ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746546

SPARSE SUBSPACE TRACKING IN HIGH DIMENSIONS

Le Trung Thanh[⋆,†], *Karim Abed-Meraim*[⋆], *Adel Hafiane*[⋆], *and Nguyen Linh Trung*[†]

[⋆]PRISME Laboratory, University of Orléans/INSA-CVL, France
[†]AVITECH Institute, VNU University of Engineering and Technology, Vietnam

trung-thanh.le@univ-orleans.fr, linhtrung@vnu.edu.vn

ABSTRACT

We studied the problem of sparse subspace tracking in the high-dimensional regime where the dimension is comparable to or much larger than the sample size. Leveraging power iteration and thresholding methods, a new provable algorithm called OPIT was derived for tracking the sparse principal subspace of data streams over time. We also presented a theoretical result on its convergence to verify its consistency in high dimensions. Several experiments were carried out on both synthetic and real data to demonstrate the effectiveness of OPIT.

Index Terms— Sparse subspace tracking, data streams, high dimensions, thresholding, power iteration.

1. INTRODUCTION

Subspace tracking (ST) is an essential and fundamental problem in signal processing with various applications to sensor array processing, wireless communication, and image/video processing, to name a few [1]. It corresponds to the problem of tracking a low-rank subspace that can represent data streams. Most of subspace tracking methods are designed to estimate the underlying subspace from the empirical/sample covariance matrix. We refer the reader to [1–3] for good surveys on standard and robust ST algorithms.

In recent years, many theoretical results in random matrix theory (e.g. [4–6]) indicated that the sample covariance matrix is not a good estimator of the actual covariance matrix in the high-dimensional regime where datasets are massive in both dimension n and sample size T, and typically $n/T \to c \in (0, \infty]$. It might occur due to the time variation of the principal subspace, and hence the "effective" sample size of the considered processing window is limited accordingly. Without further structural knowledge about the data, subspace tracking algorithms turn out to be inconsistent in such a regime. Interestingly, the consistency of covariance estimation can be guaranteed under suitably structured sparsity regularizations [7–10]. Therefore, sparse subspace estimation and tracking have recently gained much attention in the signal processing community. In the literature, several good methods have been proposed for sparse subspace estimation, see [11–13] for examples and [14] for a comprehensive survey. However, in an adaptive (online) setting, there have been few studies on sparse subspace tracking (SST) so far.

Related works: As mentioned before, some online algorithms have been introduced for sparse subspace tracking [3]. A few of them are based on a two-stage approach in which one first utilizes a standard ST algorithm to estimate the underlying subspace and then seek a sparse basis of the estimation under some sparsity criteria. Particularly in [15–17], several variants of OPAST and FAPI were proposed to track the sparse principal subspace. Another good approach is to regularize the objective function that aims at accounting for the sparse basis. In [18], the authors modified the objective function of PAST by adding a ℓ_1-norm regularization term on the subspace matrix and then proposed a new robust variant of PAST called ℓ_1-PAST to optimize it. In [19], a Bayesian-based algorithm called OVBSL was proposed to deal with the sparsity constraint on the subspace matrix. An advantage of OVBSL is that it is fully automated, i.e., no finetuning parameter is required. However, these algorithms are only effective in the classical regime where the sample size is much larger than the dimension, i.e., $n/T \to 0$ asymptotically.

Through the lens of machine learning and statistics, SST is generally referred to as the problem of online sparse PCA which often emphasizes the leading eigenvectors. In [20], the authors proposed an extended version of the Oja algorithm for online sparse PCA, namely OIST. Its convergence, steady-state, and phase transition were also derived to investigate the use of OIST in high dimensions. OIST is, however, designed only for rank-1 sparse subspaces. In [21], another online sparse PCA algorithm was proposed and could deal with rank-k subspaces. Specifically, this algorithm uses a simple row truncation operator, which sets rows whose scores are smaller than a threshold to zero, for tracking the sparse principal subspace over time. However, this truncation operator is only designed for subspaces with a row-sparse support (i.e. all eigenvectors must share the same sparsity patterns) which may not always meet in practice. Indeed, it turns out to be ineffective for a sparse subspace with another support (e.g. elementwise sparsity). Its performance in terms of estimation accuracy is typically lower than other SST algorithms, see Fig. 1 for an illustration.

