Convolutional PCA for Multiple Time Series

Xi-Lin Li

Abstract—**We study a fundamental generalization of principal component analysis (PCA) that looks for a small set of common time series, i.e., principal components (PCs), whose filtered versions can explain most of the variances of multiple observed time series. This problem boils down to PCA in the frequency domain, in principle. But, frequency domain processing suffers from aliasing and brings inconveniences when handling certain time domain properties like nonstationarity and sparsity. We propose novel time and z-domain costs for such PCA, and study its properties in detail with setting of either finite or diverging filter lengths. We further discuss its implementations in batch and online processing forms, and present numerical results for empirical performance study. Convolution is used to extract these PCs, thus the name convolutional PCA.**

Index Terms—**Principal component analysis (PCA), time series, second order statistics, convolution, signal detection.**

I. Introduction

Principal component analysis (PCA) is one of the most widely used exploratory data analysis tools [1], [2]. It can be generalized to data of different formats and decompositions under diverse constraints in many ways [1]. Here, we focus on its application to time series. Let $\boldsymbol{x}(t)$, $t = \ldots, -1, 0, 1, \ldots$, be a series of real or complex valued $M \times 1$ vector with zero mean. In the simplest case, only spatial dependence is exploited, and t merely serves as a sample index. The first principal component (PC) can be extracted as $y(t) = \boldsymbol{u}\boldsymbol{v}^H\boldsymbol{x}(t)$ to keep as much variance as possible by

$$\min\, E[\|\boldsymbol{x}(t) - \boldsymbol{u}\boldsymbol{v}^H\boldsymbol{x}(t)\|^2], \text{ or, } \min \sum_{\forall t} \|\boldsymbol{x}(t) - \boldsymbol{u}\boldsymbol{v}^H\boldsymbol{x}(t)\|^2 \tag{1}$$

with respect to the two $M \times 1$ vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ [1], where superscript H denotes Hermitian transpose, the expectation is used for second-order stationary series, and sum on all valid t for series with finite length, but not necessarily stationary. The two costs make no difference in practice. We prefer the use of expectation to simplify our notations. It is shown that the optimal $\boldsymbol{u}$ and $\boldsymbol{v}$ are the same eigenvector of auto-correlation matrix $E[\boldsymbol{x}(t)\boldsymbol{x}^H(t)]$ associated with the largest eigenvalue [1]. This is the familiar eigendecomposition method for PCA.

However, many real world data, e.g., biomedical signals, audio recordings, seismic waves, records in many disciplines like atmospheric science and econometrics, etc., are sequential in nature and possess strong statistical dependence among adjacent samples. There are several possible ways to extend PCA to time series, to name a few, singular spectrum analysis (SSA) and empirical orthogonal function (EOF) analysis [1], PCA in the frequency domain or spectral PCA [3], [6], common signal detection [4], functional dynamic PCA where t is continuous [5], and time series PCA (TS-PCA) [7]. Among them, SSA, EOF and their multivariate extensions are PCA based on the embedding of time series, TS-PCA [7] is more like a factor analysis method since it pursuits to separate the observations into groups of spatiotemporally uncorrelated sources, and all the other extensions try to keep more variances from the observations with the same number of PCs as an ordinary PCA through exploiting the temporal dependence among adjacent samples. Specifically, we assume that most of the variances of observations can be explained by the filtered versions of N PCs, where $1 < N < M$. Similar to (1), the first PC can be extracted as $y(t) = \boldsymbol{u}(z)\boldsymbol{v}^H(z)\boldsymbol{x}(t)$ by

$$\min\, E[\|\boldsymbol{x}(t) - \boldsymbol{u}(z)\boldsymbol{v}^H(z)\boldsymbol{x}(t)\|^2] \tag{2}$$

where $\boldsymbol{u}(z)$ and $\boldsymbol{v}(z)$ are two filters [3]. When both $\boldsymbol{u}(z)$ and $\boldsymbol{v}(z)$ are filters with infinite length, minimizing (2) boils down to PCA in the frequency domain [3], also known as spectral PCA [6] or dynamic PCA [5]. But, frequency domain processing suffers from aliasing, and brings inconveniences when handling time domain signal properties like nonstationarity and sparsity [8], [9], [10]. It is not clear on how to perform the PCA defined in (2) in the time domain using filters with finite lengths. To our best knowledge, all such PCAs are done in the frequency domain [1], [3], [4], [5], [6], [8].

