Local partial least square regression for spectral mapping in voice conversion

Xiaohai Tian*, Zhizheng Wu†, Eng Siong Chng*†
*Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore
†School of Computer Engineering, Nanyang Technological University, Singapore
xhtian@ntu.edu.sg, wuzz@ntu.edu.sg, aseschng@ntu.edu.sg

***Abstract*—Joint density Gaussian mixture model (JD-GMM) based method has been widely used in voice conversion task due to its flexible implementation. However, the statistical averaging effect during estimating the model parameters will result in over-smoothing the target spectral trajectories. Motivated by the local linear transformation method, which uses neighboring data rather than all the training data to estimate the transformation function for each feature vector, we proposed a local partial least square method to avoid the over-smoothing problem of JD-GMM and the over-fitting problem of local linear transformation when training data are limited. We conducted experiments using the VOICES database and measure both spectral distortion and correlation coefficient of the spectral parameter trajectory. The experimental results show that our proposed method obtain better performance as compared to baseline methods.**

I. Introduction

Voice conversion is a process to modify a speech signal uttered by one speaker (source) to sound like a desired target speaker without changing the linguistic information. The conversion process, which transforms the source feature vectors into the target feature space, includes two phases: off-line training phase and real-time conversion phase. During the off-line training phase, a transformation function is estimated from parallel source and target feature vector sequence. While in the real-time conversion phase, the transformation function is applied to an input testing utterance to generated converted speech signal. The features can be any parameters which represent the speaker identity, such as spectral envelop [1], [2], prosody [3], [4], [5], duration [3], [6]. As spectral envelops contain more speaker characteristics information, spectral mapping is the one of the most important techniques in voice conversion.

To implement a robust spectral conversion function, a number of techniques have been proposed. A linear conversion function has been implemented by joint density Gaussian mixture model (JD-GMM) with both minimum mean square error and maximum likelihood criteria [1], [2], partial least square regression [7], mixture of factor analyzers [8], local linear transformation [9] and so on. In addition to the linear conversion function, by assuming that the source and target speech features have non-linear relationship, methods such as artificial neural network [10], kernel partial least square [11], and conditional restricted Boltzmann machine [12], have also been proposed. Due to the probabilistic treatment and flexible implementation, JD-GMM based method [1], [2] has become the mainstream method.

However, *over-smoothing* and *over-fitting* problems of JD-GMM method have been reported in many studies [13], [7], [9], [11]. To address the over-fitting problem caused by the full covariance estimation, in [7], partial least square regression method is combined with Gaussian mixture model to replace the transformation matrix estimated by the full covariance matrix, while keeping the mean vectors of the original JD-GMM vector in conversion function. This combination, however, cannot avoid the over-smoothing caused by the statistical average during estimating the mean vectors of JD-GMM. One of the successful methods for reducing over-smoothing problem is the local linear transformation method [9]. In this method, each frame has its own transformation matrix, which is estimated from its K-nearest neighbourhood frame pairs in terms of Euclidean distance in the training data. One problem of this method is that when the number of nearest neighbourhood is small, over-fitting problem will still be observed. When K is too large, however, over-smoothing problem will occur [9]. Therefore, choosing the optimal number of K is an important issue in local linear transformation method.

In this study, motivated by the local linear transformation method, we proposed a local partial least square (LPLS) regression method to avoid over-fitting and over-smoothing problems in voice conversion. Similar to local linear transformation, each testing frame has individual linear transformation matrix estimated from the testing frame's K-nearest neighbour in training data. Our strategy is then to use partial least square regression to project both source and target speech vectors to a low-dimensional space. As such, we are able to estimate a robust transformation function from limited parallel frames.

II. Baseline conversion methods

In this section, we briefly introduce the two baseline conversion methods used in this study: joint density Gaussian mixture model (JD-GMM) and local linear transformation methods.

A. Joint density Gaussian mixture model method

The first baseline method is the JD-GMM method [1], [2], [14], which employs Gaussian mixture model to model the joint probability distribution of source and target feature

vectors. This method is original proposed by Kain et al in [14], and has became a mainstream approach [2].