Contributions: In this paper, we introduce a new adaptive algorithm called OPIT (OPIT stands for Online Power Iteration via Thresholding) for sparse subspace tracking. OPIT takes both advantages of power iteration and thresholding methods, and hence offers several appealing features over the state-of-the-art SST/online sparse PCA algorithms. Among them is that OPIT is capable of tracking the sparse principal subspace in both classical regime and high-dimensional regime. In addition, its procedure is flexible and very adaptable for dealing with multiple incoming data streams. This feature is useful for application areas wherein a block processing is required, i.e., a block of data samples is processed and analysed at one time. Also, OPIT belongs to the class of provable subspace tracking algorithms in which its convergence is guaranteed.

2. PROBLEM FORMULATION

Assume that at time t, we collect a data sample $\mathbf{x}_t \in \mathbb{R}^{n\times 1}$ satisfying the standard signal model

$$\mathbf{x}_t = \boldsymbol{\ell}_t + \mathbf{n}_t. \quad (1)$$

Here, $\boldsymbol{\ell}_t \in \mathbb{R}^{n\times 1}$ is a low-rank signal living in a subspace[1] spanned by a sparse matrix $\mathbf{A}^{n\times r}$ with $r \ll n$ (i.e. $\boldsymbol{\ell}_t = \mathbf{A}\mathbf{w}_t$, where $\mathbf{w}_t \in \mathbb{R}^{r\times 1}$ is a weight vector) and $\mathbf{n}_t \in \mathbb{R}^{n\times 1}$ is an additive spatially white noise vector independent of $\boldsymbol{\ell}_t$. Sparse subspace tracking (SST) problem can be stated as follows:

SST Problem: *Given a streaming set of data samples $\{\mathbf{x}_t\}_{t=1}^T$, we aim to estimate a sparse principle subspace $\mathbf{A}_t$ that compactly represents the span of signals $\{\boldsymbol{\ell}_t\}_{t=1}^T$.*

Generally, the underlying subspace can be estimated from the spectral analysis of the actual covariance matrix

$$\mathbf{C} = \mathbb{E}\{\mathbf{x}_t\mathbf{x}_t^\top\} = \mathbf{A}\mathbb{E}\{\mathbf{w}_t\mathbf{w}_t^\top\}\mathbf{A}^\top + \mathbb{E}\{\mathbf{n}_t\mathbf{n}_t^\top\}. \quad (2)$$

Applying eigenvalue decomposition (EVD) on $\mathbf{C}$ results in

$$\mathbf{C} \overset{\text{EVD}}{=} \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top = \begin{bmatrix}\mathbf{U}_s & \mathbf{U}_n\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_s & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Lambda}_n\end{bmatrix}\begin{bmatrix}\mathbf{U}_s^\top\\ \mathbf{U}_n^\top\end{bmatrix}. \quad (3)$$

Here, $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose diagonal elements are eigenvalues of $\mathbf{C}$ sorted in decreasing order and $\mathbf{U} \in \mathbb{R}^{n\times n}$ contains the corresponding eigenvectors. Accordingly, $\mathbf{U}_s \in \mathbb{R}^{n\times r}$ and $\mathbf{U}_n \in \mathbb{R}^{n\times(n-r)}$ represent the principal subspace and the minor subspace of $\mathbf{C}$, respectively. The orthogonal projection matrix of the sparse principal subspace is unique (i.e., $\mathbf{U}_s\mathbf{U}_s^\top = \mathbf{A}\mathbf{A}^\#$ where $(.)^\#$ is the pseudo-inverse operator), so $\mathbf{A}$ can be obtained as $\mathbf{A} = \mathbf{U}_s\mathbf{Q}^*$ with

$$\mathbf{Q}^* = \underset{\mathbf{Q}\in\mathbb{R}^{r\times r}}{\operatorname{argmin}} \|\mathbf{U}_s\mathbf{Q}\|_0 \text{ s.t. } \mathbf{Q} \text{ is full-rank}, \quad (4)$$

where $\|.\|_0$ promotes the sparsity on $\mathbf{A}$. In several applications, we often emphasize the principal subspace rather than its specific basis, such as dimensionality reduction [22] and array processing [23]. In this work, our main objective is to track the principal (signal) subspace of $\mathbf{A}$ while the sparsifying step (4) is optional.