The main contribution here is to develop a novel time domain PCA for series with spatiotemporal dependence. Enlightened by the projection approximation subspace tracking (PAST) method [11], we simply force $\boldsymbol{u}(z) = \boldsymbol{v}(z)$ in (2) to define our PCA cost. We show that this cost also leads to PCA in the frequency domain with diverging filter length, and yet maintains an elegant relationship of conservation of variances with finite filter length, i.e., the power of observations equals the sum of that of PCs and residual errors. Implementation details and numerical results are provided as well. We name our method convolutional PCA to distinguish it from PCAs done in the frequency domain.

II. Convolutional PCA

A. Preliminary: Calculus in the z-Domain

The z-transform of a series $\{\ldots, \boldsymbol{A}_{-1}, \boldsymbol{A}_0, \boldsymbol{A}_1, \ldots\}$ is denoted by $\boldsymbol{A}(z) = \sum_{\forall i} \boldsymbol{A}_i z^{-i}$, where all $\boldsymbol{A}_i$ are matrices of the same size. Trace of $\boldsymbol{A}(z)$ is defined as

$$\operatorname{tr} \boldsymbol{A}(z) = \frac{1}{2\pi j} \operatorname{tr} \oint_{|z|=1} \frac{\boldsymbol{A}(z)}{z} dz = \operatorname{tr} \boldsymbol{A}_0 \tag{3}$$

where $\oint$ is counter-clockwise here, and $j = \sqrt{-1}$. It is clear that $\operatorname{tr} \boldsymbol{A}(z)\boldsymbol{B}(z) = \operatorname{tr} \boldsymbol{B}(z)\boldsymbol{A}(z)$. Let $f[\boldsymbol{A}(z)]$ be a real valued

The Author is with GMEMS Technologies, Inc., 366 Fairview Way, Milpitas, CA 95035 (e-mail: lixilinx@gmail.com).

function of $\boldsymbol{A}(z)$. The derivative, or gradient ascent direction, of $f[\boldsymbol{A}(z)]$ with respect to $\boldsymbol{A}(z)$ is defined as

$$\sum_{\forall i}\left(\frac{\partial f}{\partial \mathrm{Re}\boldsymbol{A}_i}+j\frac{\partial f}{\partial \mathrm{Im}\boldsymbol{A}_i}\right)z^{-i}=\sum_{\forall i}\frac{\partial f}{\partial \boldsymbol{A}_i^*}z^{-i}=\frac{\partial f}{\partial \boldsymbol{A}^*(z)} \tag{4}$$

where superscript $*$ denotes conjugate, and Re and Im take the real and imaginary parts of $\boldsymbol{A}_i$, respectively. When series $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ share the same set of valid indices i, we have

$$\frac{\partial \mathrm{tr}\,\boldsymbol{A}^H(z)\boldsymbol{B}(z)}{\partial \boldsymbol{A}^*(z)}=\sum_{\forall i}\boldsymbol{B}_i z^{-i}=\boldsymbol{B}(z) \tag{5}$$

where $\boldsymbol{A}^H(z)=[\boldsymbol{A}(z)]^H=\sum_{\forall i}\boldsymbol{A}_i^H z^i$. Relationship (5) is repetitively used in our derivations.

B. The PCA Cost Functions

Without loss of generality, we pursuit a filter $\boldsymbol{W}(z)$ to simultaneously extract N PCs from $\boldsymbol{x}(t)$ as $\boldsymbol{y}(t)=\boldsymbol{W}^H(z)\boldsymbol{x}(t)$, where $1\le N<M$, $\boldsymbol{W}(z)=\sum_{\forall i}\boldsymbol{W}_i z^{-i}$, $\boldsymbol{W}^H(z)=\sum_{\forall i}\boldsymbol{W}_i^H z^i$, and each $\boldsymbol{W}_i$ is an $M\times N$ matrix. The reconstructed series and reconstruction errors are defined as $\hat{\boldsymbol{x}}(t)=\boldsymbol{W}(z)\boldsymbol{y}(t)$ and $\boldsymbol{e}(t)=\boldsymbol{x}(t)-\hat{\boldsymbol{x}}(t)$, respectively. Then, the optimal $\boldsymbol{W}(z)$ can be obtained by minimizing the following time domain cost

$$c(\boldsymbol{W}(z))=E[\|\boldsymbol{e}(t)\|^2] \tag{6}$$

To facilitate the analysis, we prefer to rewrite $c(\boldsymbol{W}(z))$ in the z-domain as

$$c(\boldsymbol{W}(z))=\mathrm{tr}\,E[\boldsymbol{e}(t)\boldsymbol{e}^H(t)]=\mathrm{tr}\,\boldsymbol{R}_e(z) \tag{7}$$

where $\boldsymbol{R}_e(z)=\sum_{\tau=-\infty}^{\infty}\boldsymbol{E}[\boldsymbol{e}(t)\boldsymbol{e}^H(t-\tau)]z^{-\tau}$. Note that $\boldsymbol{R}_e(z)$ is Hermitian in the sense that $\boldsymbol{R}_e(z)=\boldsymbol{R}_e^H(z)$.

