ROBUST FEATURE EXTRACTION USING KERNEL PCA

Tetsuya Takiguchi and Yasuo Ariki

Department of Computer and System Engineering
Kobe University, Japan
takigu@kobe-u.ac.jp

ABSTRACT

We investigate a robust speech feature extraction method using kernel PCA (Principal Component Analysis). Kernel PCA has been suggested for various image processing tasks requiring an image model such as, e.g., denoising, where a noise-free image is constructed from a noisy input image [1].

Much research for robust speech feature extraction has been done, but it is difficult to completely remove the non-stationary noise or reverberation. The most commonly used noise-removal techniques are based on the spectral-domain operation, and then for the speech recognition, MFCC (Mel Frequency Cepstral Coefficient) is computed, where DCT (Discrete Cosine Transform) is applied to the mel-scale filter bank output. In this paper, we propose robust feature extraction based on kernel PCA instead of DCT, where the main speech element is projected onto low-order features, while noise or reverberant element is projected onto high-order ones. Its effectiveness is confirmed by word recognition experiments on reverberant speech.

1. INTRODUCTION

In hands-free speech recognition, one of the key issues for practical use is the development of technologies that allow accurate recognition of noisy and reverberant speech. Current speech recognition systems are capable of achieving impressive performance in clean acoustic environments. However, if the user speaks at a distance from the microphone, the recognition accuracy is seriously degraded by the influence of noise and reverberation.

In current speech recognition technology, MFCC (Mel Frequency Cepstral Coefficient) has been widely used. The feature is derived from the mel-scale filter bank output by DCT (Discrete Cosine Transform). The low-order MFCCs account for the slowly changing spectral envelope, while the high-order ones describe the fast variations of the spectrum. Therefore a large number of MFCCs is not used for the speech recognition, because we are only interested in the spectral envelope, not in the fine structure.

To solve problems caused by noise and reverberation, many methods have been presented in robust speech recognition (e.g. [2, 3, 4, 5, 6, 7]). But it is difficult to completely remove the non-stationary noise or reverberation. The most commonly used noise-removal techniques are based on the spectral-domain operation, and then for the speech recognition, MFCC (Mel Frequency Cepstral Coefficient) is computed, where DCT is applied to the mel-scale filter bank output.

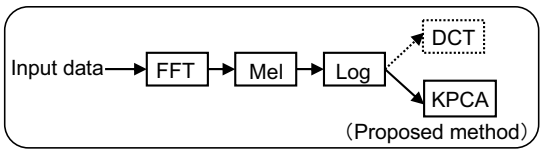

Fig. 1. Feature Extraction Using Kernel PCA

A feature extraction approach using kernel PCA has been also proposed in [8][9], where the kernel PCA was applied only to the low-order MFCCs which account for the spectral envelope. In this paper we investigate robust feature extraction using kernel PCA instead of DCT, where kernel PCA is applied to the mel-scale filter bank output (Fig. 1), because we expect that kernel PCA will project the main speech element onto low-order features, while noise (reverberant) element onto high-order ones. Our recognition results show that the use of kernel PCA instead of DCT provides better performance for reverberant speech.

2. FEATURE EXTRACTION USING KERNEL PCA

PCA is a powerful technique for extracting structure from possibly high-dimensional data sets. But it is not effective for data with nonlinear structure. In kernel PCA, the input data with nonlinear structure is transformed into a higher-dimensional feature space with linear structure, and then linear PCA is performed in the high-dimensional space [10].

Given the mel-scale filter bank output $\mathbf{x}_j$ at j-frame, the covariance matrix is defined as

$$C = \frac{1}{N}\sum_{j=1}^{N} \bar{\mathbf{\Phi}}(\mathbf{x}_j)\bar{\mathbf{\Phi}}(\mathbf{x}_j)^T, \quad (1)$$

$$\bar{\mathbf{\Phi}}(\mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_j) - \frac{1}{N}\sum_{j=1}^{N}\mathbf{\Phi}(\mathbf{x}_j), \tag{2}$$

where the total number of frames is N, and $\mathbf{\Phi}$ is a nonlinear map.

$$\mathbf{\Phi} : \mathbf{R}^d \rightarrow \mathbf{R}^\infty \tag{3}$$

Note that the data in the high-dimensional space could have an arbitrarily large, possibly infinite, dimensionality, and d is the dimension of $\mathbf{x}$.

