Neurocomputing 000 (2017) 1–9

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A fast and effective principal singular subspace tracking algorithm

Xiaowei Feng[a,b], Xiangyu Kong[a,*], Donghui Xu[a], Jianqiang Qin[a]

[a] *Xi'an Research Institute of High Technology, Xi'an 710025, China*
[b] *Mailbox 158-6-50, Baoji 721012, China*

ARTICLE INFO

Article history:
Received 19 April 2016
Revised 30 December 2016
Accepted 2 June 2017
Available online xxx

Communicated by Prof. Liu Guangcan

Keywords:
Principal singular value
Principal singular subspace
Singular value decomposition
Cross-correlation neural network

ABSTRACT

In this paper, we propose a fast and effective neural network algorithm to perform singular value decomposition (SVD) of a cross-covariance matrix between two high-dimensional data streams. Firstly, we derive a dynamical system from a newly proposed information criterion. This system exhibits a single stable stationary point if and only if the weight matrices of the left and right neural networks span the left and right principal singular subspace of a cross-covariance matrix, respectively, and the other stationary points are (unstable) saddle points. Then, a principal singular subspace (PSS) tracking algorithm is obtained from the dynamical system. Moreover, convergence analysis shows that the proposed algorithm converges to a stationary point that relates to the principal singular values. Thus, compared with traditional algorithms who can only track the PSS, the proposed algorithm can not only track the PSS but also estimate all of the corresponding principal singular values based on the extracted subspace. Finally, numerical simulations and practical application are carried to further demonstrate the efficiency of the proposed algorithm.

1. Introduction

Principal component analysis (PCA) is a well-known statistical criterion that has been used for many years, typically in feature extraction and data compression applications [1–3]. This criterion turned out to be closely related to the learning rule proposed half a century ago by Anderson [4] for the self-organization of neural network. In 1982, Oja [5] showed that a single linear unit trained with a normalized version of the Hebbian rule asymptotically extracts the principal component of the input sequence. In the wake of the important initial contributions by Oja, a plethora of neural network learning rules for principal component analysis (PCA) have been developed (see e.g. [6]).

Besides PCA method, many signal-processing tasks can also efficiently be achieved by singular value decomposition (SVD) of a nonsquared matrix [7], such as pattern recognition [8] and face recognition [9]. In the signal processing field of SVD, the first problem is how to get the SVD of a priori known nonsquared matrix. To solve this problem, early methods which can get the exact or approximate SVD of a nonsquared matrix have been proposed based on matrix algebra [10–13]. These batch-processing methods are robust, however, they are inefficient and cannot be used for real-time processing. The second problem is how to extract features of a cross-covariance matrix between two vector sequences on-line. These algorithms can avoid the computation of the cross-covariance matrix and instead directly work on the data samples. This is advantageous especially for high-dimensional data where the cross-covariance matrices would consume a large amount of memory and their update would be computationally expensive [14]. For on-line SVD, some neural network algorithms such as [15–19] were proposed based on Hebbian rule. These neural networks can also obtain SVD of a rectangular matrix, if and only if their weight matrix is taken as $\boldsymbol{A}^T\boldsymbol{A}$ or $\boldsymbol{A}\boldsymbol{A}^T$. However, if the data matrix is ill-conditioned, then the operation $\boldsymbol{A}^T\boldsymbol{A}$ or $\boldsymbol{A}\boldsymbol{A}^T$ usually is numerically unstable and should be avoided [20]. The third problem is how to improve the stability and availability of algorithms. To solve this problem, some gradient-based algorithm such as [21–25] were proposed. However, convergence of the gradient-based algorithms always depend on the appropriate selection of the learning rate, but it is difficult to be determined in advance because the learning rate are directly related to the underlying matrix. Moreover, these algorithms can only extract the principal cross correlation feature according to the largest singular value. In fact, multiple principal singular vectors or PSS are important in some situations. Thus, the fourth problem is how to extract multiple principal singular vectors or track PSS of two vector sequences. Until now, subspace tracking methods have been widely used in many practical applications, such as spectral clustering [26], authentic samples recovering [27] and subspace segmentation [28]. In the respect of PSS tracking problem, some sequential or parallel

