Kernel PCA for Speech Enhancement

Christina Leitner, Franz Pernkopf, Gernot Kubin

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Graz, Austria
christina.leitner@tugraz.at, pernkopf@tugraz.at, gernot.kubin@tugraz.at

Abstract

In this paper, we apply kernel principal component analysis (kPCA), which has been successfully used for image denoising, to speech enhancement. In contrast to other enhancement methods which are based on the magnitude spectrum, we rather apply kPCA to complex spectral data. This is facilitated by Gaussian kernels. In the experiments, we show good noise reduction with few artifacts for noise corrupted speech at different SNR levels using additive white Gaussian noise. We compared kPCA with linear PCA and spectral subtraction and evaluated all algorithms with perceptually motivated quality measures.

Index Terms: Kernel PCA, speech enhancement

1. Introduction

Speech enhancement has many applications such as speech communications and speech recognition. Algorithms for speech enhancement can be divided into three main classes: Spectral-subtractive algorithms, statistical-model-based algorithms and subspace algorithms [1].

Subspace methods – which we consider in this work – are based on the assumption that a speech signal only lives in a subspace of a given signal space. Noise on the other hand is distributed over the entire space. Noise reduction can therefore be achieved by restricting the enhanced signal only to the speech subspace and setting other signal components to zero. This is usually realized by Principal Component Analysis (PCA) which performs an eigenvalue decomposition (EVD) on the estimated covariance matrices of the clean speech and the noise signal. In addition, the speech signal is filtered to reduce noisy components in the subspace of speech.

Speech enhancement is always a tradeoff between the suppression of noise and the introduction of speech distortion caused by the algorithm. Many speech enhancement algorithms like, e.g., spectral subtraction suffer from musical noise. Furthermore, these algorithms work on the magnitude spectrum only. The phase of the original noisy signal is used for the final transformation to time domain. Although a minor problem compared to the occurrence of musical noise, this can reduce the speech quality at low signal to noise ratios [1].

In this paper, we directly perform de-noising on the complex coefficients of the short time Fourier transform (STFT) using kernel PCA (kPCA). Hence, we do not have to rely on the phase of the noisy signal when transforming back to time domain. Kernel PCA is a non-linear extension of linear PCA. It performs a mapping to a high-dimensional feature space and then applies PCA on the covariance matrix of the mapped data. The covariance matrix, however, is not computed explicitly. Instead the EVD is performed on the kernel matrix. Among other applications, kPCA has been successfully applied to image denoising [2, 3]. In speech processing, Takiguchi and Akiri used kPCA to extract robust features from reverberant speech [4]. In the experiments our approach shows a good performance for additive white Gaussian noise at 5 and 10 dB SNR compared to Hu and Loizou's subspace method [5] and spectral subtraction [6]. Both algorithms introduce musical noise whereas our approach generates a buzz-like artifact.

The paper is organized as follows: Section 2 introduces kernel PCA and the reconstruction of samples in input space. The implementation of the algorithm is explained in section 3. Experiments and results are presented in Section 4. Section 5 concludes the paper and gives a perspective on future work.

2. From linear PCA to Kernel PCA

Linear PCA is an orthogonal transformation of the space containing the data samples of the problem at hand. The transformed space is spanned by the eigenvectors that are found by eigenvalue decomposition of the covariance matrix estimated from the data samples. The coordinates of the data samples after transformation are referred to as principal components. Often few principal components capture most of the characteristics of the data. The directions of these components are given by the eigenvectors corresponding to large eigenvalues, as a large eigenvalue means that its eigenvector covers relevant information of the data. Several applications such as data compression and de-noising exploit this fact. For de-noising, directions with small eigenvalues are assumed to contain no information about the signal but only noise. These directions are dropped by projecting the signal onto eigenvectors corresponding to large eigenvalues.