Let us further rewrite $c(\boldsymbol{W}(z))$ as a direct function of $\boldsymbol{W}(z)$ and $\boldsymbol{x}(t)$. Note that $\boldsymbol{e}(t)$ is explicitly related to $\boldsymbol{x}(t)$ by

$$\boldsymbol{e}(t)=\boldsymbol{x}(t)-\boldsymbol{W}(z)\boldsymbol{y}(t)=[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]\boldsymbol{x}(t)$$

Thus, $\boldsymbol{R}_e(z)$ is related to $\boldsymbol{R}_x(z)$, the z-transform of the series of correlation matrices of $\boldsymbol{x}(t)$, by

$$\boldsymbol{R}_e(z)=[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]\boldsymbol{R}_x(z)[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]$$

Now, this gives us the explicit form of the z-domain cost

$$\begin{aligned}c(\boldsymbol{W}(z)) &= \mathrm{tr}\,\boldsymbol{R}_e(z)\\ &= \mathrm{tr}\,[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]\boldsymbol{R}_x(z)[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]\\ &= \mathrm{tr}\,[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]^2\boldsymbol{R}_x(z)\end{aligned} \tag{8}$$

Note that this cost is insensitive to the ‘rotations’ of the columns of $\boldsymbol{W}(z)$ as $c(\boldsymbol{W}(z))=c(\boldsymbol{W}(z)\boldsymbol{Q}(z))$ for any $N\times N$ paraunitary filter $\boldsymbol{Q}(z)$ satisfying $\boldsymbol{Q}(z)\boldsymbol{Q}^H(z)=\boldsymbol{I}$.

C. Properties of Convolutional PCA with Infinite Filter Length

In this subsection, we assume that $\boldsymbol{W}(z)$ is doubly infinite. The gradient of $c(\boldsymbol{W}(z))$ with respect to $\boldsymbol{W}(z)$ is given by

$$\begin{aligned}\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)} &= -[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]\boldsymbol{R}_x(z)\boldsymbol{W}(z)\\ &\quad -\boldsymbol{R}_x(z)[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]\boldsymbol{W}(z)\end{aligned} \tag{9}$$

A stationary point of $c(\boldsymbol{W}(z))$ is a solution for $\boldsymbol{W}(z)$ such that $\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}=\boldsymbol{0}$. We say that a stationary point $\boldsymbol{W}(z)$ is trivial if there exists at least one $0\le\omega<2\pi$ such that matrix $\boldsymbol{W}^H(e^{j\omega})\boldsymbol{W}(e^{j\omega})$ is singular. For example, it is straightforward to verify that $\boldsymbol{W}(z)=\boldsymbol{0}$ is a trivial stationary point. We have the following property for those nontrivial stationary points of $c(\boldsymbol{W}(z))$.

Property 1: Assume that $\boldsymbol{R}_x(e^{j\omega})$ is positive definite for all $0\le\omega<2\pi$. Then, we have $\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{I}$ at a nontrivial stationary point.

Proof: Since $\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}=\boldsymbol{0}$ at a stationary point, we have

$$\boldsymbol{W}^H(z)\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}=\boldsymbol{0} \tag{10}$$

Then, substituting (9) into (10) leads to

$$[\boldsymbol{I}-\boldsymbol{W}^H(z)\boldsymbol{W}(z)]\boldsymbol{R}_y(z)+\boldsymbol{R}_y(z)[\boldsymbol{I}-\boldsymbol{W}^H(z)\boldsymbol{W}(z)]=\boldsymbol{0} \tag{11}$$

where $\boldsymbol{R}_y(z)=\boldsymbol{W}^H(z)\boldsymbol{R}_x(z)\boldsymbol{W}(z)$ is the z-transform of the series of correlation matrices of $\boldsymbol{y}(t)$. Let

$$\boldsymbol{I}-\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{U}(z)\boldsymbol{D}(z)\boldsymbol{U}^H(z)$$

where $\boldsymbol{U}^H(z)\boldsymbol{U}(z)=\boldsymbol{I}$ and $\boldsymbol{D}(z)$ is diagonal. Then, (11) can be rewritten as

$$\boldsymbol{D}(z)\boldsymbol{U}^H(z)\boldsymbol{R}_y(z)\boldsymbol{U}(z)+\boldsymbol{U}^H(z)\boldsymbol{R}_y(z)\boldsymbol{U}(z)\boldsymbol{D}(z)=\boldsymbol{0} \tag{12}$$

