately.

In this paper, we propose a reformulation of NMF as an encoder-decoder network (EDN) and conduct dictionary learning using mini-batch Stochastic Gradient Descent (SGD). With this formulation, we are able to obtain compact dictionaries that can well reconstruct the source and target frames, respectively. Our experiments demonstrate that, even with much smaller dictionaries, the proposed EDN-based SC system outperforms the conventional ENMF-based SC system.

The rest of this paper is organized as follows. We first briefly review ENMF-based spectral conversion in Section 2. Then, we describe our proposed method in Section 3, and present our experimental results and discussions in Section 4. Finally, Section 5 concludes this paper.

2. ENMF-based spectral conversion

NMF factorizes a sequence of spectral frames $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$ into a dictionary matrix $\boldsymbol{U}_X = [\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_K]$ and an activation weight matrix

$\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_N]$:

$$\boldsymbol{X} \approx \boldsymbol{U}_X \boldsymbol{V}. \tag{1}$$

Typically, $\boldsymbol{U}_X$ and $\boldsymbol{V}$ can be learned alternately by keeping the other matrix fixed.

For applying NMF to SC, there are one requirement and one assumption. First, a set of parallel frames (source-target frame pairs) whose acoustic contents are aligned is required. The paired source and target dictionaries, $\boldsymbol{U}_X$ and $\boldsymbol{U}_Y$, can therefore be directly constructed from the exemplars. Second, the intrinsic topologies of the source and target speech have to share some similarity. Satisfying these two conditions reduces SC to a self-reconstruction problem, letting ENMF come into play. Similarity of intrinsic topologies implies similarity of the activation weights for reconstruction. In this manner, conversion is realized by finding the shared activation weights $\boldsymbol{V}$ from the source speech $\boldsymbol{X}$ and then applying them directly to the target dictionary $\boldsymbol{U}_Y$ to generate the converted speech $\boldsymbol{Y}$:

$$\boldsymbol{V}^* = \underset{\boldsymbol{V}}{\operatorname{argmin}}\, D(\boldsymbol{U}_X \boldsymbol{V}, \boldsymbol{X}) + C(\boldsymbol{V}), \tag{2}$$

$$\boldsymbol{Y} \approx \boldsymbol{U}_Y \boldsymbol{V}, \tag{3}$$

where $D(\cdot, \cdot)$ is a cost function in terms of, for example, mean square error, and $C(\cdot)$ is a constraint such as L1 norm (sparsity). It should be noted that there are no training phases in ENMF-based SC. Conversion, i.e., solving $\boldsymbol{V}$ based on (2) and then applying the resulting $\boldsymbol{V}^*$ to (3), is conducted online.

3. The proposed encoder-decoder-based dictionary update framework

3.1. NMF as an encoder-decoder network

To better understand the proposed encoder-decoder framework for SC, we begin with reformulating ENMF as an encoder-decoder pair. We will then generalize it to accommodate NMF in SC. Here, ENMF is distinguished from NMF as follows. If the dictionary is directly derived from exemplars without training, it is called ENMF; otherwise, it is NMF. In ENMF, the goal is to find the best activation $\boldsymbol{V}$ given the input $\boldsymbol{X}$ and dictionary $\boldsymbol{U}_X$. We can reformulate ENMF in (2) and (1) as

$$\boldsymbol{V} = f(\boldsymbol{X}, \boldsymbol{U}_X), \tag{4}$$

$$\boldsymbol{X} \approx g_X(\boldsymbol{V}, \boldsymbol{U}_X) = \boldsymbol{U}_X \boldsymbol{V}. \tag{5}$$

Although $f(\cdot)$ has no analytic form due to the non-negativity constraint, we can still approximate it using a feed-forward neural network $\hat{f}_\Theta(\cdot)$ parameterized by Θ as

$$\boldsymbol{V} \approx \hat{\boldsymbol{V}} = \hat{f}_\Theta(\boldsymbol{X}). \tag{6}$$

We refer to $\hat{f}_\Theta(\cdot)$ as the encoding function (a.k.a. encoder). Function $g_X(\cdot)$ in (5) naturally emerges as a decoding function (a.k.a. decoder) because it reconstructs the input using the code $\boldsymbol{V}$. The encoder-decoder pair $(\hat{f}_\Theta, g_X)$ thus constitutes an auto-encoder. With $\hat{f}_\Theta(\cdot)$ designed to yield non-negative outputs (e.g. having rectified outputs), we can approximate an ENMF using an auto-encoder [16].