Kernel PCA performs a non-linear transformation of the sample $\mathbf{x}$ in input space, $\mathbf{x} \in \mathbb{R}^N$, to the probably high-dimensional feature space F expressed by the map

$$\begin{aligned} \mathbf{\Phi} : \mathbb{R}^N &\rightarrow F \\ \mathbf{x} &\mapsto \mathbf{X}. \end{aligned} \tag{1}$$

PCA is then performed in this highdimensional space. The covariance matrix in feature space can be expressed as

$$\bar{\mathbf{C}} = \frac{1}{M} \sum_{j=1}^{M} \mathbf{\Phi}(\mathbf{x}_j) \mathbf{\Phi}(\mathbf{x}_j)^{\mathrm{T}}, \tag{2}$$

where $\Phi(\mathbf{x}_j)$ are the samples mapped into feature space and M

This research has been carried out in the context of the project NFN-SISE. We gratefully acknowledge funding by the Austrian Science Fund (FWF) under the project number S10604-N13.

denotes the number of samples.[1] To perform PCA, we have to solve the eigenvalue problem

$$\lambda \mathbf{V} = \bar{\mathbf{C}}\mathbf{V}. \tag{3}$$

All eigenvectors $\mathbf{V}^k$ that solve this equation must lie in the span of the $\mathbf{\Phi}$-images. Therefore we can solve the equivalent system

$$\lambda\, \mathbf{\Phi}(\mathbf{x}_k)^{\mathrm{T}}\, \mathbf{V} = \mathbf{\Phi}(\mathbf{x}_k)^{\mathrm{T}}\, \bar{\mathbf{C}}\, \mathbf{V} \quad \text{for all} \quad k = 1, \ldots, M. \tag{4}$$

Furthermore, each eigenvector $\mathbf{V}$ can be expanded as linear combination of the $\mathbf{\Phi}$'s using the coefficients $\alpha_1, \ldots, \alpha_M$

$$\mathbf{V} = \sum_{i=1}^{M} \alpha_i \mathbf{\Phi}(\mathbf{x}_i). \tag{5}$$

Substituting Eq. (2) and Eq. (5) into Eq. (4) leads to

$$\lambda \sum_{i=1}^{M} \alpha_i \mathbf{\Phi}(\mathbf{x}_k)^{\mathrm{T}} \mathbf{\Phi}(\mathbf{x}_i) = \frac{1}{M} \sum_{i=1}^{M} \alpha_i \mathbf{\Phi}(\mathbf{x}_k)^{\mathrm{T}} \sum_{j=1}^{M} \mathbf{\Phi}(\mathbf{x}_j) \mathbf{\Phi}(\mathbf{x}_j)^{\mathrm{T}} \mathbf{\Phi}(\mathbf{x}_i) \quad \text{for all} \quad k = 1, \ldots, M. \tag{6}$$

The multiplication of $\mathbf{\Phi}$-images can be expressed as kernel in terms of input samples $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i)^{\mathrm{T}} \cdot \mathbf{\Phi}(\mathbf{x}_j)$. Defining a kernel matrix $\mathbf{K}$ with entries

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) \tag{7}$$

Eq. (7) can be reformulated as

$$M\lambda \mathbf{K}\boldsymbol{\alpha} = \mathbf{K}^2 \boldsymbol{\alpha}, \tag{8}$$

where $\boldsymbol{\alpha}$ is a column vector with the entries $\alpha_1, \ldots, \alpha_M$. This system is equivalent to

$$M\lambda \boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}, \tag{9}$$

which is another eigenvalue problem. It is solved by the eigenvectors $\boldsymbol{\alpha}^k$ that equally solve Eq. (8). Requiring $\mathbf{V}^k$ to be normalized leads to the normalization condition $\lambda_k (\boldsymbol{\alpha}^k)^{\mathrm{T}} \boldsymbol{\alpha}^k = 1$.