Specifically, diagonals of the left size of (12) are zeros as well. Note that $\boldsymbol{U}^H(z)\boldsymbol{R}_y(z)\boldsymbol{U}(z)$ is positive definite for any $z=e^{j\omega}$ since $\boldsymbol{R}_x(e^{j\omega})$ is positive definite and $\boldsymbol{W}(e^{j\omega})$ has full column rank. Thus, its diagonals are positive for any $z=e^{j\omega}$. Hence, we must have $\boldsymbol{D}(e^{j\omega})=\boldsymbol{0}$, which implies $\boldsymbol{I}-\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{0}$. This finishes the proof. ■

When $\boldsymbol{R}_x(e^{j\omega})$ is singular for certain $0\le\omega<2\pi$, we still can enforce $\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{I}$ at a nontrivial stationary point without increasing the cost. Thus, we always assume that $\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{I}$ for any nontrivial stationary point.

Property 2: At a nontrivial stationary point, the reconstruction errors are orthogonal to the PCs and reconstructed series in the sense that $E[\boldsymbol{e}(t)\boldsymbol{y}^H(t-\tau)]=\boldsymbol{0}$ and $E[\boldsymbol{e}(t)\hat{\boldsymbol{x}}^H(t-\tau)]=\boldsymbol{0}$ for any integer τ.

Proof: We define $\boldsymbol{R}_{ey}(z)=\sum_{\tau=-\infty}^{\infty}E[\boldsymbol{e}(t)\boldsymbol{y}^H(t-\tau)]z^{-\tau}$ and $\boldsymbol{R}_{e\hat{x}}(z)=\sum_{\tau=-\infty}^{\infty}E[\boldsymbol{e}(t)\hat{\boldsymbol{x}}^H(t-\tau)]z^{-\tau}$. With relationship $\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{I}$, we have

$$\begin{aligned}\boldsymbol{R}_{ey}(z) &= \boldsymbol{R}_x(z)\boldsymbol{W}(z)-\boldsymbol{W}(z)\boldsymbol{R}_y(z)\\ \boldsymbol{R}_{e\hat{x}}(z) &= \boldsymbol{R}_{ey}(z)\boldsymbol{W}^H(z)\end{aligned}$$

At the same time, the gradient can be simplified to

$$\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}=\boldsymbol{W}(z)\boldsymbol{R}_y(z)-\boldsymbol{R}_x(z)\boldsymbol{W}(z)=-\boldsymbol{R}_{ey}(z) \tag{13}$$

with relationship $\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{I}$. Thus, we have $\boldsymbol{R}_{ey}(z)=\boldsymbol{0}$ and $\boldsymbol{R}_{e\hat{x}}(z)=\boldsymbol{R}_{ey}(z)\boldsymbol{W}^H(z)=\boldsymbol{0}$ at a nontrivial stationary point. ■

With (13), let us rewrite $\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}=\boldsymbol{0}$ as

$$\boldsymbol{R}_x(z)\boldsymbol{W}(z)=\boldsymbol{W}(z)\boldsymbol{R}_y(z) \tag{14}$$

Relationship (14) is comparable to the eigendecomposition method for an ordinary PCA. The following property further elaborates this point.

Property 3: The nontrivial stationary points of cost $c(\boldsymbol{W}(z))$ can be found by optimization problem

$$\max \operatorname{tr} \boldsymbol{W}^H(z)\boldsymbol{R}_x(z)\boldsymbol{W}(z), \quad \text{s.t.}\, \boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{I} \tag{15}$$

Proof: We only need to show that maximizing this objective is equivalent to minimize $c(\boldsymbol{W}(z))$ under constraints $\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{I}$. Expanding the $c(\boldsymbol{W}(z))$ in (8) leads to

$$\begin{aligned} c(\boldsymbol{W}(z)) &= \operatorname{tr}[\boldsymbol{I}-2\boldsymbol{W}(z)\boldsymbol{W}^H(z)+ \\ &\qquad \boldsymbol{W}(z)\boldsymbol{W}^H(z)\boldsymbol{W}(z)\boldsymbol{W}^H(z)]\boldsymbol{R}_x(z) \\ &= \operatorname{tr}[\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z)]\boldsymbol{R}_x(z) \\ &= \operatorname{tr}\boldsymbol{R}_x(z) - \operatorname{tr}\boldsymbol{W}^H(z)\boldsymbol{R}_x(z)\boldsymbol{W}(z) \end{aligned}$$

