HMM-Based Maximum Likelihood Frame Alignment for Voice Conversion from a Nonparallel Corpus

Ki-Seung LEE†a), Member

SUMMARY One of the problems associated with voice conversion from a nonparallel corpus is how to find the best match or alignment between the source and the target vector sequences without linguistic information. In a previous study, alignment was achieved by minimizing the distance between the source vector and the transformed vector. This method, however, yielded a sequence of feature vectors that were not well matched with the underlying speaker model. In this letter, the vectors were selected from the candidates by maximizing the overall likelihood of the selected vectors with respect to the target model in the HMM context. Both objective and subjective evaluations were carried out using the CMU ARCTIC database to verify the effectiveness of the proposed method.

key words: nonparallel speech corpus, voice conversion, hidden Markov model, maximum likelihood criterion

1. Introduction

Voice conversion (VC) is a process whereby the features derived from speech signals are changed, so that one voice resembles another. If the features of one speaker (source speaker) are modified so they approximate those of another specific speaker (target speaker), the resultant speech signals sound as if they are from the target speaker. This technique is referred to as voice personality transformation. Voice personality transformation has numerous applications in a variety of areas such as personification of text-to-speech synthesis systems, preprocessing for speech recognition [1], enhancing the intelligibility of abnormal speech [2], and foreign language training systems [3].

A large parallel training corpus is necessary for robust, high-quality VC. However, such parallel data may not always be available. In practice, even if two speakers utter the same words, their different speaking rates make it unlikely that a synchronized set of spectral feature sequences would result. To time-align these sequences, dynamic time warping (DTW) is applied in a preprocessing step. This time-alignment algorithm, however, can produce errors that affect the final performance of the conversion task.

To tackle the problems associated with parallel training data, nonparallel training methods for VC have been proposed [4]–[6]. A transformation approach similar to maximum likelihood linear regression (MLLR) was proposed by Ye et al. [4]. A parameter adaptation approach was proposed by Mouchtaris et al. wherein the conversion parameters for a given pair of source and target speakers were adapted to a particular pair of speakers for which no parallel corpus was available [5]. Kumar et al. proposed a phoneme-by-phoneme basis conversion scheme where the phonemes of a source speaker are transformed into the corresponding phonemes of a target speaker [6].

In an effort to overcome the limited performance of VC from a nonparallel corpus, an iterative combination of a nearest neighbor search step and a conversion step alignment method (INCA) was proposed [7], [8]. In this approach, a voice conversion function was iteratively trained from the nearest neighbor alignment between the intermediate converted voice and the target voice. The underlying assumption was that the spectral features that were near one another belonged to the same phone. Dynamic characteristics were partially utilized in the temporal-context INCA (TC-INCA) method [9] where alignment was carried out on a set of context vectors obtained by concatenating a number of successive vectors. Such forced-alignment methods yielded a sequence of spectral vectors that were not well matched with the underlying speaker model, since the speaker-specific information was not sufficiently considered in the alignment procedure. In the present study, the alignment problem was formulated as finding the optimum sequence of vector pairs in the sense of maximizing not only similarities between them, but also the overall likelihood of the sequence of the aligned vectors with respect to the speaker model (e.g., HMM).

The remainder of this letter is structured as follows: In Sect. 2 the problems of the INCA method are described. In Sect. 3, the proposed vector alignment method is explained. In Sect. 4, the experimental results are presented and discussed. Finally, concluding remarks are summarized in Sect. 5.

2. Baseline INCA Algorithm

We assumed that the training data would contain two sets of spectral feature vectors $X = \{x_k\}_{k=1}^{N_x}$ and $Y = \{y_j\}_{j=1}^{N_y}$ that, respectively, are derived from the speech of the source speaker and the target speaker. $N_x$ and $N_y$ are the number of spectral feature vectors for the source and target speakers, respectively. The INCA algorithm iteratively finds the set of paired vectors $\{x_k,y_j\}$ (alignment step) and finds the auxiliary conversion function $\hat{F}$ using the set of paired vectors (training step). In the $i$-th alignment step, two matching functions
that could satisfy the following conditions: 

\[
    p_i(k) = \arg \min_j \| \mathcal{F}_{i-1}(\mathbf{x}_k) - \mathbf{y}_j \| \\
    q_i(j) = \arg \min_k \| \mathcal{F}_{i-1}(\mathbf{x}_k) - \mathbf{y}_j \|
\]