We now have to find eigenvalues λ and eigenvectors $\mathbf{v}$ satisfying

$$\lambda\mathbf{v} = C\mathbf{v}, \tag{4}$$

$$\lambda(\bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot\mathbf{v}) = (\bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot C\mathbf{v}), \quad k = 1,\ldots,N \tag{5}$$

Also, there exist coefficients α_i such that

$$\mathbf{v} = \sum_{i=1}^{N}\alpha_i\bar{\mathbf{\Phi}}(\mathbf{x}_i). \tag{6}$$

Substituting (1) and (6) in (5), we get for the left side of the equation

$$\begin{aligned}\lambda(\bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot\mathbf{v}) &= \lambda\sum_i \alpha_i\bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot\bar{\mathbf{\Phi}}(\mathbf{x}_i)\\ &= \lambda\sum_i \alpha_i\bar{K}_{ki},\end{aligned} \tag{7}$$

where

$$\bar{K}_{ki} = \bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot\bar{\mathbf{\Phi}}(\mathbf{x}_i). \tag{8}$$

Also, for the right side of the equation

$$\begin{aligned}&\bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot C\mathbf{v}\\ &= \bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot\frac{1}{N}\sum_j\bar{\mathbf{\Phi}}(\mathbf{x}_j)\bar{\mathbf{\Phi}}(\mathbf{x}_j)^T\sum_i\alpha_i\bar{\mathbf{\Phi}}(\mathbf{x}_i)\\ &= \bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot\frac{1}{N}\sum_i\alpha_i\left\{\sum_j\bar{\mathbf{\Phi}}(\mathbf{x}_j)\bar{\mathbf{\Phi}}(\mathbf{x}_j)^T\bar{\mathbf{\Phi}}(\mathbf{x}_i)\right\}\\ &= \frac{1}{N}\sum_i\alpha_i\left[\bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot\left\{\sum_j\bar{\mathbf{\Phi}}(\mathbf{x}_j)\bar{\mathbf{\Phi}}(\mathbf{x}_j)^T\bar{\mathbf{\Phi}}(\mathbf{x}_i)\right\}\right]\\ &= \frac{1}{N}\sum_i\alpha_i\sum_j\left\{\bar{\mathbf{\Phi}}(\mathbf{x}_k)\cdot\bar{\mathbf{\Phi}}(\mathbf{x}_j)\right\}\left\{\bar{\mathbf{\Phi}}(\mathbf{x}_j)\cdot\bar{\mathbf{\Phi}}(\mathbf{x}_i)\right\}\\ &= \frac{1}{N}\sum_i\alpha_i\sum_j\bar{K}_{kj}\bar{K}_{ji}.\end{aligned} \tag{9}$$

Thus we get

$$\begin{aligned}N\lambda\alpha &= \bar{\mathbf{K}}\alpha\\ \hat{\lambda}\alpha &= \bar{\mathbf{K}}\alpha.\end{aligned} \tag{10}$$

Consequently, we only need to diagonalize $\bar{\mathbf{K}}$ which is computed as follows.

$$\begin{aligned}\bar{K}_{ij} &= \bar{\mathbf{\Phi}}(\mathbf{x}_i)\cdot\bar{\mathbf{\Phi}}(\mathbf{x}_j)\\ &= (\mathbf{\Phi}(\mathbf{x}_i) - \frac{1}{N}\sum_{m=1}^{N}\mathbf{\Phi}(\mathbf{x}_m))\\ &\quad\cdot(\mathbf{\Phi}(\mathbf{x}_j) - \frac{1}{N}\sum_{n=1}^{N}\mathbf{\Phi}(\mathbf{x}_n))\\ &= \mathbf{\Phi}(\mathbf{x}_i)\cdot\mathbf{\Phi}(\mathbf{x}_j) - \frac{1}{N}\sum_{m=1}\mathbf{\Phi}(\mathbf{x}_m)\cdot\mathbf{\Phi}(\mathbf{x}_j)\\ &\quad -\frac{1}{N}\sum_{n=1}\mathbf{\Phi}(\mathbf{x}_n)\cdot\mathbf{\Phi}(\mathbf{x}_i)\\ &\quad +\frac{1}{N^2}\sum_{m,n=1}\mathbf{\Phi}(\mathbf{x}_m)\cdot\mathbf{\Phi}(\mathbf{x}_n)\\ &= K_{ij} - \frac{1}{N}\sum_{m=1}1_{im}K_{mj} - \frac{1}{N}\sum_{n=1}K_{in}1_{nj}\\ &\quad +\frac{1}{N^2}\sum_{m,n=1}1_{im}K_{mn}1_{nj}\end{aligned} \tag{11}$$