For de-noising a test sample $\mathbf{\Phi}(\mathbf{x})$, it is projected onto the eigenvectors $\mathbf{V}^k$, $k = 1 \ldots n$, that correspond to the n largest eigenvalues. The projection can be expressed in terms of kernel functions using Eq. (5)

$$\beta_k = (\mathbf{V}^k)^{\mathrm{T}} \mathbf{\Phi}(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i^k k(\mathbf{x}, \mathbf{x}_i). \tag{10}$$

Consequently, the projected sample in feature space equals

$$\mathbf{P}_n \mathbf{\Phi}(\mathbf{x}) = \sum_{k=1}^{n} \beta_k \mathbf{V}^k, \tag{11}$$

where $\mathbf{P}_n$ denotes the projection operator. For de-noising, however, the de-noised sample $\mathbf{z}$ in input space and not the de-noised sample $\mathbf{P}_n \mathbf{\Phi}(\mathbf{x})$ in feature space is needed. As the mapping from input to feature space is non-linear, it is not guaranteed that such a sample in input space exists, and if it does it is not necessarily unique. So we try to find a sample $\mathbf{z}$ that satisfies $\mathbf{\Phi}(\mathbf{z}) = \mathbf{P}_n \mathbf{\Phi}(\mathbf{x})$. If the kernel is an invertible function $\mathbf{z}$ can be computed directly as derived in [7]. For non-invertible kernel functions like the Gaussian kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / c), \tag{12}$$

where c denotes the variance, $\mathbf{z}$ can be approximated by minimizing

$$\rho(\mathbf{z}) = \|\mathbf{\Phi}(\mathbf{z}) - \mathbf{P}_n \mathbf{\Phi}(\mathbf{x})\|^2. \tag{13}$$

This leads to an iterative update equation for $\mathbf{z}$

$$\mathbf{z}_{t+1} = \frac{\sum_{i=1}^{M} \gamma_i k(\mathbf{z}_t, \mathbf{x}_i) \mathbf{x}_i}{\sum_{i=1}^{M} \gamma_i k(\mathbf{z}_t, \mathbf{x}_i)}, \tag{14}$$

where $\gamma_i = \sum_{k=1}^{n} \beta_k \alpha_i^k$ (see also [2]). Note that the resulting pre-image $\mathbf{z}$ is always a linear combination of the input data $\mathbf{x}_i$. The algorithm is sensitive to initialization, however this can be tackled by reinitializing with different values. In our experiments, we used the noisy data sample which results in robust performance.

Until so far we have assumed that the data in feature space is centered, i.e., $\sum_{i=1}^{M} \mathbf{\Phi}(\mathbf{x}_i) = 0$. Generally this assumption does not hold and centering has to be done explicitly. Instead of centering the mapped data samples $\mathbf{\Phi}(\mathbf{x}_i)$ which are usually not computed, centering can be done by modifying the kernel matrix $\mathbf{K}$ to get the centered kernel matrix

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_M \mathbf{K} - \mathbf{K}\mathbf{1}_M + \mathbf{1}_M \mathbf{K} \mathbf{1}_M, \tag{15}$$

where $\mathbf{1}_M$ is a matrix with all entries equaling $1/M$ (for more details see [8]). All following PCA steps can be conducted on the centered kernel matrix.

For reconstruction the centering procedure has to be inverted. According to [9] this can be done by using $\tilde{\gamma}_i = \gamma_i + 1/N(1 - \sum_{j=1}^{M} \gamma_k)$ in the reconstruction equation (14).