which is referred to as the nearest neighbor (NN) alignment. In the subsequent training step, parallel training sets \{\mathbf{x}_k, \mathbf{y}_{\mathcal{F}_i(k)}\}_{k=1}^{N_i}, \{\mathbf{y}_{\mathcal{F}_i(k)}, \mathbf{y}_j\}_{j=1}^{N_y} are used to obtain the auxiliary conversion function, \(\mathcal{F}_i\). A new conversion function is then used for the next iteration and the process is repeated until the convergence threshold is reached. At each iteration, the objective function is calculated as follows:

\[
    D_{\text{NN}} = \frac{1}{N_x + N_y} \left( \sum_{k=1}^{N_i} \| \mathbf{x}_k' - \mathbf{y}_{\mathcal{F}_i(k)} \|^2 + \sum_{j=1}^{N_y} \| \mathbf{x}_k' - \mathbf{y}_j \|^2 \right)
\]

where \(\mathbf{x}_k' = \mathcal{F}(\mathbf{x}_k)\) and \(\mathbf{x}_k' = \mathcal{F}(\mathbf{y}_{\mathcal{F}_i(k)})\).

One of the drawbacks associated with the INCA algorithm is that the vector sequence resulted from NN alignment lacks similarity with the target speaker model. This is because only the distance between two vectors is minimized or, equivalently, only the inter-speaker similarities are considered in the alignment step. Statistical similarities between the selected vectors and the target model (intraspeaker similarity) are not considered. Our assumption was that the conversion function obtained by using a vector sequence closer to not only the converted vectors but also the target model (HMM) would further improve the conversion performance, particularly for vector stream derived from natural speech.

3. Maximum Likelihood of Frame Alignment Using HMM

In the present study, the previous NN alignment method was modified so that the resultant vector stream would be a good match with the underlying speaker model to some degree. A block diagram of the proposed frame alignment method is presented in Fig. 1, which is composed of a candidate-selection step and a vector-selection step. The speaker model \(\Lambda\) is represented in the HMM context, and, hence, the model includes the following HMM parameters:

\[
    \Lambda = \{\mathbf{A}, \mathbf{B}, \pi\} = \{a_{ij}, b_i, \pi_i\}, 1 \leq i, j \leq N_y
\]

where \(a_{ij}\) is the transient probability density function (PDF) from states \(i\) and \(j\), \(b_i\) is the state observation PDF for state-\(i\), and \(\pi_i\) is the initial PDF of state-\(i\). \(N_y\) is the number of states. Let \(\Lambda_x\) and \(\Lambda_y\) be the source and target models, respectively, and our objective was to find the source/target vector sequences \(\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_k\}_{k=1}^{N_i}\) and \(\tilde{\mathbf{Y}} = \{\tilde{\mathbf{y}}_j\}_{j=1}^{N_y}\) that could satisfy the following conditions:

\[
    \mathbf{X}^* = \arg \max_{\mathbf{X}} P_{\tilde{\mathbf{X}}|\Lambda_x}(\tilde{\mathbf{X}}|\Lambda_y)
\]

These vector sequences can be constructed via a Viterbi trellis search for all combinations of training vectors. In this case, however, there were two problems: 1) Since only the maximum likelihood criterion is employed in (4), the resultant vector stream lacks inter-speaker similarity. 2) In the case of using a large training corpus, a huge number of combinations are evaluated. Hence, practical implementation is not a trivial undertaking. To tackle this problem, a set of the candidates was constructed \(a\) prior as follows:

\[
    S_y(\mathbf{x}_k) = \{\mathbf{y} | \| \mathcal{F}(\mathbf{x}_k) - \mathbf{y} \| \leq \epsilon\}
\]

\[
    S_x(\mathbf{y}_j) = \{\mathbf{x} | \| \mathcal{F}(\mathbf{x}) - \mathbf{y}_j \| \leq \epsilon\}
\]

where \(S_y(\mathbf{x}_k)\) is a set of the target candidates for the source vector \(\mathbf{x}_k\) and \(S_x(\mathbf{y}_j)\) is a set of the source candidates for the target vector \(\mathbf{y}_j\). \(\epsilon\) is the threshold, which can be adjusted so that the number of candidates, \(N_y\) is 100.