Most state-of-the-art SST algorithms estimate the principal subspace of the sample covariance matrix $\mathbf{C}_T = 1/T\sum_{t=1}^T \mathbf{x}_t\mathbf{x}_t^\top$ [3]. However, in a high-dimensional regime where $n/T \nrightarrow 0$ a.s., $\mathbf{C}_T$ is not a good estimator of $\mathbf{C}$. This limitation in an adaptive scheme is not necessarily due to a data shortage but to the time variation which forces us to use a limited window of time instead of all the data. Particularly, the relation between $\mathbf{C}_T$ and $\mathbf{C}$ is specified by the following proposition.

Proposition 1. *The following error matrix is bounded in the operator norm with a probability at least $1-\delta$:*

$$\|\mathbf{C} - \mathbf{C}_T\|_2 \le c_\delta\Big(\sigma_x^2\sqrt{\frac{r}{T}} + \big(2\sigma_n\sigma_x + \sigma_n^2\big)\sqrt{\frac{n}{T}}\Big), \quad (5)$$

where $\sigma_x^2 = \mathbb{E}\{\|\boldsymbol{\ell}_t\|^2\}$, $\sigma_n^2 = \mathbb{E}\{\|\mathbf{n}_t\|^2\}$, and $c_\delta = C\sqrt{\log(2/\delta)}$ with a universal positive number $C > 0$.

Due to the space limitation, we omit its proof here. As a result, most of SST algorithms are not good in the high-dimensional regime, as illustrated in Fig. 1(c)-(d).

Under certain conditions, it is proved in [7, 24, 25] that

$$\|\mathbf{C} - \eta(\mathbf{C}_T)\|_2 \to 0 \ \text{ a. s. as } T \to \infty, \quad (6)$$

where $\eta(.)$ is an appropriate thresholding operator. Thanks to (6), in the next section, we derive a novel adaptive (online) algorithm based on power iteration and thresholding technique that is capable of tracking the sparse principal subspace in both classical and high-dimensional regime.

3. PROPOSED METHOD

In this section, we introduce a novel sparse subspace tracking algorithm namely OPIT which main steps are summarized in Algorithm 1.

3.1. Assumptions

In what follows, we make some assumptions which are widely-used in the theoretical analysis of subspace tracking algorithms. In particular, (A1)-(A3) are necessary assumptions to establish the convergence of OPIT.

(A1) The sparse basis matrix $\mathbf{A}$ is characterized by $\mathbf{A} = \boldsymbol{\Omega}\circledast\mathbf{U}$. Here, $\circledast$ denotes the Hadamard product, $\boldsymbol{\Omega} \in \mathbb{R}^{n\times r}$ is a binary mask whose entries are i.i.d. Bernoulli random variables with a probability $1-\rho$, and $\mathbf{U}$ is a matrix chosen in the set $\mathcal{U} \triangleq \{\mathbf{U} \in \mathbb{R}^{n\times r}, \|\mathbf{u}_k\|_2 \le 1, 1 \le \kappa(\mathbf{U}) \le \alpha\}$ where $\kappa(\mathbf{U})$ denotes the condition number of $\mathbf{U}$ and $1 \le \alpha < \infty$. Accordingly, the parameter ρ represents the sparsity level of $\mathbf{A}$. Moreover, $\mathbf{A}$ is sparse enough in the sense, the average number of non-zero elements in each column is upper bound by $\sqrt{n\log n}$, i.e., $\rho \ge 1 - \sqrt{\log n/n}$. Together with the Bernoulli matrix $\boldsymbol{\Omega}$, this condition allows trackers to correctly recover the sparse subspace [26]. The constraint set $\mathcal{U}$ is to prevent very large entries in $\mathbf{A}$ and ill-conditioned problems.