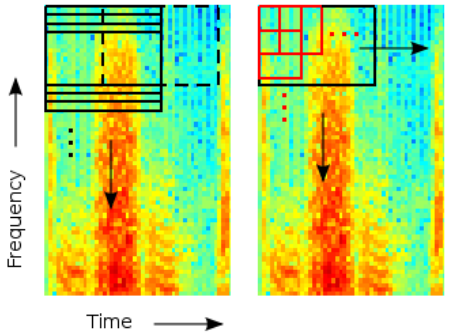

Figure 1: *Spectral detail of the clean utterance /t a sh e/. Left hand side: Extraction of frequency bands with hopsize 2. Right hand side: Extraction of 12 × 12 patches with hopsize 6.*

3. Implementation

Initially we applied kPCA on the magnitude of the STFT over the whole test utterance. Like other speech enhancement methods we used the phase of the noisy signal for final transformation into time domain. In these experiments the phase again corrupted the enhanced signal. This can be explained by the fact that if two signals are additive in the domain of the complex STFT their magnitudes are usually not [10]. Therefore, we perform kPCA on the complex values of the STFT instead of the magnitude. This avoids the usage of the phase of the noisy

[1] For the moment we assume that the data is centered, more details follow at the end of this section.

Freq. band height in patches	Frequency hop in patches
4	1
8	2
16	4
Patch size in bins	**Hop size in bins**
12×12	1
16×16	2

Table 1: *Parameters for frequency band and patch extraction.*

signal. kPCA of complex data can easily be realized by using the Gaussian kernel as it returns a real value.

To extract feature vectors from the sequence of STFTs we apply the following procedure: First, the STFT is computed from frames of 256 samples with 50% overlap. The resulting time-frequency respresentation is split into shorter segments of 0.25 seconds to avoid large kernel matrices, as large kernel matrices increase computation times. Experiments showed that the application of kPCA on the full frequency range results in a suppression of high frequencies. This can be explained by the lower energy of speech in high frequencies. To compensate for different energy levels, each time segment is processed in overlapping frequency bands as shown in Fig. 1 (left side). On each frequency band kPCA is computed separately, i.e., for each band we build an individual kernel matrix[2]. To retrieve sample vectors for kernel matrix computation each frequency band is divided into overlapping patches of size 12×12 (see Fig. 1, right side). The height of one frequency band is 8 patches, the band overlap is 4. In initial experiments we observed that better de-noising is achieved when the patches are windowed. Hence, a 2D Hamming window is applied and the patches are rearranged as vectors to obtain samples for kPCA.

For resynthesis, patches, frequency bands and time segments have to be merged properly. Patches at the same time-frequency position but of different frequency bands are averaged. Then the patches are added in an overlapping manner and weighted to compensate for windowing. For weighting the standard method described in [11] is adapted to the 2D domain. The time segments are merged, resulting in the full sequence of STFT coefficients. Then inverse Fourier transform is applied to the spectrum of each time instant and the signal is synthesized using the weighted overlap add method of [11].

4. Experiments

We tested the algorithm on recordings of six speakers (three male, three female). Each speaker uttered 20 sentences which leads to 120 sentences in total. Recordings were performed with a close-talk microphone and 16 kHz sampling frequency. White Gaussian noise was added to the recordings at 15, 10 and 5 dB SNR. The algorithm was tested at 10 dB SNR with several configurations for the time segmentation, the dimension of frequency bands, the overlap of the frequency bands, the patch size, the patch overlap, and the variance of the Gaussian kernel. The tested values are listed in Table 1. The best configuration of parameters was determined by listening to the files of two speakers (one female and one male). The performance for different SNR levels depends on the variance of the Gaussian kernel, therefore it was adapted for each condition. The values are 0.5, 1 and 2 for 15, 10 and 5 dB SNR, respectively.

[2] As the bands are overlapping, the kernel matrices partially have the same entries - this is exploited in our implementation.