Give a parallel training speech corpus from source speaker $\mathbf{X}$ and target speaker $\mathbf{Y}$, dynamic time warping (DTW) is employed to align the source spectral vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N]$ and target spectral vectors $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m, \ldots, \mathbf{y}_M]$, to obtain feature vectors $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, \ldots, \mathbf{z}_T]$. Here, $\mathbf{x}_n \in \mathcal{R}^d$, $\mathbf{y}_m \in \mathcal{R}^d$, and $\mathbf{z}_t = [\mathbf{x}_n^\top, \mathbf{y}_m^\top]^\top \in \mathcal{R}^{2d}$.

With the paired spectral vectors $\mathbf{Z}$, Gaussian mixture model (GMM) is adopted to represent the joint probability density of $\mathbf{X}$ and $\mathbf{Y}$, written as follows:

$$P(\mathbf{Z}) = P(\mathbf{X}, \mathbf{Y}) = \sum_{l=1}^{L} w_l^{(z)} \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_l^{(z)}, \boldsymbol{\Sigma}_l^{(z)}), \quad (1)$$

where $\boldsymbol{\mu}_l^{(z)} = \begin{bmatrix} \boldsymbol{\mu}_l^{(x)} \\ \boldsymbol{\mu}_l^{(y)} \end{bmatrix}$ and $\boldsymbol{\Sigma}_l^{(z)} = \begin{bmatrix} \boldsymbol{\Sigma}_l^{(xx)} & \boldsymbol{\Sigma}_l^{(xy)} \\ \boldsymbol{\Sigma}_l^{(yx)} & \boldsymbol{\Sigma}_l^{(yy)} \end{bmatrix}$ are the mean vector and the covariance matrix of the l^{th} Gaussian component $\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_l^{(z)}, \boldsymbol{\Sigma}_l^{(z)})$, respectively, L is the total number of Gaussian components and $w_l^{(z)}$ is the prior probability of the l^{th} Gaussian component with $\sum_{l=1}^{L} w_l^{(z)} = 1$ constraint.

During the off-line training process, the expectation maximization (EM) algorithm is employed to estimated the parameters of the joint density Gaussian mixture model $\lambda^{(z)} = \{w_l^{(z)}, \mu_l^{(z)}, \boldsymbol{\Sigma}_l^{(z)} | l = 1, 2, \ldots, L\}$ in maximum-likelihood sense.

During the conversion phase, the joint probability density GMM model is employed to formulated a conversion function. Therefore, for each source speech feature vector $\mathbf{x}$, the conversion function $F(\mathbf{x})$ in minimum mean square error sense to predict the target speaker's feature vector $\hat{\mathbf{y}}$ is written as follows:

$$F(\mathbf{x}) = \sum_{l=1}^{L} p_l(\mathbf{x})(\boldsymbol{\mu}_l^{(y)} + \boldsymbol{\Sigma}_l^{(yx)}(\boldsymbol{\Sigma}_l^{(xx)})^{-1}(\mathbf{x} - \boldsymbol{\mu}_l^{(x)})), \quad (2)$$

where $p_l(\mathbf{x}) = \frac{w_l \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_l^x, \boldsymbol{\Sigma}_l^{xx})}{\sum_{k=1}^{L} w_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^x, \boldsymbol{\Sigma}_k^{xx})}$ is the posterior probability of the source vector $\mathbf{x}$ generating from the l^{th} Gaussian component.

We note that during the JD-GMM model parameter estimation process, the mean vector of each Gaussian component is calculated as:

$$\boldsymbol{\mu}_l^{(z)} = \frac{\sum_{t=1}^{T} p_l(\mathbf{z}_t, \lambda^{(z)}) \mathbf{z}_t}{\sum_{t=1}^{T} p_l(\mathbf{z}_t, \lambda^{(z)})}. \quad (3)$$

Similarly, the covariance matrix of each Gaussian component is obtained as:

$$\boldsymbol{\Sigma}_l^{(z)} = \frac{\sum_{t=1}^{T} p_l(\mathbf{z}_t, \lambda^{(z)})(\mathbf{z}_t - \boldsymbol{\mu}_l^{(z)})(\mathbf{z}_t - \boldsymbol{\mu}_l^{(z)})^\top}{\sum_{t=1}^{T} p_l(\mathbf{z}_t, \lambda^{(z)})} \quad (4)$$

From (3) and (4), we observe that when calculating mean and covariance for each Gaussian component, all the training samples are used, which is the so-called *statistical average*. The statistical average results in the over-smoothing of the converted speech.