* Corresponding author.
E-mail addresses: xiangyukong@126.com, xiangyukong01@163.com (X. Kong).

http://dx.doi.org/10.1016/j.neucom.2017.06.006
0925-2312/

algorithms such as [29–33] were proposed. It is found that most of the aforementioned algorithms can only extract principal singular vector(s) or track PSS. However, the singular values which are also very useful in some engineering practice have been ignored. To address this issue, Hassan [30] derived an interesting algorithm for PSS tracking in which all principal singular values can also be estimated. But as pointed out [14], the straight-forward formulation of a stochastic online algorithm for Hassans algorithm is infeasible due to the matrix inversion involved in the equations. In [14], an interesting algorithm, which is actually a coupled SVD algorithm, was proposed. In this coupled algorithm, the singular value is estimated alongside the singular vectors. In [34], another coupled SVD algorithm was also proposed. However, these two algorithms are only available for single component analysis, and multiple component analysis can only be achieved by sequential extraction method. The objective of this paper is to propose a fast and effective PSS tracking algorithm, which can also extract all of the principal singular values and does not involve matrix inversion computation.

In this paper, based on a novel information criterion, we first propose a novel dynamical system which exhibits a single stable stationary point attained if and only if the weight matrices of the left and right neural network span the left and right principal singular subspace of a cross-covariance matrix, respectively. Then, we obtain an effective online gradient-based algorithm for PSS tracking from the dynamical system. Moreover, all principal singular values of the PSS can be estimated in a parallel way based on the extracted subspace. Compared with traditional methods, the proposed algorithm can not only track the PSS but also estimate all of the corresponding principal singular values. Compared with Kaiser et al. [14] algorithm, the proposed algorithm can extract multiple principal singular triplets (singular value and singular vectors) in a parallel way. Compared with Hasan's [30] algorithm, the proposed algorithm does not involve matrix inversion computation. Experiment results show that the proposed algorithm performs better than existing algorithms in terms of convergence speed, orthogonality deviation and numerical stability.

2. Preliminary

Considering an m-dimensional sequence $\boldsymbol{x}(k)$ and an n-dimensional sequence $\boldsymbol{y}(k)$ with sampling number k large enough. If $\boldsymbol{x}(k)$ and $\boldsymbol{y}(k)$ are jointly stationary, then their cross-covariance matrix $\boldsymbol{A}=E[\boldsymbol{x}\boldsymbol{y}^T]$ can be estimated by

$$\boldsymbol{A}(k)=\frac{1}{k}\sum_{j=1}^{k}\boldsymbol{x}(j)\boldsymbol{y}^T(j)\in\mathbb{R}^{m\times n}. \tag{1}$$

If $\boldsymbol{x}(k)$ and $\boldsymbol{y}(k)$ are jointly nonstationary and even slowly time-varying, then their cross-covariance matrix can be estimated by

$$\boldsymbol{A}(k)=\sum_{j=1}^{k}\alpha^{k-j}\boldsymbol{x}(j)\boldsymbol{y}^T(j)\in\mathbb{R}^{m\times n}, \tag{2}$$

where $0<\alpha<1$ denotes the forgetting factor which makes the past data samples be less weighted than the recent ones.

The SVD of a real matrix $\boldsymbol{A}$ is given by Kaiser et al. [14]

$$\boldsymbol{A}=\bar{\boldsymbol{U}}\bar{\boldsymbol{S}}\bar{\boldsymbol{V}}^T+\bar{\boldsymbol{U}}_2\bar{\boldsymbol{S}}_2\bar{\boldsymbol{V}}_2^T, \tag{3}$$