(A2) Data samples $\{\mathbf{x}_t\}_{t\ge1}$ are norm bounded, i.e., $\|\mathbf{x}_t\|_2 \le M < \infty\ \forall t$. Low-rank signals $\{\boldsymbol{\ell}_t\}_{t\ge t}$ are supposed to be deterministic and bounded. Noise vectors $\{\mathbf{n}_t\}_{t\ge1}$ are i.i.d. random variables of zero mean and their power is lower than the signal power. Indeed, (A2) is a common assumption for subspace tracking problems and holds in many applications [27].

(A3) Subspace coefficient vectors $\{\mathbf{w}_t\}_{t\ge1}$ are constrained to the set $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^{r\times 1}, \omega_1 \le |\mathbf{w}(i)| \le \omega_2\}$ with $0 \le \omega_1 < \omega_2 < \infty$. Since both $\mathbf{A}$ and $\boldsymbol{\ell}_t$ are bounded, $\mathbf{w}_t$ is naturally bounded.

3.2. OPIT Algorithm

We first recall the main steps of the standard power iteration (PI) method on which we primarily leverage in order to develop our OPIT algorithm, for computing the dominant eigenvectors of $\mathbf{C}_t$. At the ℓ-th iteration, PI particularly updates (i) $\mathbf{S}_\ell \leftarrow \mathbf{C}_t\mathbf{U}_{\ell-1}$ and (ii) $\mathbf{U}_\ell \leftarrow \text{QR}(\mathbf{S}_\ell)$ be the Q-factor of QR factorization of $\mathbf{S}_\ell$. PI starts from an initial matrix $\mathbf{U}_0 \in \mathbb{R}^{n\times r}$ and returns an orthonormal matrix $\mathbf{U}_L$ where L is the number of iterations [1].

[1] In an adaptive scheme, the matrix $\mathbf{A}$ may be slowly varying with time, i.e., $\mathbf{A} = \mathbf{A}_t$. Our algorithm is capable of successfully estimating the subspace as well as tracking its variation along the time.

Algorithm 1: Online Power Iteration by Thresholding (OPIT)

Inputs: $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}, \mathbf{x}_i \in \mathbb{R}^{n \times 1}$, true rank r, window of length $W \geq 1$, a forgetting factor $0 < \lambda \leq 1$, and a thresholding factor m

$$m = \begin{cases} \text{round}\big((1-\rho)n\big) & \text{if } \rho \text{ is given,} \\ \text{round}\big(10r\log n\big) & \text{if } \rho \text{ is unknown,} \end{cases}$$

where ρ is the sparsity level of the sparse basis.

Initial: $\mathbf{U}_0 = \text{randn}(n, r), \mathbf{S}_0 = \mathbf{0}_{n\times r}, \mathbf{E}_0 = \mathbf{0}_{r \times r}$

Procedure:

for $t = 1, \ldots, T/W$ **do**

1. $\mathbf{X}_t = [\mathbf{x}_{(t-1)W+1}, \ldots, \mathbf{x}_{tW}]$
2. $\mathbf{Z}_t = \mathbf{U}_{t-1}^\top \mathbf{X}_t$
3. $\mathbf{S}_t = \lambda \mathbf{S}_{t-1}\mathbf{E}_{t-1} + \mathbf{X}_t \mathbf{Z}_t^\top$
4. $\hat{\mathbf{S}}_t = \eta\big(\mathbf{S}_t, m\big)$
5. $\mathbf{U}_t = \text{QR}\big(\hat{\mathbf{S}}_t\big)$
6. $\mathbf{E}_t = \mathbf{U}_{t-1}^\top \mathbf{U}_t$