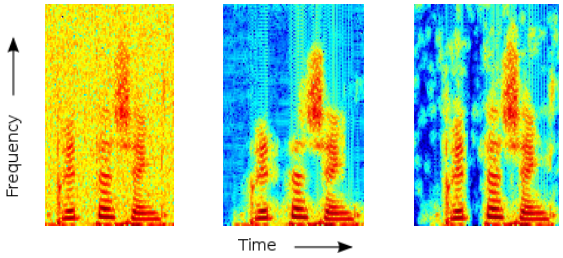

Figure 2: *Spectral details of a speech utterance corrupted by white noise at 10 dB SNR (left), its spectrogram after enhancement by kPCA (middle) and Hu and Loizou's algorithm (right).*

The resulting files were evaluated perceptually and objectively by quality measures. For comparison we applied subspace filtering as proposed by Hu and Loizou in [5] and spectral subtraction as described in [6]. Audio examples are provided on `http://www2.spsc.tugraz.at/people/chrisl/audio/` for all algorithms. Fig. 2 shows spectrogram details of one utterance: on the left corrupted by noise, in the middle enhanced by kPCA, and on the right enhanced by Hu and Loizou's algorithm.

For perceptual evaluation, we listened to files of two speakers. Noise reduction works well for all three conditions. No musical noise occurs, only a buzz-like artifact is perceivable. This can be caused by windowing and requires further investigation. At 5 dB SNR, some residual noise in the low frequency range can be perceived. This can be explained by the fact that the reconstructed samples of Eq. (14) are linear combinations of the noisy input samples.

The reconstruction from noisy samples can be avoided by a supervised procedure where analysis and de-noising are applied to different data sets. In such a procedure clean speech is used for EVD. For de-noising, the noisy speech samples are projected onto the eigenvectors of clean speech. Furthermore, the clean samples are used for reconstruction and less noise is introduced in the resulting signal.

For objective evaluation we used the following quality mea-

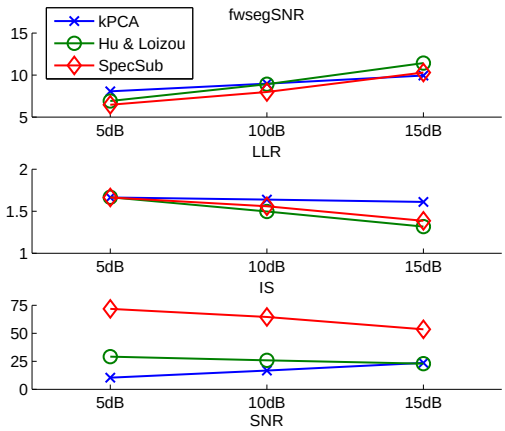

Figure 3: *Comparison between kPCA, Hu and Loizou's algorithm, and spectral subtraction using the frequency-weighted SNR (fwsegSNR), the log-likelihood ratio (LLR), and the Itakura-Saito distance (IS) for different SNR values.*

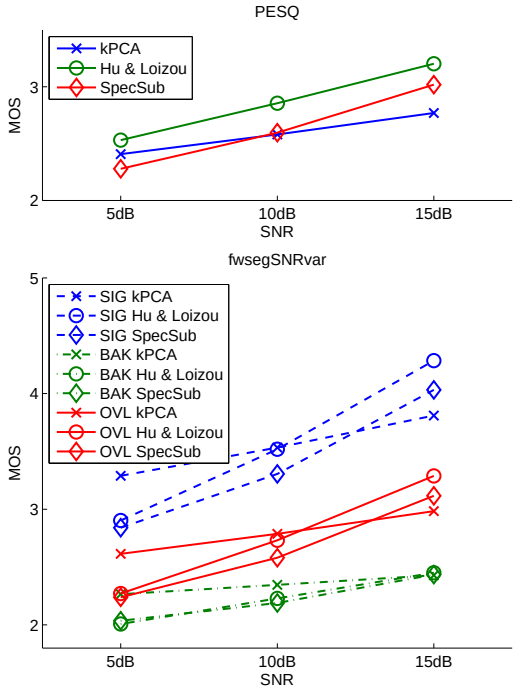

Figure 4: *Evaluation of kPCA, Hu and Loizou's algorithm, and spectral subtraction by the perceptual speech quality (PESQ) measure and a variant of the frequency-weighted SNR (fwsegSNRvar) [1]. Both return a Mean Opinion Score (MOS), where a higher value means a higher quality. The fwsegSNRvar separately evaluates the quality of the speech signal (SIG), the background intrusion (BAK), and the overall quality (OVL).*