The decoder $g_X(\cdot)$ is parameterized by $\boldsymbol{U}_X$, which can be trained, and training $\boldsymbol{U}_X$ corresponds to dictionary update in NMF terminology. Dictionary update brings performance gain at the cost of requiring a training process, which is absent in ENMF. As both the activation and the dictionary are learnable, the auto-encoder now approximates an NMF. Taking NMF as an auto-encoder gives us a clearer insight on how to update the parallel dictionaries for SC, which will be described in 3.3.2.

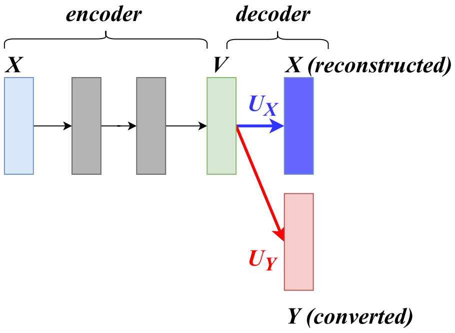

Figure 2: *Illustration of EDN-based spectral conversion. There is only one encoder because only source speech is available during the conversion phase.*

3.2. Encoder-decoder network for spectral conversion

Now we extend the encoder-decoder framework to tackle the problem of SC. Conversion requires an additional decoder that transforms the code $\boldsymbol{V}$ into a target spectrum without changing linguistic contents. This is exactly what the target dictionary $\boldsymbol{U}_Y$ is for. We can again regard it as a decoder, denoted by $g_Y(\cdot)$, and concatenate it to the encoder $\hat{f}_\Theta(\cdot)$. There are one encoder and two juxtaposed decoders now. For brevity, we refer to the proposed method as an Encoder-Decoder Network (EDN). Figure 2 depicts its architecture. We summarize it using the following equations:

$$\text{Encoding} : \hat{f}_\Theta(\boldsymbol{X}) = \hat{\boldsymbol{V}}, \tag{7}$$

$$\text{Decoding to the source} : g_X(\hat{\boldsymbol{V}}) = \boldsymbol{U}_X \hat{\boldsymbol{V}} = \hat{\boldsymbol{X}}, \tag{8}$$

$$\text{Decoding to the target} : g_Y(\hat{\boldsymbol{V}}) = \boldsymbol{U}_Y \hat{\boldsymbol{V}} = \hat{\boldsymbol{Y}}. \tag{9}$$

Note that the decoders merely perform a linear transformation with either dictionary $\boldsymbol{U}_X$ or $\boldsymbol{U}_Y$. The conversion phase is similar to that of the ENMF-base SC. Spectral frames $\boldsymbol{X}$ from a source speaker are fed into the encoder network to get the code $\hat{\boldsymbol{V}}$, which is then used by the target decoder to obtain the target frames $\hat{\boldsymbol{Y}}$ (cf. (7) and (9)).

3.3. Training the EDN

3.3.1. Training the encoder

The training process of EDN is divided into two stages. The first stage involves only the auto-encoder, which approximates an ENMF. The encoder parameters Θ are updated to minimize the approximation divergence between input $\boldsymbol{X}$ and the auto-encoder output $\hat{\boldsymbol{X}}$, which is defined as,

$$J_{encoder} = D(\boldsymbol{X}, \hat{\boldsymbol{X}}) = \sum_{n=1}^{N} D_{KL}(\boldsymbol{x}_n || \hat{\boldsymbol{x}}_n), \tag{10}$$

where $\boldsymbol{x}_n$ is a reference frame (ground truth), $\hat{\boldsymbol{x}}_n$ is its reconstructed version, n is the frame index, and N is the number of frames. Kullback-Leibler divergence (KLD), which is defined as

$$D_{KL}(\boldsymbol{x}_n || \hat{\boldsymbol{x}}_n) := \sum_{m=1}^{M} x_{m,n} \frac{\log x_{m,n}}{\log \hat{x}_{m,n}}, \tag{11}$$

where m is the dimension index, and M is the dimensionality, is chosen as the cost function $D(\cdot, \cdot)$ for the following reasons. When dealing with power or magnitude spectra, the order of magnitude varies across dimensions drastically. Logarithms in KLD help alleviate the issue. In our experiments, both input and output spectra are normalized to unit-sum so that KLD can be applied readily.