Thus, minimizing $c(\boldsymbol{W}(z))$ is equivalent to maximizing $\operatorname{tr}\boldsymbol{W}^H(z)\boldsymbol{R}_x(z)\boldsymbol{W}(z)$ since $\operatorname{tr}\boldsymbol{R}_x(z)$ is a constant with respect to $\boldsymbol{W}(z)$. ■

We can restate (15) in the frequency domain as

$$\begin{aligned} \max \quad & \int_{\omega=0}^{2\pi} \operatorname{tr}\boldsymbol{W}^H(e^{j\omega})\boldsymbol{R}_x(e^{j\omega})\boldsymbol{W}(e^{j\omega})d\omega, \\ \text{s.t.} \quad & \boldsymbol{W}^H(e^{j\omega})\boldsymbol{W}(e^{j\omega})=\boldsymbol{I}, \quad \text{for all } 0\le\omega<2\pi \end{aligned}$$

This is the familiar PCA in the frequency domain.

D. Properties of Convolutional PCA with Finite Filter Length

In practice, we prefer filters with finite length. Needless to say, properties 1, 2 and 3 still approximately hold when the filter length is large enough. Here, we assume that the filter is causal and has order L, i.e., $\boldsymbol{W}(z)=\sum_{i=0}^{L}\boldsymbol{W}_i z^{-i}$. The gradient still has the form given in (9). But, the exponents i of z^{-i} can only be integers in range $[-L,0]$. We denote this truncated gradient as $\left[\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}\right]_{-L,0}$. Now, at a stationary point of $c(\boldsymbol{W}(z))$, we only have $\left[\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}\right]_{-L,0}=\boldsymbol{0}$, but not $\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}=\boldsymbol{0}$, nor $\boldsymbol{W}^H(z)\boldsymbol{W}(z)=\boldsymbol{I}$.

Property 4: At a stationary point, the reconstructed series and reconstruction errors are weakly orthogonal in the sense that $E[\boldsymbol{e}^H(t)\hat{\boldsymbol{x}}(t)]=0$, or equivalently, $E[\|\boldsymbol{x}(t)\|^2]=E[\|\boldsymbol{e}(t)\|^2]+E[\|\hat{\boldsymbol{x}}(t)\|^2]$ since $\boldsymbol{x}(t)=\hat{\boldsymbol{x}}(t)+\boldsymbol{e}(t)$.

Proof: On one hand, we have

$$(E[\boldsymbol{e}^H(t)\hat{\boldsymbol{x}}(t)])^* = \operatorname{tr}E[\boldsymbol{e}(t)\hat{\boldsymbol{x}}^H(t)] = \operatorname{tr}\boldsymbol{R}_{e\hat{x}}(z)$$

which is further shown to be

$$\begin{aligned} \operatorname{tr}\boldsymbol{R}_{e\hat{x}}(z) &= \operatorname{tr}(\boldsymbol{I}-\boldsymbol{W}(z)\boldsymbol{W}^H(z))\boldsymbol{R}_x(z)\boldsymbol{W}(z)\boldsymbol{W}^H(z) \\ &= \operatorname{tr}\boldsymbol{R}_y(z) - \operatorname{tr}\boldsymbol{W}(z)\boldsymbol{R}_y(z)\boldsymbol{W}^H(z) \end{aligned} \tag{16}$$

On the other hand, we have

$$\operatorname{tr}\boldsymbol{W}^H(z)\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)} = \operatorname{tr}\boldsymbol{W}^H(z)\left[\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}\right]_{-L,0} = 0$$

at a stationary point. Expanding $0=\operatorname{tr}\boldsymbol{W}^H(z)\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}$ leads to

$$\begin{aligned} 0 &= \operatorname{tr}\boldsymbol{W}^H(z)\boldsymbol{W}(z)\boldsymbol{R}_y(z) + \operatorname{tr}\boldsymbol{R}_y(z)\boldsymbol{W}^H(z)\boldsymbol{W}(z) \\ &\quad -2\operatorname{tr}\boldsymbol{R}_y(z) \\ &= 2\operatorname{tr}\boldsymbol{W}(z)\boldsymbol{R}_y(z)\boldsymbol{W}^H(z) - 2\operatorname{tr}\boldsymbol{R}_y(z) \end{aligned} \tag{17}$$

Now, combining (16) and (17) leads to $\operatorname{tr}\boldsymbol{R}_{e\hat{x}}(z)=0$. ■

Property 5: At a stationary point, PCs and the reconstructed series have the same power, i.e., $E[\|\hat{\boldsymbol{x}}(t)\|^2]=E[\|\boldsymbol{y}(t)\|^2]$.