Output: $\mathbf{U}_t \in \mathbb{R}^{n \times r}$

In an adaptive scheme, the iteration step of PI can coincide with the data collection in time. At time t, the sample covariance matrix $\mathbf{C}_t$ can be recursively updated by: $\mathbf{R}_t = \mathbf{R}_{t-1} + \mathbf{x}_t\mathbf{x}_t^\top$ and $\mathbf{C}_t = \frac{1}{t}\mathbf{R}_t$. Accordingly, we can rewrite the first step of PI as

$$\mathbf{S}_t = \mathbf{S}_{t-1}\mathbf{E}_{t-1} + \mathbf{x}_t\mathbf{z}_t^\top, \tag{7}$$

where $\mathbf{E}_{t-1} = \mathbf{U}_{t-2}^\top \mathbf{U}_{t-1}$ and $\mathbf{z}_t = \mathbf{U}_{t-1}^\top \mathbf{x}_t$. In this work, the update (7) is further followed by an appropriate perturbation $\mathbf{G}_t$ defined by the following thresholding operation $\eta(.)$ as:

$$\hat{\mathbf{S}}_t \stackrel{\Delta}{=} \eta(\mathbf{S}_t, m) = \mathbf{C}_t\mathbf{U}_{t-1} + \mathbf{G}_t, \tag{8}$$

where the factor m can be determined as in Algorithm 1. Here, $\hat{\mathbf{S}}_t$ is particularly derived from $\mathbf{S}_t$ by keeping the m strongest (absolute value) elements in each column of $\mathbf{S}_t$ and setting the remaining elements to zero. Then, the second step of PI is replaced with $\mathbf{U}_t \leftarrow \text{QR}(\hat{\mathbf{S}}_t)$. In addition to the nice property (6), another main motivation for using the thresholding operation $\eta(.)$ stems from the following proposition:

Proposition 2. *When the perturbation $\mathbf{G}_t$ satisfies: $\|\mathbf{G}_t\|_2 \leq \xi\sigma_x^2$ and $4\|\mathbf{A}_t^\top \mathbf{G}_t\|_2 \leq \sigma_x^2 \cos\theta(\mathbf{A}_t, \mathbf{U}_{t-1})$ for some positive $\xi < 1$, we obtain*

$$\tan\theta(\mathbf{A}_t, \mathbf{C}_t\mathbf{U}_{t-1} + \mathbf{G}_t) \leq \gamma \tan\theta(\mathbf{A}_t, \mathbf{U}_{t-1}),$$

where $0 < \gamma \leq 1$ and $\theta(.,.)$ denotes the canonical angle (the largest principal angle) between two subspaces.

Proof of Proposition 2 follows immediately Lemma 2.2 in [28]. As a corollary, $\mathbf{U}_t$ will get closer to $\mathbf{A}_t$ over time.

The proposed OPIT algorithm also introduces two other parameters λ and W. Here, the forgetting factor λ $(0 < \lambda \leq 1)$ is aimed at discounting the impact of old observations as well as facilitating the tracking ability of OPIT in time-varying environments. The inclusion of W is useful in some applications where we often collect multiple data samples instead of a single sample at each time t.

Computational complexity of OPIT is of order $\mathcal{O}(nr^2)$ while it only requires a space of $\mathcal{O}(2nr + r^2)$ for saving $\mathbf{U}_t$, $\mathbf{S}_t$, and $\mathbf{E}_t$ at each time t. Convergence of OPIT can be specified by Lemma 1 whose proof is omitted due to the space limitation.