$$K_{ij} = \mathbf{\Phi}(\mathbf{x}_i)\cdot\mathbf{\Phi}(\mathbf{x}_j) \tag{12}$$

$$1_{ij} = 1 \quad \text{for all } i,j \tag{13}$$

Using the $N\times N$ matrix $(\mathbf{1}_N)_{ij} := 1/N$, we get the more compact expression

$$\bar{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N. \tag{14}$$

We thus can compute $\bar{\mathbf{K}}$ from $\mathbf{K}$, and then solve the eigenvalue problem (10).

Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ denote the eigenvalues, and $\alpha^{(1)}$, $\cdots$, $\alpha^{(N)}$ the corresponding complete set of eigenvectors, with λ_p being the first nonzero eigenvalue. We normalize $\alpha^{(p)}$, $\cdots$, $\alpha^{(N)}$ by requiring that the corresponding vectors are normalized:

$$\mathbf{v}^{(l)}\cdot\mathbf{v}^{(l)} = 1, \ \text{ for all } l = p,\cdots,N \tag{15}$$

From (6) and (10) we get

$$\begin{aligned}1 &= \sum_{i,j}^{N}\alpha_i^{(l)}\alpha_j^{(l)}(\mathbf{\Phi}(\mathbf{x}_i)\cdot\mathbf{\Phi}(\mathbf{x}_j))\\ &= \sum_{i,j}^{N}\alpha_i^{(l)}\alpha_j^{(l)}K_{ij}\\ &= (\alpha^{(l)}\cdot\bar{\mathbf{K}}\alpha^{(l)})\\ &= \hat{\lambda}_l(\alpha^{(l)}\cdot\alpha^{(l)}).\end{aligned} \tag{16}$$

Therefore, we finally normalize α by

$$\hat{\alpha}^{(l)} = \frac{\alpha^{(l)}}{\sqrt{\hat{\lambda}_l}}. \tag{17}$$

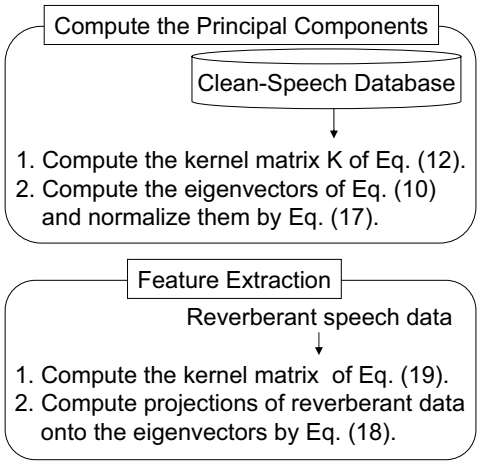

Fig. 2. Procedure of Feature Extraction

Next, for feature extraction, we project test data $\mathbf{y}$ onto eigenvectors $\mathbf{v}^{(l)}$ in the high-dimensional space.

$$\begin{aligned}(\mathbf{v}^{(l)} \cdot \bar{\mathbf{\Phi}}(\mathbf{y})) &= \sum_{i=1}^{N} \hat{\alpha}_i^{(l)} (\bar{\mathbf{\Phi}}(\mathbf{x}_i) \cdot \bar{\mathbf{\Phi}}(\mathbf{y})) \\ &= \sum_{i=1}^{N} \hat{\alpha}_i^{(l)} \bar{K}^{test}(\mathbf{x}_i, \mathbf{y}) \end{aligned} \quad (18)$$

Similar to (11), we can compute $\bar{K}^{test}$ from K^{test}.