Proof: Note that

$$\begin{aligned} E[\|\hat{\boldsymbol{x}}(t)\|^2] &= \operatorname{tr}\boldsymbol{W}(z)\boldsymbol{W}^H(z)\boldsymbol{R}_x(z)\boldsymbol{W}(z)\boldsymbol{W}^H(z) \\ &= \operatorname{tr}\boldsymbol{W}(z)\boldsymbol{R}_y(z)\boldsymbol{W}^H(z) \end{aligned}$$

Then, (17) gives the proof since $E[\|\boldsymbol{y}(t)\|^2]=\operatorname{tr}\boldsymbol{R}_y(z)$. ■

Properties 4 and 5 together show that sum of the powers of PCs and residual errors equals that of the observations, i.e.,

$$E[\|\boldsymbol{x}(t)\|^2] = E[\|\boldsymbol{e}(t)\|^2] + E[\|\boldsymbol{y}(t)\|^2] \tag{18}$$

Eq. (18) justifies the use of convolutional PCA with any finite filter length. Yet, it is not always feasible to enforce $\boldsymbol{y}(t)$ to be a vector process of N spatiotemporally uncorrelated series using filters with finite lengths. Instead, one can apply a nested convolutional PCA on $\boldsymbol{y}(t)$ to decompose it as $\boldsymbol{y}(t)=\boldsymbol{e}'(t)+\boldsymbol{W}'(z)\boldsymbol{y}'(t)$, where $\boldsymbol{W}'(z)$ is an $N\times K$ filter, and $0<K<N<M$. With sufficiently large filter length, $\mathbf{e}'(t)$ and $\boldsymbol{y}'(t)$ are approximately spatiotemporally decorrelated due to property 2. The relationship of conservation of variances,

$$\begin{aligned} E[\|\boldsymbol{x}(t)\|^2] &= E[\|\boldsymbol{e}(t)\|^2] + E[\|\boldsymbol{y}(t)\|^2] \\ &= E[\|\boldsymbol{e}(t)\|^2] + E[\|\boldsymbol{e}'(t)\|^2] + E[\|\boldsymbol{y}'(t)\|^2] \end{aligned}$$

always holds as long as $\boldsymbol{W}(z)$ and $\boldsymbol{W}'(z)$ are stationary points.

III. Implementations Details

We assume that the filter is causal and has order L. Note that different from an ordinary PCA, generally, there is no guarantee of global convergence for convolutional PCA with finite filter length. Still, our experiences suggests that initial guess $\boldsymbol{C}^H z^{-\lfloor L/2 \rfloor}$ works well most of the time, where $\boldsymbol{C}$ are the first N rows of an $M\times M$ discrete cosine transform (DCT) matrix, and $\lfloor\cdot\rfloor$ denotes rounding to the nearest integer. The PCA cost seems have multiple saddle points. Our experiences suggests that momentum [12] could improve the convergence.

1) Batch Processing: We use gradient descent with momentum to optimize $\boldsymbol{W}(z)$ by repeating updates

$$\begin{aligned} \boldsymbol{M}(z) &\leftarrow 0.9\boldsymbol{M}(z) + 0.1\left[\frac{\partial c(\boldsymbol{W}(z))}{\partial \boldsymbol{W}^*(z)}\right]_{-L,0} \\ \boldsymbol{W}(z) &\leftarrow \boldsymbol{W}(z) - \mu\boldsymbol{M}(z) \end{aligned}$$

where $\boldsymbol{M}(z)$ is the smoothed truncated gradient, and $\mu>0$ is the step size. Note that it is sufficient to truncate $\boldsymbol{R}_x(z)$ to $[\boldsymbol{R}_x(z)]_{-2L,2L}$ since terms with absolute exponents of z larger than $2L$ do not contribute to $\boldsymbol{R}_e(0)$.