Lemma 1. *Suppose that assumptions (A1)-(A3) are met, the true basis $\mathbf{A}$ is deterministic and unchanged over time, $\lambda = 1$, and that the initialization matrix $\mathbf{U}_0$ and the number of data samples satisfy the following conditions*

$$t \geq \frac{c_\delta}{W\epsilon^2}\Big(\sqrt{r} + (2\sigma_n/\sigma_x + \sigma_n^2/\sigma_x^2)\sqrt{n}\Big)^2, \tag{9}$$

$$\tan\theta\big(\mathbf{A}, \mathbf{U}_0\big) \leq \frac{\sigma_x^2 + \sigma_n^2}{(1 + \sqrt{r}(1+\sqrt{2}))\sigma_x^2 - (2+\sqrt{2})\sigma_n^2}, \tag{10}$$

with a small predefined error ϵ and a positive number $c_\delta = C\sqrt{\log(2/\delta)}$ where $0 < \delta \ll 1$ and C is a universal positive number. If $\mathbf{U}_\tau$ is generated by OPIT at time t, then

$$\sin\theta(\mathbf{A}, \mathbf{U}_t) \leq \epsilon, \tag{11}$$

with a probability at least $1 - \delta$.

4. EXPERIMENTAL RESULTS

In this section, we conduct some experiments on both synthetic and real data to demonstrate the effectiveness of OPIT. Performance of OPIT is evaluated in comparison with the state-of-the-art algorithms. Our simulations are implemented using MATLAB on a laptop of Intel core $i7$ and 16GB of RAM.[2]

4.1. Experiments With Synthetic Data

Following the formulation in Section 2, data samples $\{\mathbf{x}_t\}_{t\geq 1}$ are randomly generated under the standard model

$$\mathbf{x}_t = \mathbf{A}_t\mathbf{w}_t + \sigma_n\mathbf{n}_t, \tag{12}$$

where $\mathbf{n}_t \in \mathbb{R}^{n\times 1}$ is a noise vector derived from $\mathcal{N}(0, \mathbf{I}_n)$, $\sigma_n > 0$ is to control the effect of the noise on algorithm performance, $\mathbf{w}_t \in \mathbb{R}^{r \times 1}$ is an i.i.d. Gaussian random vector of zero-mean and unit-variance to represent the subspace coefficient. The sparse matrix $\mathbf{A}_t \in \mathbb{R}^{n\times r}$ at each time t is simulated as $\mathbf{A}_t = \mathbf{\Omega} \circledast (\mathbf{A}_{t-1} + \varepsilon\mathbf{N}_t)$, where $\mathbf{\Omega} \in \mathbb{R}^{n \times r}$ is a Bernoulli random matrix with probability $1-\rho$, $\mathbf{N}_t$ is a normalized Gaussian white noise matrix, and $\varepsilon > 0$ is the time-varying factor aimed to control the subspace variation over time. Simulation results are averaged over 10 independent runs.

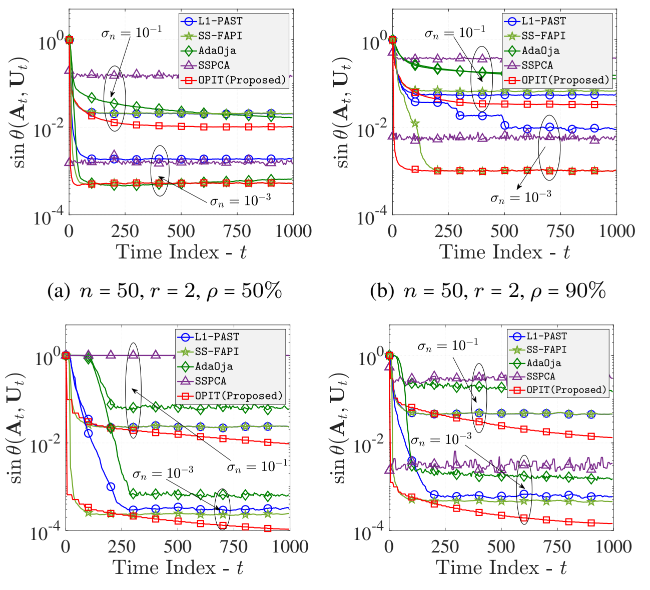
In order to evaluate the subspace estimation performance, we measure the following distance between two subspaces

$$d(\mathbf{A}_t, \mathbf{U}_t) \stackrel{\Delta}{=} \sin\theta(\mathbf{A}_t, \mathbf{U}_t), \tag{13}$$

where $\mathbf{U}_t$ refers to the estimated subspace at time t.