The optimal sequence was constructed from the vectors selected from the set of candidates. This was performed on an utterance-by-utterance basis in this study. For the target model \(\Lambda_y\), and the arbitrary selected HMM state sequence \(\Omega = \{\omega_i\}_{i=1}^{T}\), the log-likelihood of the vector sequence \(\tilde{\mathbf{Y}} = \{\tilde{\mathbf{y}}_i\}_{i=1}^{T}\) is given by

\[
    \log P_{\tilde{\mathbf{y}}|\Lambda_y}(\tilde{\mathbf{Y}}|\Lambda_y) = \sum_{t=1}^{T} \left[ b'_{\omega_t}(\tilde{\mathbf{y}}_t) + C_{t-1,t} \right]
\]

where

\[
    b'_{\omega_t}(\mathbf{y}_t) = \log b_{\omega_t}(\mathbf{y}_t),
\]

\[
    C_{t-1,t} = \begin{cases} 
        \log \pi_{\omega_t} & \text{if } t = 1, \\
        \log a_{\omega_{t-1},\omega_t} & \text{otherwise}
    \end{cases}
\]
and $T$ is the length of the utterance. The optimal vector sequence $\mathbf{Y}^*$ is then given by

$$
\mathbf{Y}^* = \arg \max_{\mathbf{Y} \in S(Y_X)} \left( \max_{t \in [1,T]} \log[P_{\mathbf{Y}|\mathbf{Y}^{-1}}(\mathbf{Y} | \Lambda_{\gamma}, \Omega)] \right)
$$

where $S_y(X) = \{S_y(x)\}_{t=1}^T$ is the set of candidates for $1 \leq t \leq T$ and $\Omega_T$ denotes the set of all possible HMM state sequences. Equation (7) can be maximized using a dynamic programming technique, such as a Viterbi-trellis search. In the trellis, each node corresponds to a combination of HMM programming technique, such as a Viterbi-trellis search. In the trellis, each node corresponds to a combination of HMM


Let $\mathcal{S}_t(n)$ be the accumulated likelihood for the $n$-th node at time $t$, and the forward recursion is as follows:

$$
\mathcal{S}_t(n) = \max_m \{ \mathcal{S}_{t-1}(m) + \log a_{\omega(m),\omega(n)} + b'(n,m) \}
$$

where $1 \leq n \leq M$, $1 \leq m \leq M$, and $\omega(m)$ is the HMM-state index of the $m$-th node, and $\mathcal{S}_t(n)$ is the backtracking pointer for the $n$-th node at time $t$. Let $\mathbf{y}_t^{(n)}$ denote the $n$-th candidate at time $t$, and $b'(n,m)$ is given by

$$
b'(n,m) = b'_{\omega(m)}(1 \mathbf{y}_t^{(n)} \Delta \mathbf{y}_t^{(n,m)} \mathbf{y}_t^{(m)})
$$

where $1 \leq n,m \leq N_c$, and $\Delta \mathbf{y}_t^{(n,m)} = \mathbf{y}_t^{(n)} - \mathbf{y}_t^{(m)}$, i.e., the difference between the $m$-th candidate vector at time $t-1$ and the $n$-th candidate vector at time $t$. This means that the sequence of vectors is selected to maximize the overall likelihood of both static and dynamic features with respect to the target model. As a result, both static and dynamic characteristics of the selected vector sequence are statistically similar to those of the target speaker.

After the final accumulated likelihoods, $\mathcal{S}_t(n)$, for all $n$ have been computed, the best node sequence, $\mathbf{z}^* = \mathbf{z}_1^*, \mathbf{z}_2^*, \ldots, \mathbf{z}_T^*$, is obtained using the following backward recursion:

$$
\mathbf{z}_t^* = \arg \max_{1 \leq n \leq M} \mathcal{S}_t(n)
$$

$$
\mathbf{z}_{t+1}^* = \mathbf{y}_{t+1}(\mathbf{z}_{t+1}^*), t = T - 1, T - 2, \ldots, 1.
$$

The optimal transformed sequence $\mathbf{Y}^* = \{\mathbf{y}_t^{(n)}\}_{t=1}^T$ is constructed by selecting the candidate vector corresponding to the best node.