2) Online Processing: The instantaneous gradient of $\|\boldsymbol{e}(t)\|^2$ with respect to $\boldsymbol{W}_i$ is given by

$$\boldsymbol{G}_i(t) = \frac{\partial\|\boldsymbol{e}(t)\|^2}{\partial \boldsymbol{W}_i^*} \approx -\boldsymbol{e}(t)\boldsymbol{y}^H(t-i) - \sum_{j=0}^{L}\boldsymbol{x}(t+i-j)\boldsymbol{e}^H(t)\boldsymbol{W}_j$$

The above gradient is approximate since $\boldsymbol{y}(t-i)$ with $i>0$ are stale. This is acceptable when the updating step size is small enough. In our implementations, the filter coefficients at time t are updated using a normalized stochastic gradient descent (SGD) method with momentum as below

$$\boldsymbol{M}_i \leftarrow 0.9\boldsymbol{M}_i + 0.1\boldsymbol{G}_i(t), \quad \sigma \leftarrow 0.9\sigma + 0.1\sum_{i=0}^{L}\|\boldsymbol{G}_i(t)\|^2$$

$$\boldsymbol{W}_i \leftarrow \boldsymbol{W}_i - \mu\boldsymbol{M}_i/\sqrt{\sigma+\varepsilon}, \quad 0 \le i \le L \qquad (19)$$

where offset $\varepsilon \ge 0$ is for reducing steady state adaptation errors.

3) Possible Extensions: Convolutional PCA can be extended in several ways. One example is to replace the ℓ^2 norm in cost (6) with ℓ^1 norm to make it robust to outliers as in ℓ^p-PCA [9]. Another example is to add penalty term $\eta\sum_{\forall i}|\boldsymbol{W}_i|$ to cost (6) or (7) as in sparse PCA [10], where $\eta > 0$ and $|\boldsymbol{W}_i|$ denotes the ℓ_1 norm of $\boldsymbol{W}_i$. Utilities of such extensions are not the focus here as they depend on the specific problems at hand.

IV. Experimental Results

Please refer to https://github.com/lixilinx/ConvPCA for the data and Matlab/Octave[1] code reproducing the results reported below, and more results not reported here due to limited space.

1) Time Difference of Arrival (TDOA) Estimation: The observations are two microphone recordings of a single speech source in a highly reverberant conference room. The true TDOA is about 12 samples at 16 KHz sampling rate. Thus, we expect that $\boldsymbol{W}(z)$ converges to solution $[\sqrt{0.5}z^{-12-d}; \sqrt{0.5}z^{-d}]$ up to an arbitrary delay uncertainty d. To encourage sparse solutions, we add penalty $\eta\sum_{\forall i}|\boldsymbol{W}_i|$ to our PCA cost. Fig. 1 shows the estimation results by generalized cross correlation (GCC) [13] and the batch version of convolutional PCA with $L = 32$ and $\eta = 0.2$. Convolutional PCA clearly outperforms the classic GCC baseline.

2) PCA for Electrocardiogram (ECG) Signal Analysis: In this experiment, we test different PCA methods on the ECG data measured from a pregnant woman and distributed by B. De Moor [14]. Eight signals are recorded with sampling rate 250 Hz and duration 10 second. The mixtures mainly consist of three components: strong and slow heart beating of mother ECG, weak and fast heart beating of fetus ECG, and a low frequency interference due to breathing. Dynamic ranges of different channels vary a lot. Thus, we normalize the variance of each channel to unit before applying PCA.

For convolutional PCA, we use the batch processing with $L = 10$. For PCA in the frequency domain, we use Welch's method with frame size $2L$ and cosine window for power spectral matrices estimation. Fig. 2 summarizes the results by different PCAs with different number of PCs. Convolutional PCA performs the best in the sense of preserving more percentages of variances and weak fetus ECG. Aliasing is inherent to frequency domain processing. Here, it significantly contaminates the extracted weak fetus ECG signal.

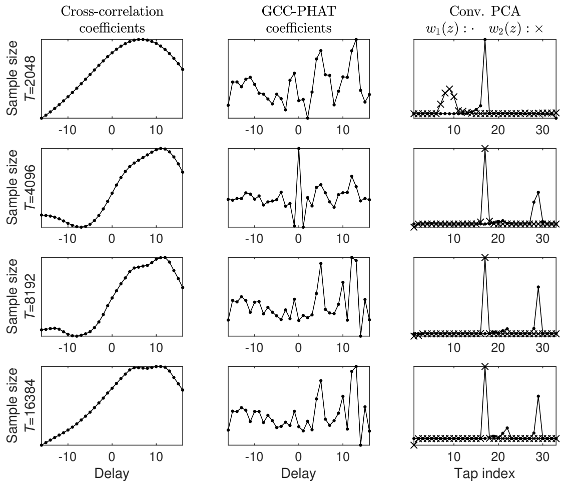

Fig. 1. TDOA estimations from GCC (location of the maximum peak) and convolutional PCA (gap between the locations of maximum peak of each filter) with different sample size T. PHAT denotes phase transform, a special case of GCC that only uses phase information for TDOA estimation. GCC shows either blunt peaks with cross correlation, or strong spurious peaks with PHAT. Convolutional PCA delivers sharper and more stable TDOA estimates with reasonably large sample size T.