where $\bar{\boldsymbol{U}}=[\bar{\boldsymbol{u}}_1,\dots,\bar{\boldsymbol{u}}_M]\in\mathbb{R}^{m\times M}$ and $\bar{\boldsymbol{V}}=[\bar{\boldsymbol{v}}_1\dots,\bar{\boldsymbol{v}}_M]\in\mathbb{R}^{n\times M}$ denotes the left and right PSS that containing the left and right principal singular vectors, respectively, and $\bar{\boldsymbol{S}}=\mathrm{diag}(\sigma_1\dots,\sigma_M)\in\mathbb{R}^{M\times M}$ denotes the matrix with the principal singular values on its diagonal of $\boldsymbol{A}$. Here we refer to these matrices as the principal portion of the SVD. Thus, $\hat{\boldsymbol{A}}=\bar{\boldsymbol{U}}\bar{\boldsymbol{S}}\bar{\boldsymbol{V}}^T$ is the best rank-M approximation (in the least-squares sense) of $\boldsymbol{A}$, where $M\le p=\min\{m,n\}$. Moreover, $(\bar{\boldsymbol{u}}_j,\bar{\boldsymbol{v}}_j,\sigma_j)$ is called the jth singular triplet of the cross-covariance matrix $\boldsymbol{A}$. Furthermore, $\bar{\boldsymbol{U}}_2=[\bar{\boldsymbol{u}}_{M+1},\dots,\bar{\boldsymbol{u}}_p]\in\mathbb{R}^{m\times(p-M)}$, $\bar{\boldsymbol{S}}_2=\mathrm{diag}(\sigma_{M+1},\dots,\sigma_p)\in\mathbb{R}^{(p-M)\times(p-M)}$ and $\bar{\boldsymbol{V}}_2=[\bar{\boldsymbol{v}}_{M+1},\dots,\bar{\boldsymbol{v}}_p]\in\mathbb{R}^{n\times(p-M)}$ correspond to the minor portion of the SVD. $\bar{\boldsymbol{U}}$, $\bar{\boldsymbol{V}}$, $\bar{\boldsymbol{U}}_2$, $\bar{\boldsymbol{V}}_2$ are orthogonal, i.e., $\bar{\boldsymbol{U}}^T\bar{\boldsymbol{U}}=\bar{\boldsymbol{V}}^T\bar{\boldsymbol{V}}=\boldsymbol{I}_M$ and $\bar{\boldsymbol{U}}_2^T\bar{\boldsymbol{U}}_2=\bar{\boldsymbol{V}}_2^T\bar{\boldsymbol{V}}_2=\boldsymbol{I}_{p-M}$. Moreover, we assume that the singular values are ordered and mutually different with respect to their absolute value such that $|\sigma_1|>\dots>|\sigma_M|>|\sigma_{M+1}|>\dots>|\sigma_p|$. In the following, all considerations (e.g., concerning fixed points) depend on the principal portion of the SVD in the domain $\{(\bar{\boldsymbol{U}},\bar{\boldsymbol{V}})|\bar{\boldsymbol{U}}^T\boldsymbol{A}\bar{\boldsymbol{V}}>0\}$ only, thus $\sigma_j>0\ \forall j\le p$.

3. Novel information criterion and algorithm

For a cross-covariance matrix $\boldsymbol{A}=E[\boldsymbol{x}\boldsymbol{y}^T]\in\mathbb{R}^{m\times n}$, and given $\boldsymbol{U}\in\mathbb{R}^{m\times r}$ and $\boldsymbol{V}\in\mathbb{R}^{n\times r}$ in the domain $\{(\boldsymbol{U},\boldsymbol{V})|\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V}>0\}$, we present a novel information criterion for PSS tracking of $\boldsymbol{A}$ as follows:

$$\min\left\{J(\boldsymbol{U},\boldsymbol{V})\right\}$$

$$J(\boldsymbol{U},\boldsymbol{V})=-\frac{1}{2}\mathrm{tr}(\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{D})-\mathrm{tr}\left(\frac{\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V}}{\|\boldsymbol{U}\|\|\boldsymbol{V}\|}\right)+\frac{1}{2}\mathrm{tr}(\boldsymbol{U}^T\boldsymbol{U}+\boldsymbol{V}^T\boldsymbol{V}), \tag{4}$$

where $\boldsymbol{D}$ is a diagonal matrix whose eigenvalues are all positive.