B. Local linear transformation method

The equation (2) can also be presented as a linear regression model:

$$\mathbf{y}_i = \mathbf{B}\mathbf{x}_i + \varepsilon, \quad (5)$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ denote the source and target observation data for i-th frame, $\mathbf{B}$ is regression matrix or transformation matrix and ε is the regression residual.

Different from equation (2), which use all the training samples to estimate the transformation matrix, the local linear transformation (LLT), [9], uses the neighboring data to estimate the individual transformation matrix for each feature vector. We use this method as our second baseline, and following is a brief introduction.

Firstly, we select the K-nearest neighbors (KNN) in terms of Euclidean distance in $\mathbf{X}$ for the feature vector of each test frame $\mathbf{x}_i^{test}$:

$$d(\mathbf{x}_i^{test}, \mathbf{X}) = \| \mathbf{x}_i^{test} - \mathbf{X} \| \quad (6)$$

The K paired vectors in $\mathbf{Y}$ are selected simultaneously. The $\mathbf{X}_i^{KNN}$ and $\mathbf{Y}_i^{KNN}$ are:

$$\begin{aligned} \mathbf{X}_i^{KNN} &= [\mathbf{x}_{i,1},\ \mathbf{x}_{i,2}, \cdots,\ \mathbf{x}_{i,k}], \\ \mathbf{Y}_i^{KNN} &= [\mathbf{y}_{i,1},\ \mathbf{y}_{i,2}, \cdots,\ \mathbf{y}_{i,k}], \end{aligned} \quad (7)$$

where, $\mathbf{x}_{i,k}$ means the k-th nearest vector in $\mathbf{X}$ for $\mathbf{x}_i^{test}$, $\mathbf{X}_i^{KNN} \in \mathcal{R}^{d \times k}$ and $\mathbf{Y}_i^{KNN} \in \mathcal{R}^{d \times k}$.

Then, for each test feature vector $\mathbf{x}_i^{test}$, the linear transformation $\mathbf{B}_i \in \mathcal{R}^{d \times d}$ could be calculated by the neighborhood we just selected, using the linear regression model:

$$\mathbf{Y}_i^{KNN} = \mathbf{B}_i \mathbf{X}_i^{KNN} \quad (8)$$

Using least squares criterion, the $\mathbf{B}_i$ is:

$$\mathbf{B}_i = \left(\mathbf{X}_i^{KNN}(\mathbf{X}_i^{KNN})^\top\right)^{-1} \mathbf{X}_i^{KNN}(\mathbf{Y}_i^{KNN})^\top \quad (9)$$

Finally, the converted speech vector $\hat{\mathbf{y}}_i$ of source vector $\mathbf{x}_i^{test}$ is given as:

$$\hat{\mathbf{y}}_i = \mathbf{B}_i \mathbf{x}_i^{test} \quad (10)$$

Obviously, the quality of the conversion will be influence by the neighborhood selection. The converted results can be used for the selection of new K-nearest neighbors to improve the performance. We can combine the testing source $\mathbf{x}_i^{test}$ and converted results $\hat{\mathbf{y}}_i$ into a new vector set, and then choose the KNN again from the whole training data set $\mathbf{Z}$.

$$d\left(\begin{bmatrix} \mathbf{x}_i^{test} \\ \hat{\mathbf{y}}_i \end{bmatrix}, \mathbf{Z}\right) = \left\| \begin{bmatrix} \mathbf{x}_i^{test} \\ \hat{\mathbf{y}}_i \end{bmatrix} - \mathbf{Z} \right\| \quad (11)$$

This reselection process is iterated till the neighborhoods determined in consecutive steps become virtually identical or sufficiently similar.