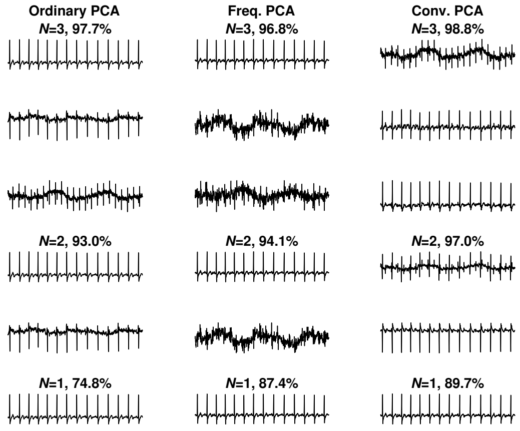

Fig. 2. Comparisons of different PCAs on ECG signal analysis. Decomposition results of an ordinary PCA and PCA in the frequency domain do not change with N. Still, their PCs with different N are plotted here side by side with those of convolutional PCA for easy comparisons. Percentage numbers of preserved variances by PCA in the frequency domain are just for reference purpose as aliasing introduces an error about 2% here. Only convolutional PCA can keep fetus ECG, the fast and weak beatings, with $N = 2$.

V. Conclusion

We have proposed a novel principal component analysis (PCA), convolutional PCA, for multiple time series, and studied its properties in details. Convolutional PCA is closely related to PCA in the frequency domain. But, it can be solved in the time or z-domain. This makes it completely free of aliasing, and ready to exploit time domain properties like nonstationarity and sparsity. Experimental results are reported for performance study as well.

[1] www.mathworks.com and www.gnu.org/software/octave/, respectively.

References

[1] I. T. Jolliffe, *Principal Component Analysis*. 2nd ed., Springer-Verlag New York, 2002.

[2] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philos Trans A Math Phys Eng Sci.*, Apr. 2016. Accessed on Jun., 2020, doi:10.1098/rsta.2015.0202, [Online].

[3] D. R. Brillinger, *Time Series: Data Analysis and Theory*. San Francisco: Holden-Day, 1981.

[4] D. S. Stoffer, “Detecting common signals in multiple time series using the spectral envelope,” *J. Amer. Statist. Assoc.*, vol. 94, no. 448, pp. 1341–1356, Dec. 1999.

[5] S. Hörmann, L. Kidziński, and M. Hallin, “Dynamic functional principal component,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 77, no. 2, pp. 319–348, Jun. 2015.

[6] N. F. Thornhill, S. L. Shah, B. Huang, and A. Vishnubhotla, “Spectral principal component analysis of dynamic process data,” *Control Engineering Practice*, vol. 10, no. 8, pp. 833-846, Aug. 2002.

[7] J. Chang, B. Guo, and Q. Yao, “Principal component analysis for second-order stationary vector time series,” *Annals of Statistics*, vol. 46, no. 5, pp. 2094–2124, 2018.

[8] H. Ombaoa and M. R. Ho, “Time-dependent frequency domain principal components analysis of multichannel non-stationary signals,” *Computational Statistics & Data Analysis*, vol. 50, no. 9, pp. 2339–2360, May 2006.

[9] N. Kwak, “Principal component analysis by L_p-norm maximization,” *IEEE Trans. Cybernetics*, vol. 44, no. 5, pp. 594–609, May 2014.

[10] H. Zou, T. Hastie, and R. Tibshirani, “Sparse principal components,” *J. Comput. Graph. Stat.*, vol. 15, pp. 262–264, 2006.

[11] B. Yang, “Projection approximation subspace tracking,” *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.

[12] G. Gabriel, “Why momentum really works,” *Distill*, 2017. Accessed on Jun., 2020, doi:10.23915/distill.00006. [Online]. Available: http://distill.pub/2017/momentum

[13] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 24, pp. 320–327, Aug. 1976.

[14] B. De Moor, *DaISy: database for the identification of systems*. [Online]. Available: http://homes.esat.kuleuven.be/ smc/daisy/