Based on the gradient of $J(\boldsymbol{U},\boldsymbol{V})$ respect to $\boldsymbol{U}$ and $\boldsymbol{V}$, we can obtain a dynamical system as

$$\dot{\boldsymbol{U}}=\boldsymbol{A}\boldsymbol{V}\boldsymbol{D}+\frac{\boldsymbol{A}\boldsymbol{V}}{\|\boldsymbol{U}\|\|\boldsymbol{V}\|}-\frac{\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V}}{\|\boldsymbol{U}\|^3\|\boldsymbol{V}\|}-\boldsymbol{U} \tag{5}$$

$$\dot{\boldsymbol{V}}=\boldsymbol{A}^T\boldsymbol{U}\boldsymbol{D}+\frac{\boldsymbol{A}^T\boldsymbol{U}}{\|\boldsymbol{U}\|\|\boldsymbol{V}\|}-\frac{\boldsymbol{V}\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V}}{\|\boldsymbol{U}\|\|\boldsymbol{V}\|^3}-\boldsymbol{V}. \tag{6}$$

Clearly, we can obtain an adaptive algorithm with normalized steps from the above dynamical system as

$$\tilde{\boldsymbol{U}}(k+1)=\boldsymbol{U}(k)+\eta\left\{\frac{\boldsymbol{A}(k+1)\boldsymbol{V}(k)(\boldsymbol{D}+\boldsymbol{I}_r)}{\|\boldsymbol{U}(k)\|\|\boldsymbol{V}(k)\|}-\boldsymbol{U}(k)\left[\frac{\boldsymbol{U}^T(k)\boldsymbol{A}(k+1)\boldsymbol{V}(k)}{\|\boldsymbol{U}(k)\|^3\|\boldsymbol{V}(k)\|}+\boldsymbol{I}_r\right]\right\} \tag{7}$$

$$\boldsymbol{U}(k+1)=\frac{\tilde{\boldsymbol{U}}(k+1)}{\|\boldsymbol{U}(k+1)\|} \tag{8}$$

$$\tilde{\boldsymbol{V}}(k+1)=\boldsymbol{V}(k)+\eta\left\{\frac{\boldsymbol{A}^T(k+1)\boldsymbol{U}(k)(\boldsymbol{D}+\boldsymbol{I}_r)}{\|\boldsymbol{U}(k)\|\|\boldsymbol{V}(k)\|}-\boldsymbol{V}(k)\left[\frac{\boldsymbol{U}^T(k)\boldsymbol{A}(k+1)\boldsymbol{V}(k)}{\|\boldsymbol{U}(k)\|\|\boldsymbol{V}(k)\|^3}+\boldsymbol{I}_r\right]\right\} \tag{9}$$

$$\boldsymbol{V}(k+1)=\frac{\tilde{\boldsymbol{V}}(k+1)}{\|\boldsymbol{V}(k+1)\|}, \tag{10}$$

where k is the time step, and η is the learning rate. If $\boldsymbol{x}$ and $\boldsymbol{y}$ are stationary, then $\boldsymbol{A}$ is updated by

$$\boldsymbol{A}(k+1)=\frac{k}{k+1}\boldsymbol{A}(k)+\frac{1}{k+1}\boldsymbol{x}(k+1)\boldsymbol{y}^T(k+1). \tag{11}$$

If $\boldsymbol{x}$ and $\boldsymbol{y}$ are nonstationary, then $\boldsymbol{A}$ is updated by

$$\boldsymbol{A}(k+1)=\alpha\boldsymbol{A}(k)+\boldsymbol{x}(k+1)\boldsymbol{y}^T(k+1). \tag{12}$$

Note that in the above algorithm two normalized steps (8) and (10) are added to ensure convergence. Since $\boldsymbol{U}(k)$ and $\boldsymbol{V}(k)$ are normalized at each step, we have $\|\boldsymbol{U}(k)\|=\|\boldsymbol{V}(k)\|=1,\ \forall\ k>0$. Substituting $\|\boldsymbol{U}(k)\|=\|\boldsymbol{V}(k)\|=1$ into (7)–(9) and ignoring the normalize step, it yields