$$\begin{aligned}\bar{K}_{ij}^{test} &= \left(\mathbf{\Phi}(\mathbf{y}_i) - \frac{1}{N}\sum_{m=1}^{N}\mathbf{\Phi}(\mathbf{x}_m)\right) \\ &\quad \cdot \left(\mathbf{\Phi}(\mathbf{x}_j) - \frac{1}{N}\sum_{n=1}^{N}\mathbf{\Phi}(\mathbf{x}_n)\right) \end{aligned} \quad (19)$$

$$\bar{\mathbf{K}}^{test} = \mathbf{K}^{test} - \mathbf{1}'_N\mathbf{K} - \mathbf{K}^{test}\mathbf{1}_N + \mathbf{1}'_N\mathbf{K}\mathbf{1}_N \quad (20)$$

Here $\mathbf{1}'_N$ is the $L \times N$ matrix with all entries equal to $1/N$, and the total number of frames for the test data is L. The procedure of the feature extraction is summarized in Fig. 2.

3. RECOGNITION EXPERIMENT

3.1. Experimental Conditions

The new feature extraction method was evaluated on reverberant speech recognition tasks. Reverberant speech was simulated by a linear convolution of clean speech and impulse response. The impulse response was taken from the RWCP sound scene database [11]. The reverberation time was 470 msec. The distance to the microphone was about 2 m. The size of the recording room was about 6.7 m $\times$ 4.2 m (width $\times$ depth).

In order to compute the matrix $\mathbf{K}$, it would be necessary to use all the training data. But it is not realistic in terms of the cost of the computation. Therefore, N = 2500 frames were randomly picked up from the training data. Then, in this experiments, we used polynomial kernel function.

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p \quad (21)$$

The speech signal was sampled at 12 kHz and windowed with a 32-msec Hamming window every 8 msec. The models of 54 context-independent phonemes were trained by using 2,620 words in the ATR Japanese speech database for the speaker-dependent HMM. Each HMM has three states and three self-loops, and each state has four Gaussian mixture components. The tests were carried out on 1,000-word recognition tasks, and three males spoke the 1,000 words. The baseline recognition rate was 63.9%, where 16-order MFCCs and their delta coefficients were used as feature vectors.

3.2. Experimental Results

Figure 3 shows the recognition rates using kernel PCA (p = 1 in polynomial function). As can be seen from this figure, the use of kernel PCA instead of DCT improves the recognition rates from 63.9% to 75.0%. Here, in the new feature extraction, kernel PCA was applied to 32 mel-scale filter bank output, and then the delta coefficients were also computed. Figure 4 shows the recognition rates using kernel PCA (p = 2 in polynomial function). These results clearly show that the use of kernel PCA instead of DCT achieves good performance.

Next, we applied kernel PCA to 16-order MFCCs which account for the spectral envelope [8][9]. The recognition rate improved from 63.9% to 67.8%. As can be seen from Figure 4, a further improvement was obtained by the new method, where kernel PCA was applied to the mel-scale filter bank output. This is because it can expect that kernel PCA in the spectral domain will project the main speech element onto low-order features, while reverberant element onto high-order ones.

Figure 5 shows the recognition rates for the clean speech using kernel PCA. The recognition rate with the new feature extraction was 97.6%, and the baseline performance using DCT was 97.3%. In the clean environments, the experiment results indicate that the new method achieves almost the same performance as that by DCT.

4. SUMMARY

This paper has described a robust feature extraction technique using kernel PCA instead of DCT, where kernel PCA is applied to the mel-scale filter bank output. It can expect that kernel PCA will project the main speech element onto low-order features, while reverberant (noise) element onto high-order ones. From our recognition results, it is shown that the use of

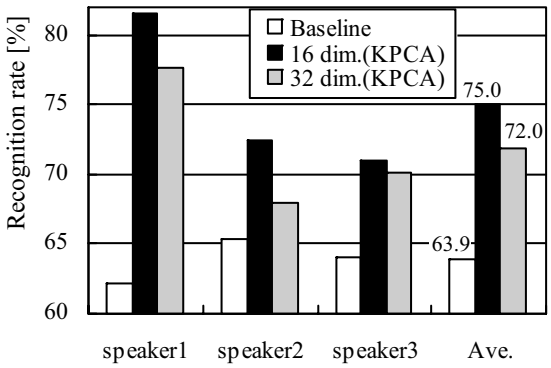

Fig. 3. Recognition rates for the reverberant speech (reverberation time: 470 msec) by the proposed method. ($p = 1$ in polynomial function)