III. Proposed local partial least square regression method

From section II, we note that the local linear transformation method uses adjacent training data to estimate the transformation matrix rather than the whole training data to avoid the over-smoothing problem. The size of the neighborhood, however, affects the performance. In order to obtain a robust transformation matrix, partial least square regression is employed to relax the KNN selection constraint.

A. Basic model of partial least square

Compare with the linear regression in (5), partial least square (PLS) [15] is a regression method used to find a relationship between source and target feature vectors in a new low-dimensional space. The model of PLS is written as follows:

$$\mathbf{X} = \mathbf{R}\mathbf{W}^\top + \mathbf{E}, \tag{12}$$

$$\mathbf{Y} = \mathbf{Q}\mathbf{U}^\top + \mathbf{F}, \tag{13}$$

where $\mathbf{R}$ and $\mathbf{Q}$ are the factor loading matrices of source and target vectors; $\mathbf{W}^\top$ and $\mathbf{U}^\top$ are low-dimensional representation matrices for source $\mathbf{X}$ and target $\mathbf{Y}$, respectively. $\mathbf{E}$ and $\mathbf{F}$ are residual components. Here, $\mathbf{R} \in \mathcal{R}^{d\times h}$, $\mathbf{Q} \in \mathcal{R}^{d\times h}$, $\mathbf{W}^\top \in \mathcal{R}^{h\times T}$, $\mathbf{U}^\top \in \mathcal{R}^{h\times T}$, $\mathbf{X} \in \mathcal{R}^{d\times T}$ and $\mathbf{Y} \in \mathcal{R}^{d\times T}$. d is the dimension of feature vector, and h is the number of PLS components. If $h = d$, which means the number of variables is set to the number of predictors, PLS becomes equivalent to standard linear multivariate regression as that in (5) and (9).

Usually, the number of PLS components h is lower than the dimension of the feature vector d. Therefore, it is able to produce a robust transformation with a small amount of training data. In this regard, PLS regression is suitable for using limited data to estimate a transformation matrix.

B. Combining PLS with LLT

As described in section 3.A, PLS is able to achieve a good performance given limited training observations. In order to prevent the over-fitting problem caused in LLT method, we propose to combine PLS with LLT. Specifically, PLS could be used to substitute the least squares solution used in LLT method to obtain the transformation matrix of each testing frame. In this case, the K-nearest neighbors, $\mathbf{X}_i^{KNN}$ and $\mathbf{Y}_i^{KNN}$, obtained in the first phase of LLT mentioned in section 2.B, are used as input for PLS method. Different to [9], where the KNN selection is based on perceptual evaluation, we will use objective evaluation methods to choose the neighborhood.

C. SIMPLS Algorithm

Several variant methods could be used for solving the PLS regression problem. In this paper, we use the SIMPLS (simple partial least squares) algorithm proposed by de Jong [16]. This algorithm could provide faster computation speed, as the weight factors can be obtained without matrix inverses. Following is a brief description of the algorithm.