In these experiments, we investigated the performance of OPIT in the classical and high-dimensional regimes. We used 1000 independent data samples from Eq. (12) in which the time-varying factor ε was fixed at 10^{-3} and the value of σ_n was set to two levels: 10^{-1} and 10^{-3}. Here, two sparsity levels were also considered, including 50% and 90%. The length of window was set to $W = \lfloor \log n \rceil$ for the cases of high dimensions and low SNR levels, while we used $W = 1$ for others. We fixed

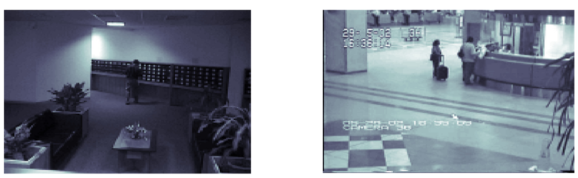
[2] MATLAB Code: https://github.com/thanhtbt/SST/.



(a) n = 50, r = 2, ρ = 50% (b) n = 50, r = 2, ρ = 90%

(c) n = 10000, r = 10, ρ = 50% (d) n = 10000, r = 10, ρ = 90%

Fig. 1: Time-varying environments: data samples T = 1000 and time-varying factor $\varepsilon = 10^{-3}$.



a) Lobby b) Hall

Fig. 2: Two video sequences used in this paper.

the forgetting factor λ at 0.97 for all simulations. The tracking ability of OPIT was compared to the state-of-the-art algorithms including ℓ_1-PAST [18], SS-FAPI [17], SSPCA [21], and AdaOja [29]. Their parameters were kept default to have a fair comparison.

Experimental results are shown as in Fig. 1. We can see that in the classical regime, OPIT was one of the two best effective SST algorithms, together with SS-FAPI, see Fig. 1(a)-(b). At a high sparsity level (i.e. ρ = 90%), the two algorithms provided much better estimation accuracy than ℓ_1-PAST, SSPCA, and AdaOja. Indeed, OPIT's convergence rate was faster than SS-FAPI. At a high noise level (i.e. $\sigma_n = 10^{-1}$), OPIT provided the best subspace estimation accuracy. In the high dimensional regime, OPIT outperformed other SST algorithms completely at both low and high levels of noise as well as sparsity, as shown in Fig. 1(c)-(d).

4.2. Experiments With Real Video Data

Two video sequences were used to illustrate the effectiveness of OPIT, including "Lobby" and "Hall" (see Fig. 2 for an illustration). In these experiments, we compared the performance of OPIT with the state-of-the-art subspace tracking algorithms (i.e., ℓ_1-PAST, SS-FAPI, and PETRELS-ADMM [27]) and tensor tracking algorithms (i.e., SOAP [30], OLCP [31], OLSTEC [32], and ROLCP [33]). In order to apply these subspace tracking algorithms to the video sequences, each video

Dataset	"Lobby"		"Hall"	
Tensor size	$128 \times 160 \times 1546$		$174 \times 144 \times 3584$	
Matrix size	20480×1546		25056×3584	
Evaluation metrics	time(s)	error	time(s)	error
SOAP	14.29	0.842	21.72	0.989
OLCP	10.50	0.161	19.98	0.154
OLSTEC	44.25	0.037	92.82	0.041
ROLCP	4.32	0.114	10.74	0.120
PETRELS-ADMM	118.4	0.015	305.5	0.018
ℓ_1-PAST	14.11	0.031	33.73	0.101
SS-FAPI	12.99	0.023	32.72	0.100
OPIT ($W = 1$)	16.32	0.013	50.78	0.056
OPIT ($W = \lfloor \log(IJ) \rfloor$)	1.89	0.021	5.62	0.086

Table 1: Runtime and averaged relative error of adaptive algorithms on tracking the four video sequences.