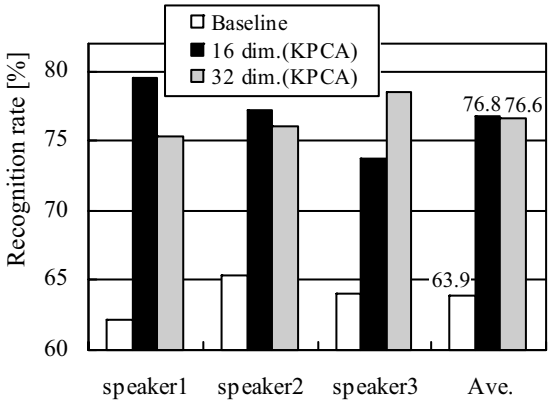

Fig. 4. Recognition rates for the reverberant speech (reverberation time: 470 msec) by the proposed method. ($p = 2$ in polynomial function)

kernel PCA instead of DCT provides better performance for reverberant speech (reverberation time: 470 msec).

5. REFERENCES

[1] S. Mika, B. Scholkopf, A.J. Smola, K.-R. Muller, M. Scholz, and G. Ratsch, "Kernel PCA and de-noising in feature spaces," In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, Advances in Neural Information Processing Systems 11, pp. 536–542, MIT Press, 1999.

[2] C. Avendano, S. Tivrewala, and H. Hermansky, "Multiresolution channel normalization for ASR in reverberant environments," Eurospeech, pp. 1107-1110, 1997.

[3] U. H. Yapanel and J. H. L. Hansen, "A New Perspective on Feature Extraction for Robust In-Vehicle Speech Recognition," Eurospeech, pp. 1281-1284, 2003.

[4] B. J. Shannon and K. K. Paliwal, "Influence of Auto-

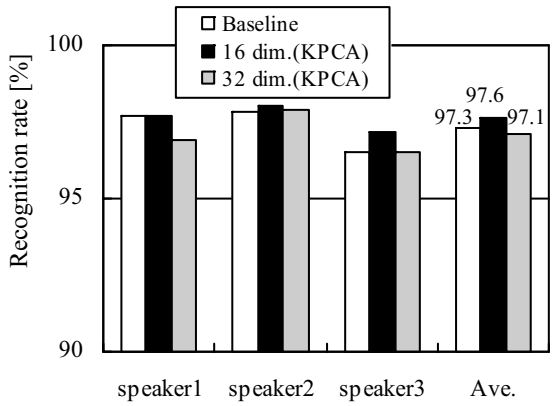

Fig. 5. Recognition rates for the clean speech by the proposed method. ($p = 2$ in polynomial function)

correlation Lag Ranges on Robust Speech Recognition," ICASSP, pp. 545-548, 2005.

[5] W. Li, K. Itou, K. Takeda and F. Itakura, "Two-Stage Noise Spectra Estimation and Regression Based In-Car Speech Recognition Using Single Distant Microphone," ICASSP, pp. 533-536, 2005.

[6] M. Fujimoto, S. Nakamura, "Particle Filter Based Non-Stationary Noise Tracking for Robust Speech Recognition," ICASSP, pp. 257-260, 2005.

[7] K. Kinoshita, T. Nakatani and M. Miyoshi, "Efficient Blind Dereverberation Framework for Automatic Speech Recognition," Interspeech, pp. 3145-3148, 2005.

[8] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "On the Use of Kernel PCA for Feature Extraction in Speech Recognition," IEICE Trans. Inf. & Syst., Vol. E87-D, No. 12, pp. 2802-2811, 2004.

[9] A. Lima, H. Zen, Y. Nankaku, K. Tokuda, T. Kitamura and F. G. Resende, "Applying Sparse KPCA for Feature Extraction in Speech Recognition," IEICE Trans. Inf. & Syst., Vol. E88-D, No. 3, pp. 401-409, 2005.

[10] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Computation, Vol. 10, pp. 1299-1319, 1998.

[11] S. Nakamura, K. Hiyane, F. Asano, T.Nishiura, T. Yamada, "Acoustical Sound Database in Real Environments for Sound Scene Understanding and Hands-Free Speech Recognition," Proceedings of International Conference on Language Resources and Evaluation, Vol. 2, pp. 965-968, 2000.