Substituting $\hat{\boldsymbol{X}}$ in (10) with (8) and then $\hat{\boldsymbol{V}}$ with (7), the cost function becomes

$$\begin{aligned} J_{encoder} &= D(\boldsymbol{X}, \boldsymbol{U}_X \hat{\boldsymbol{V}}) \\ &= D(\boldsymbol{X}, \boldsymbol{U}_X \hat{f}_\Theta(\boldsymbol{X})) \\ &= \sum_{n=1}^{N} D_{KL}(\boldsymbol{x}_n || \boldsymbol{U}_X \hat{f}_\Theta(\boldsymbol{x}_n)). \end{aligned} \tag{12}$$

Keeping $\boldsymbol{U}_X$ fixed, we update the encoder parameters by

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}}\, J_{encoder}. \tag{13}$$

Encoder training is very similar to the root finding procedures of ENMF-based SC (cf. (2)), except that we do not use the multiplicative update rules. Mini-batches of source frames are fed into the encoder, gradients are computed, and Θ is updated using the SGD rules.

3.3.2. *Training the decoders*

The second stage of the training process of EDN is for dictionary update, or equivalently, decoder training. The decoder parameters $(\boldsymbol{U}_X, \boldsymbol{U}_Y)$ are initially the parallel dictionaries, which consist of randomly selected exemplars, used in ENMF-based SC. Their ability to reconstruct or convert speech is limited, and that is why we need to update them. Similar to (10) and (12), we first define a joint cost function for the two decoders:

$$\begin{aligned} J_{decoder} &= \sum_{n=1}^{N} \alpha D(\boldsymbol{X}, \hat{\boldsymbol{X}}) + (1-\alpha) D(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) \\ &= \sum_{n=1}^{N} \alpha D_{KL}(\boldsymbol{x}_n || \boldsymbol{U}_X \hat{f}_\Theta(\boldsymbol{x}_n)) \\ &+ \sum_{n=1}^{N} (1-\alpha) D_{KL}(\boldsymbol{y}_n || \boldsymbol{U}_Y \hat{f}_\Theta(\boldsymbol{x}_n)), \end{aligned} \tag{14}$$

where α is the importance weight of self-reconstruction. We also choose KLD in (11) as our cost functions. Dictionaries are updated by

$$\{\boldsymbol{U}_X^*, \boldsymbol{U}_Y^*, \Theta^*\} = \underset{\boldsymbol{U}_X, \boldsymbol{U}_Y, \Theta}{\operatorname{argmin}}\, J_{decoder}. \tag{15}$$

Special care should be taken of the dictionaries. We actually apply a rectifier and a unit-sum normalizer to them so that the resulting dictionaries conform to our specified forms. Note that we also update the encoder in the second stage.

We conduct training as follows. The EDN takes as input a frame $\boldsymbol{x}_n$ from the source speaker, and tries to minimize the divergence between the decoder outputs $(\hat{\boldsymbol{x}}_n, \hat{\boldsymbol{y}}_n)$ and the regression targets $(\boldsymbol{x}_n, \boldsymbol{y}_n)$, which are the input itself and its corresponding frame from the target. Similar procedures are applied to update the encoder parameters Θ. Note that, to avoid notation cluttering, we describe the procedures using *frame* pairs as opposed to *mini-batch* pairs that are used in practice.

This architecture allows us to update dictionaries $\boldsymbol{U}_X$ and $\boldsymbol{U}_Y$ separately in form, jointly in reality since each paired source and target frames share the same activation weights. This multi-task design avoids updating a joint parallel dictionary with twice the original dimension.

4. Experiments

4.1. Experimental settings

4.1.1. *The VCC2016 speech corpus*

The proposed SC system was evaluated on a parallel English corpus from the Voice Conversion Challenge 2016 [17]. There are 5 male and 5 female speakers in this corpus. Each speaker has 150 utterances as the training set and 12 utterances as the evaluation set. Five out of the ten speakers are designated to be the conversion targets (2 female and 3 male speakers) and the other five sources (3 female and 2 male speakers). The testing set comprises 54 utterances per target speaker.

We conducted experiments on a subset of the speakers. Two speakers were chosen as sources (SF1 and SM1) and another two as targets (TF2 and TM3). We reported two types of spectral conversion: intra-gender and cross-gender.

4.1.2. *Feature sets*

We used the STRAIGHT toolkit [18] to parametrize speech into the smoothed STRAIGHT spectra (SP for short), aperiodicity (AP), and pitch contours (F0). The FFT length was set to 1024, so the resulting AP and SP were both 513-dimensional. The frame shift was 5 milliseconds (ms) and the frame length was 25 ms. Neither contextual nor dynamic features were utilized in any forms. The SP was converted using our proposed method or the baseline systems. All systems converted F0 using the same linear mean-variance transformation. Energy and AP were kept unmodified. In all the systems, every input frame of SP was normalized to unit-sum. After spectral conversion, energy was compensated back to SP, and STRAIGHT took in all the parameters to synthesize utterances.