function SIMPLS $(\mathbf{X}^{KNN}, \mathbf{Y}^{KNN}, h)$

$\mathbf{X} = (\mathbf{X}^{KNN})^\top$
$\mathbf{Y} = (\mathbf{Y}^{KNN})^\top$
$\mathbf{Y}_0 = \mathbf{Y} -$ MEAN$(\mathbf{Y})$
$\mathbf{S} = \mathbf{X}^\top \times \mathbf{Y}_0$
for $i = 1, \cdots, h$ **do**
  $\mathbf{q} =$ dominant eigenvector of $\mathbf{S}^\top \times \mathbf{S}$
  $\mathbf{r} = \mathbf{S} \times \mathbf{q}$
  $\mathbf{w} = \mathbf{X} \times \mathbf{r}$
  $\mathbf{w} = \mathbf{w} -$ MEAN$(\mathbf{w})$
  $\|\mathbf{w}\| =$ SQRT$(\mathbf{w}^\top \times \mathbf{w})$
  $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$
  $\mathbf{r} = \mathbf{r}/\|\mathbf{w}\|$
  $\mathbf{p} = \mathbf{X}^\top \times \mathbf{w}$
  $\mathbf{q} = \mathbf{Y_0}^\top \times \mathbf{w}$
  $\mathbf{u} = \mathbf{Y_0} \times \mathbf{q}$
  $\mathbf{v} = \mathbf{p}$
  if $i > 1$ **then**
    $\mathbf{v} = \mathbf{v} - \mathbf{V} \times (\mathbf{V}^\top \times \mathbf{p})$
    $\mathbf{u} = \mathbf{u} - \mathbf{W} \times (\mathbf{W}^\top \times \mathbf{u})$
  end if
  $\mathbf{v} = \mathbf{v}/$ SQRT $(\mathbf{v}^\top \times \mathbf{v})$
  $\mathbf{S} = \mathbf{S} - \mathbf{v} \times (\mathbf{v}^\top \times \mathbf{S})$
  Store $\mathbf{r}, \mathbf{w}, \mathbf{p}, \mathbf{q}, \mathbf{u}$ and $\mathbf{v}$ as i-th columns of matrices $\mathbf{R}, \mathbf{W}, \mathbf{P}, \mathbf{Q}, \mathbf{U}$ and $\mathbf{V}$, respectively
end for
Then, the regression matrix can be obtained by $\mathbf{B} = \mathbf{Q} \times \mathbf{R}^\top$.

IV. Experiments

To evaluate the performance of our proposed method, several experiments were conducted, with baseline methods for comparison.

A. Acoustic Data

The VOICE database [17] with the speech signal down-sampled from 22.5 kHz to 16 kHz is used in our experiments. Four speaker pairs: male-to-male, male-to-female, female-to-male and female-to-female are selected from the database. The STRAIGHT system [18] is used to extract spectral envelope, which is represented by mel-cepstral coefficients (MCCs) with order 24.

A parallel set of 20 sentences aligned by dynamic time warping are used as training data, and another 20 sentences are used for objective evaluations. The results are averaged over all the conversion pairs.

B. Model Settings

In order to evaluate the performance of the proposed method, the results of JD-GMM, PLS and LLT methods are also reported as references.

- *JD-GMM*: It is the mainstream method as described in section 2.A. The number of Gaussian components is set to 64.

- *LLT*: As the conversion quality varies between different numbers of neighborhood, we evaluate the system using KNN with K from 50 to 300 with the step 10. Another factor which will influence the transform performance is the number of iterations to choose K-nearest neighbors in (11). Our results show that one iteration is enough to find nearest neighbors which give the lowest distortion. Thus, in this paper, we just report the results for one iteration.
- *PLS*: The whole training data are used to learn the transformation matrix from source feature to target feature. By changing the number of components, we can find an optimal latent PLS number, which yield a low distortion error.
- *LPLS* (proposed): Using local data and partial least square regression method to estimate the transformation for each frame as described in section 3.

The spectral envelop is converted using above conversion methods, with fundamental frequency converted by equalizing the means and variances of source and target speakers in log-scale.

C. Evaluation Methods

The mel-cepstral distortion (MCD) between the target and converted mel-cepstral is used as the objective evaluation measure. The following equation is the MCD for n-th frame:

$$\text{MCD[dB]} = \frac{10}{\ln 10}\sqrt{2\sum_{i=1}^{24}(c_{n,i} - c_{n,i}^{conv})^2}, \qquad (14)$$

where $c_{n,i}$ and $c_{n,i}^{conv}$ are the i-th dimension target and converted MCCs in frame n, respectively. A lower MCD value indicates smaller distortion.

To evaluate the conversion performance, the correlation coefficient is also calculated between target and converted MCC parameters. Different to the MCD calculation which calculated by frame, the correlation coefficient was calculated for each dimension defined as follows:

$$\gamma_i = \frac{\sum_{n=1}^{N}(c_{n,i} - \overline{c_i})(c_{n,i}^{conv} - \overline{c_i^{conv}})}{\sqrt{\sum_{n=1}^{N}(c_{n,i} - \overline{c_i})^2}\sqrt{\sum_{n=1}^{N}(c_{n,i}^{conv} - \overline{c_i^{conv}})^2}}, \qquad (15)$$

where $c_{n,i}$ and $c_{n,i}^{conv}$ are the i-th dimension target and converted MCCs in n-th frame; $\overline{c_i}$ and $\overline{c_i^{conv}}$ denote the mean value of the target and converted MCCs in i-th dimension, respectively. Here, higher correlation coefficient means higher similarity between the target and converted MCCs.