The optimal source vector sequence, $\mathbf{X}^*$, can be constructed in a similar way, by replacing $\Lambda_{\gamma}$ with $\Lambda_{\gamma'}$. After finishing construction of the optimal vector sequences $\mathbf{X}^*$ and $\mathbf{Y}^*$ for all utterances in the training data, the conversion function was obtained using the set of the vector pairs $\{\mathbf{x}_t^*, \mathbf{y}_t^{(n)}\}_{t=1}^T$ and $\{\mathbf{x}_t^*, \mathbf{y}_t^{(n)}\}_{t=1}^T$ where $T(n)$ and $N_n$ are the length of the $n$-th utterance and the total number of the utterances, respectively. The GMM-based conversion function [10] that was also employed in the original INCA algorithm was adopted.

4. Experimental Results

To evaluate the effectiveness of the proposed method, the CMU ARCTIC database [11] for US English was used. Two male (M) speakers, bdl and rms, and two female (F) speakers, clb and slt, were used. Voice conversion was carried out on four different gender pairs including bdl→rms (M→M), rms→clb (M→F), clb→rms (F→M), and slt→clb (F→F) conversion. The conversion rules were obtained using 200 utterances, and the remaining 100 utterances were used for evaluation. The LPC cepstrum (LPCC) was used as a spectral feature parameter. The orders of the LPC coefficients and the LPCC were 20 and 30, respectively. A 25-ms length Hanning window was used to compute the LPC parameters in 10 ms intervals.

Since phonetic matching was not considered in the NN-alignment of the INCA algorithm, the distances between the two vectors with phonemes that did not match were involved with the objective measurement given by (2). Even in non-parallel VC, it is desirable that evaluation is made on the pairs of the two vectors with the same phoneme. In this study, since parallel corpora were available in the CMU database, evaluation was carried out on the pairs of the two LPCCs that were aligned using DTW. Note that the time-aligned LPCC sequences were used only for evaluation, and not for finding the conversion function.

Two objective measures were used to evaluate the performance of the underlying voice conversion methods. First, a distortion ratio, $D_{rat}$, was used, as follows:

$$
D_{rat} = \frac{D(\mathbf{Y}, \mathbf{Y}^*)}{D(\mathbf{X}, \mathbf{Y})} \times 100(\%)
$$

where $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Y}^*$ are the LPCC sequences for the source speaker, the target speaker, and the transformation, respectively, and $D(\mathbf{X}, \mathbf{Y})$ denotes the average Euclidean distance between vectors $\mathbf{X}$ and $\mathbf{Y}$.

Another objective measure is the log likelihood ratio as follows:

$$
L_{rat} = \log \left( \frac{p(\hat{\mathbf{Y}}|\Lambda_{\gamma})}{p(\hat{\mathbf{Y}}|\Lambda_{\gamma'})} \right) = \log p(\hat{\mathbf{Y}}|\Lambda_{\gamma}) - \log p(\hat{\mathbf{Y}}|\Lambda_{\gamma'})
$$

In the present study, an ergodic HMM was employed for both models $\Lambda_{\gamma}$ and $\Lambda_{\gamma'}$ that were trained using 200 utterances for each speaker. The number of states and the number of Gaussians were heuristically determined, which were 10 and 5, respectively. Only one HMM was trained from each speaker, which was used for construction of the optimal vector sequences and evaluation.

Table 1 shows the distortion ratios for two vector matching methods. For all cases, a lower distortion ratio was observed for the proposed ML-based selection. As shown in Table 2, the ML-based selection also yielded a higher log likelihood ratio compared with the NN-alignment. These results indicated that the vector sequences constructed by ML selection are likely to increase the VC performance.
A listening test was conducted to subjectively evaluate the feasibility of the proposed method. In this test, 10 test utterances were used and each utterance was presented three times to 18 listeners. To compensate for the differences in prosody between two subjects, the average pitch period of the source speech was converted to that of the target speech. Figure 2 shows the ABX test results. Except for F→F conversion, the ML-based selection method was 13.6 ~ 3.6% higher than the NN-alignment method. The subjects who participated in the listening test indicated that speech conversion involved with ML-selection were usually closer to the target speech. This was also confirmed by the two-way ANOVA test where the p-value of the conversion method factor was less than 0.05.

5. Conclusions

In this letter, a new vector alignment method was proposed to improve the performance of VC from non-parallel corpus. The experimental results from both objective and subjective evaluation showed that the proposed method improved the performance of previous non-parallel VC methods. Future work will focus on constructing not only the spectral feature sequence but also a prosody-related feature sequence.

References