Each parallel training set was aligned using dynamic time warping (DTW) with 24-ordered Mel-cepstral coefficients (MCC) extracted from SP. Energy-based voice activity detection (VAD) was used to exclude the silence segments. After alignment, the length of a source utterance remained the same while some frames from the target were duplicated or decimated.

4.2. The baseline systems

The first baseline SC system was a conventional ENMF-based one (as described in [12]) with dictionaries of 512 bases. For each source-target pair, 512 randomly selected frames from the whole source training samples were specified as the source dictionary $\boldsymbol{U}_X$. Their corresponding frames from the target speaker were specified to be the target dictionary $\boldsymbol{U}_Y$. The system is denoted as ENMF-512 in the following experiments. An extended baseline (ENMF-3000) was built on larger dictionaries of 3000 bases (the previously selected 512 bases plus additional 2488 bases). The unit-sum constraint naturally poses a sparse constraint on the baseline systems.

4.3. The encoder-decoder network-based system

4.3.1. *Configurations and hyper-parameters*

The encoder was a feed-forward neural network with 2 hidden layers, each with 1024 nodes. Rectifier linear unit (ReLU) [19] was applied to each layer to provide non-linearity and to ensure the non-negativity constraint of the activation. The batch size was 512. Both the activation and the dictionaries were renormalized to unit-sum. Note that the dictionaries are outputs

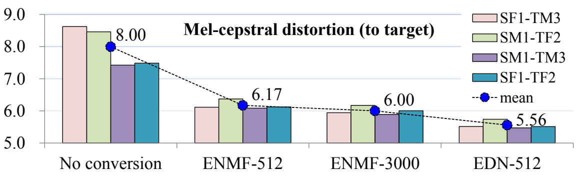

Figure 3: *Graphical comparison of MCDs.*

from ReLUs followed by a unit-sum normalization, so the non-negativity can be guaranteed.

The maximum number of epochs for encoder training was set to 100, and the learning rate was 0.001. As for dictionary update, the learning rate was set to 0.01 and decreased by a factor of 0.1, for three times per 50 epochs. The optimizers were Adam [20]. Usually it only took a few dozens of epochs to reach a reasonable solution. The importance coefficient α in (14) was empirically set to 0.15, striking a balance between the reconstruction and the conversion divergence. It might be tempting to set α to 0 so that EDN could focus on conversion only. However, our preliminary results revealed no obvious advantages for such setting, presumably because of imperfect frame alignments.

4.3.2. *Training procedures*

For each source-target pair, the 512-base source and target dictionaries used in the baseline ENMF-512 system served as initial $\boldsymbol{U}_X$ and $\boldsymbol{U}_Y$, respectively. As mentioned in Sec. 3, the encoder parameters Θ were trained in the first stage of EDN training, and then the dictionaries $\boldsymbol{U}_X$ and $\boldsymbol{U}_Y$ and the encoder parameters Θ were updated in the second stage of EDN training. We denote our proposed EDN-based SC system with 512 bases as EDN-512 in the following experiments.

4.4. Objective evaluation

4.4.1. *Mel-cepstral distortion*

We visualize mean mel-cepstral distortion (MCD) values on the evaluation set in Figure 3. Our proposed method (EDN-512) achieved the lowest distortion under all test conditions.

Performance gain of EDN-512 over ENMF-512 is attributed to training, which grants EDN access to the whole training set. Compared to ENMF-3000, EDN-512 is superior in that it achieves better quality with fewer bases. This fact indicates the effectiveness of the training process, which brings EDN a stronger power of representation so that the code can be decoded into the target speech more precisely.

4.4.2. *Code sparsity*

We can observe three things from the resulting activation (code) $\boldsymbol{V}$ shown in Figure 4. First, the activation is still sparse after dictionary update, implicitely proving that the EDN simulates an NMF. Most of the activation weights are exactly zero thanks to ReLU non-linearity. For typical voiced frames (e.g. from the 20th to the 100th) , usually less than 100 bases are activated, and only a few of them dominate. Second, consecutive frames have similar activations, meaning that temporal smoothness is guaranteed. Third, the fact that the codes are shared indicates that they carry certain speaker-independent information (assumably, acoustics) because they can be decoded into voices of different speakers.