We report the distortion and correlation coefficient results by averaging all the conversion pairs.

D. Objective Results

1) The Effect of number of PLS Components: We first evaluate the effect of PLS components number in LPLS method and PLS method. The number of KNN was fixed as 200. The number of PLS components varies from 1 to 24 (since the MCC feature is 24 order). The results of JD-GMM method, PLS and LLT methods were also shown as references.

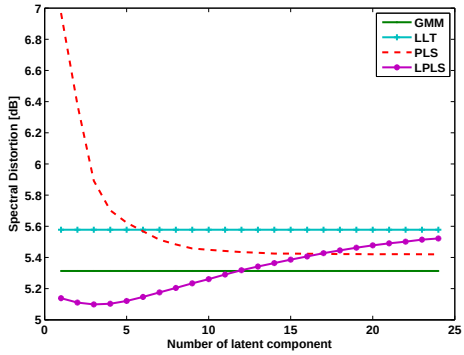

Fig. 1. Spectral distortion as function of number of latent components.

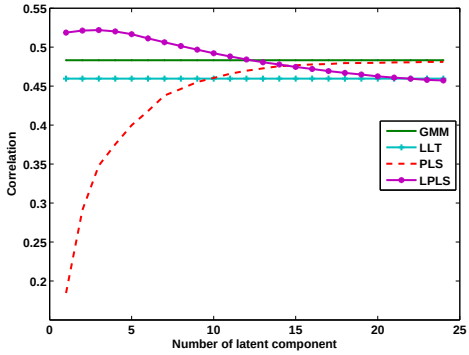

Fig. 2. The correlation coefficient as function of number of latent components.

Fig. 1 depicts the spectral distortion results. The LPLS method yield a lower error than reference methods, when the number of latent components is smaller than 11, and the optimal number of latent components for LPLS method is 3 with the lowest distortion 5.10 dB. This performance is much lower than the results of the two baseline method: JD-GMM method (5.31 dB) and LLT method (5.58 dB) respectively. With different amount of training data, the best result of PLS method is 5.42 dB for the number of latent components around 18. This result is also 0.32 dB higher than the result of our proposed method. As the latent components number increased from 3 to 24, the distortion is getting higher and higher. This occurs as the training data used to estimate the transformation matrix is limited. When the number of latent components equals to the feature dimension (24), the distortion is almost the same as using LLT method.

Correlation coefficient results are presented in Fig. 2. In accordance with the results of MCDs, with 3 latent components, the LPLS method achieved highest correlation coefficient (0.522), which is 0.039 and 0.062 higher than the results of JD-GMM method and LLT method, and 0.042 higher than the best result of PLS. The performance is getting worse when the latent components number increased above 4.

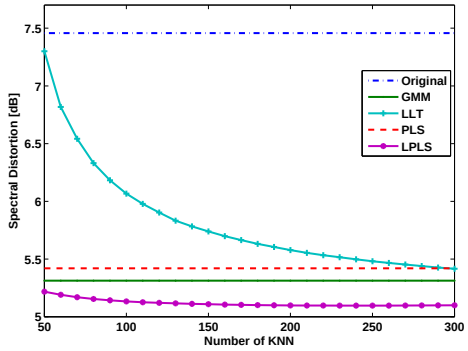

Fig. 3. Spectral distortion as function of number of nearest neighbor.

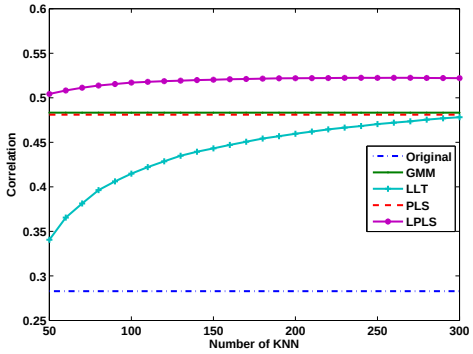

Fig. 4. The correlation coefficient as function of number of nearest neighbor.