sures: the perceptual evaluation of speech quality (PESQ) measure, the log-likelihood ratio (LLR), frequency-weighted segmental SNR (fwsegSNR), the Itakura-Saito (IS) distance, and a variant of the frequency-weighted segmental SNR (fwsegSNRvar) proposed in [1]. The PESQ returns values on the Mean Opinion Score (MOS) scale. The fwsegSNRvar returns three MOS values, where one only evaluates the signal quality (SIG), one the background intrusion (BAK) and one the overall quality (OVL). This evaluation is derived from the ITU-T [12] recommendation for subjective listening tests. By judging the speech and noise quality independently, better statements about the effects of the enhancement algorithm can be made. In [13], Hu and Loizou investigated the correlation between several objective quality measures and the results of subjective listening tests. In their experiments, PESQ, LLR and fwsegSNR showed highest correlations with the results of listening tests. The results for fwsegSNR, LLR and IS are shown in Fig. 3, the results for PESQ and fwsegSNRvar in Fig. 4.

The results for fwsegSNR and LLR are similar for all three approaches. The IS is largest for spectral subtraction and smallest for kPCA. In terms of PESQ Hu and Loizou's approach is best. The performance of kPCA measured by the fwsegSNRvar is better than or equal to the performance of the other approaches for 5 dB and 10 dB SNR. For 15 dB it is weaker, however this cannot be confirmed by subjective evaluation.

Objective evaluation measures model the outcome of a listening test not perfectly, as they do not fully correlate. Therefore, we plan to perform subjective listening tests according to the ITU-T P.835 recommendation in the future.

5. Conclusion and Future Work

We showed that kernel PCA can be applied to the complex time-frequency representation of speech utterances to perform unsupervised de-noising. Complex data can easily be handled by kPCA using a Gaussian kernel. As we do work directly in the complex domain, we do not need to rely on the noisy phase for back transformation into time domain.

The method works well for additive white Gaussian noise at SNR levels of 15, 10 and 5 dB. For a proper choice of the variance of the Gaussian kernel only minor artifacts are perceivable and no musical noise occurs.

In terms of objective quality measures we achieve a similar performance like linear PCA and spectral subtraction for 10 and 5 dB SNR. However, objective evaluation measures cannot fully predict the outcome of subjective listening tests. Therefore, we plan to evaluate the algorithm by an MOS test. In addition, we want to perform experiments on other speech databases including different noise types like, e.g., babble noise.

6. References

[1] P. C. Loizou, *Speech Enhancement: Theory and Practice*. CRC, 2007.

[2] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," *Advances in Neural Information Processing Systems 11*, pp. 536–542, 1999.

[3] K. I. Kim, M. O. Franz, and B. Schölkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1351–1366, 2005.

[4] T. Takiguchi and Y. Ariki, "PCA-based speech enhancement for distorted speech recognition." *Journal of Multimedia*, vol. 2, no. 5, pp. 13–18, 2007.

[5] Y. Hu and P. C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *IEEE Transactions on Speech and Audio Processing*, vol. 11, pp. 334–341, 2003.

[6] M. Berouti, M. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 208–211, 1979.

[7] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1000–1017, 1999.

[8] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Max Planck Institute for Biological Cybernetics, Tech. Rep., 1996.

[9] J. T. Kwok and I. W. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, pp. 408–415, 2004.

[10] B. King and L. Atlas, "Single-channel source separation using simplified-training complex matrix factorization," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4206 –4209, 2010.

[11] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 2, pp. 236 – 243, 1984.

[12] *Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm*, ITU-T Rec. P.835, 2003.

[13] Y. Hu and P. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229 –238, 2008.