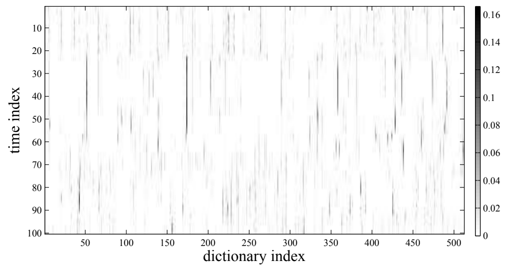

Figure 4: *EDN activation (code) matrix of a typical voiced speech segment. Most entries are exactly zero, meaning that the code is highly sparse.*

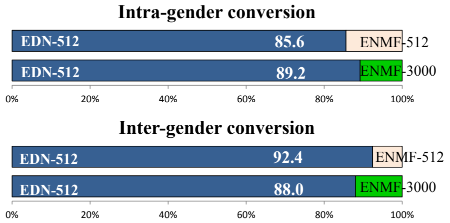

Figure 5: *Preference score from the ABX test for conversion quality. The source is a female speaker (SF1), the intra-gender target is TF2, and the inter-gender target is TM3.*

4.5. Subjective evaluation

We also evaluated voice quality using the ABX test. Ten listeners were invited to evaluate all the testing sets, each with 25 sentences. The results are shown in Figure 5. Similarity was not reported because all the three methods achieved a nearly identical level of similarity.

Obviously, subjective evaluation demonstrated significantly higher voice quality of the proposed EDN-based system over that of the two ENMF-based baseline systems. The results are consistent with the objective evaluation.

5. Conclusions

This paper has presented a dictionary update framework for NMF-based spectral conversion by reformulating NMF as an encoder-decoder network. The merits are two-fold. First, the proposed method avoids explicit joint dictionary update that doubles the dimensionality. Second, the learned dictionary is much more compact and has a higher representational power, resulting in better voice quality in the converted speech. The encoder-decoder network formulation can be easily generalized to application cases without the non-negativity constraint. We will consider these cases in the future.

6. References

[1] 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*, 2007.

[2] S. Takamichi, T. Toda, A. W. Black, and S. Nakamura, "Modulation spectrum-constrained trajectory training algorithm for GMM-based voice conversion," *Proc. ICASSP*, 2015.

[3] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Tranns on Speech and Audio Processing*, vol. 6, no. 2, 1998.

[4] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 18, no. 5, pp. 922–931, July. 2010.

[5] E. Godoy, O. Rosec, and T. Chonavel, "Voice conversion using dynamic frequency warping with amplitude scaling, for parallel or nonparallel corpora," *IEEE Trans. Audio, Speech, Lang., Process*, vol. 20, no. 4, pp. 1313–1323, May. 2012.

[6] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Trans. Audio, Speech, Lang., Process.*, vol. 18, no. 5, pp. 954–964, 2010.

[7] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets," *Proc. INTERSPEEH*, 2013.

[8] H.-T. Hwang, Y. Tsao, H.-M. Wang, Y.-R. Wang, and S.-H. Chen, "A probabilistic interpretation for artificial neural network-based voice conversion," *Proc. APSIPA*, 2015.

[9] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, pp. 1506–1521, 2014.

[10] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion in noisy environment," *Proc. Spoken Language Technology Workshop (SLT)*, pp. 313 – 317, 2012.

[11] Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li, "Exemplar based voice conversion using non-negative spectrogram deconvolution," *Proc. 8th ISCA Speech Synth. Workshop (SSW8)*, 2013.

[12] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1506–1521, 2014.

[13] Y.-C. Wu, H.-T. Hwang, C.-C. Hsu, Y. Tsao, and H.-M. Wang, "Locally linear embedding for exemplar-based spectral conversion," *Proc. INTERSPEECH*, 2016.

[14] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Adv. Neural Inf. Process. Syst.*, vol. 13, p. 556–562, 2001.

[15] R. Aihara, T. Takiguchi, and Y. Ariki, "Semi-non-negative matrix factorization using alternating direction method of multipliers for voice conversion," *Proc. ICASSP*, 2016.

[16] P. Smaragdis, "NMF? neural nets? it's all the same..." *https://drive.google.com/file/d/0B-AMJkGqwFGfRmJhZDl2QjJXcUU/view*, 2015.

[17] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The voice conversion challenge 2016," *Proc. INTERSPEECH*, in press.

[18] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive timefrequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Commun.*, no. 3-4, pp. 187–207, 1999.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proc. ICML*, p. 807–814, 2010.

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