2) The Effect of number of KNN: We then evaluate the effect of the number of KNN vectors in LPLS and LLT methods, and the spectral distortion and correlation coefficient were calculated by varying the number of KNN in 10 steps from 50 to 300. According to the results of previous experiments, the PLS components number for LPLS method and PLS method were fixed as 3 and 18 respectively. The results were averaged over four conversion pairs and presented with results of JD-GMM and original data together.

Fig. 3 indicates how the number of KNN affect the spectral distortion results for different methods. With LLT method, While the number of KNN is very small, as 50 for example, the spectral distortion is almost the same as the result calculated directly by the source and target features, which means the method does not work; Contrast with the LLT method, LPLS method is performed well with limited training observations. With 50 KNN, the result is already better than JD-GMM method and PLS method.

The correlation coefficient results with different number of KNN were shown Fig.4. It shows the similar as the results of MCDs. As in both LLT and LPLS methods, the correlation coefficient increases, with the number of KNN grows from 50 to 200, and the result of LPLS method become stable after 200, whereas the result of LLT continues increase.

V. Conclusions

In this paper, we have proposed a local partial least square (LPLS) method for voice conversion. The use of KNN data can avoid the over-smoothing problem as described in section 2. In addition, partial least square regression is able to produce more robust transformation function with limited training data. Therefore, the proposed LPLS method is capable to balance the over-smoothing and over-fitting problems in conventional methods. The experimental results indicate that our proposed method outperforms three baseline methods in terms of objective evaluation.

References

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

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

[3] Chung-Hsien Wu, Chi-Chun Hsia, Te-Hsien Liu, and Jhing-Fa Wang, "Voice conversion using duration-embedded Bi-HMMs for expressive speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1109–1116, 2006.

[4] Elina E Helander and Jani Nurminen, "A novel method for prosody prediction in voice conversion," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 4, pp. IV–509.

[5] Zhi-Zheng Wu, Tomi Kinnunen, Eng Siong Chng, and Haizhou Li, "Text-independent F0 transformation with non-parallel data for voice conversion," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[6] Damien Lolive, Nelly Barbot, and Olivier Boeffard, "Pitch and duration transformation with non-parallel data," *Speech Prosody 2008*, pp. 111–114, 2008.

[7] Elina Helander, Tuomas Virtanen, Jani Nurminen, and Moncef Gabbouj, "Voice conversion using partial least squares regression," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 5, pp. 912–921, 2010.

[8] Zhizheng Wu, Tomi Kinnunen, Eng Siong Chng, and Haizhou Li, "Mixture of factor analyzers using priors from non-parallel speech for voice conversion," *IEEE SIGNAL PROCESSING LETTERS*, vol. 19, no. 12, 2012.

[9] Victor Popa, Hanna Silen, Jani Nurminen, and Moncef Gabbouj, "Local linear transformation for voice conversion," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4517–4520.

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

[11] Elina Helander, Hanna Silén, Tuomas Virtanen, and Moncef Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 806–817, 2012.

[12] Zhizheng Wu, Eng Siong Chng, and Haizhou Li, "Conditional restricted boltzmann machine for voice conversion," in *the first IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*. IEEE, 2013.

[13] Yining Chen, Min Chu, Eric Chang, Jia Liu, and Runsheng Liu, "Voice conversion with smoothed GMM and MAP adaptation," in *Eurospeech-2003*, 2003, pp. 2413–2416.

[14] Alexander Kain and Michael W Macon, "Spectral voice conversion for text-to-speech synthesis," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. IEEE, 1998, vol. 1, pp. 285–288.

[15] Roman Rosipal and Nicole Krämer, "Overview and recent advances in partial least squares," in *Subspace, Latent Structure and Feature Selection*, pp. 34–51. Springer, 2006.

[16] Sijmen de Jong, "SIMPLS: an alternative approach to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251–263, 1993.
[17] Alexander Blouke Kain, *High resolution voice transformation*, Ph.D. thesis, Rockford College, 2001.
[18] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain de Cheveigné, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.