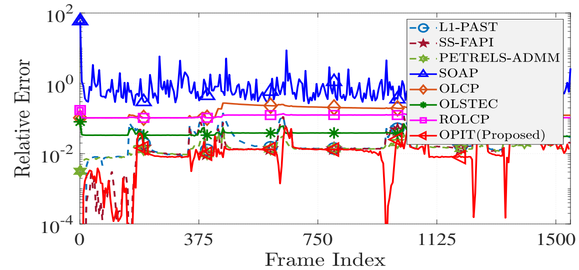


Fig. 3: Tracking ability of algorithms on the "Lobby" data.

frame of size $I \times J$ was reshaped into a $IJ \times 1$ vector. Following the studies on video tracking in [27] and [33], the tensor rank and subspace rank were set to 10 for all simulations.

Simulation results are shown statistically in Tab. 1 and graphically in Fig. 3. As can be seen that OPIT provided a competitive estimation accuracy as compared to PETRELS-ADMM while its runtime was much faster than that of the ADMM-based tracking algorithm. Indeed, OPIT had a better performance than PETRELS-ADMM on the "Lobby" data. Also, OPIT outperformed most tracking algorithms, apart from PETRELS-ADMM. With respect to runtime, ROLCP was the fastest "one-pass" tracking algorithm, several times faster than the second-best. Interestingly, our algorithm is also designed for handling a block of multiple incoming samples at each time (i.e. the length of window $W > 1$). When $W = \lfloor \log(IJ) \rfloor$, OPIT can be even faster than ROLCP while still retaining a reasonable tracking accuracy.

5. CONCLUSIONS

In this paper, we have proposed a new provable sparse subspace tracking algorithm, called OPIT, capable of tracking the sparse principal subspace in both classical regime and high dimensional regime. In the classical regime, OPIT provides a competitive subspace estimation accuracy and its convergence rate is very fast at a high SNR level. In the high-dimensional regime, OPIT outperforms other sparse subspace tracking algorithms, its estimation accuracy is much better than that of the second-best, SS-FAPI. Simulations carried out on video sequences of different scenarios have indicated that OPIT has potential for real applications.

6. REFERENCES

[1] J. Delmas, “Subspace tracking for signal processing,” *Adaptive Signal Processing: Next Generation Solutions*, pp. 211–270, 2010.

[2] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, “Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery,” *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, 2018.

[3] L. T. Thanh, N. V. Dung, N. L. Trung, and K. Abed-Meraim, “Robust subspace tracking algorithms in signal processing: A brief survey,” *REV J. Elect. Commun.*, vol. 11, no. 1-2, pp. 16–25, 2021.

[4] N. El. Karoui, “Spectrum estimation for large dimensional covariance matrices using random matrix theory,” *Ann. Stat.*, vol. 36, no. 6, pp. 2757–2790, 2008.

[5] X. Mestre, “On the asymptotic behavior of the sample estimates of eigenvalues and eigenvectors of covariance matrices,” *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5353–5368, 2008.

[6] R. Vershynin, “How close is the sample covariance matrix to the actual covariance matrix?,” *J. Theor. Probab.*, vol. 25, no. 3, pp. 655–686, 2012.

[7] N. El Karoui, “Operator norm consistent estimation of large-dimensional sparse covariance matrices,” *Ann. Stat.*, vol. 36, no. 6, pp. 2717–2756, 2008.

[8] C. Lam and J. Fan, “Sparsistency and rates of convergence in large covariance matrix estimation,” *Ann. Stat.*, vol. 37, no. 6B, pp. 4254, 2009.

[9] I. M. Johnstone and A. U. Lu, “On consistency and sparsity for principal components analysis in high dimensions,” *J. Am. Stat. Assoc.*, vol. 104, no. 486, pp. 682–693, 2009.

[10] P. J. Bickel and E. Levina, “Covariance regularization by thresholding,” *Ann. Stat.*, vol. 36, no. 6, pp. 2577–2604, 2008.

[11] A. A. Arash and J. W. Martin, “High-dimensional analysis of semidefinite relaxations for sparse principal components,” *Ann. Stat.*, vol. 37, no. 5B, pp. 2877 – 2921, 2009.

[12] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, “Generalized power method for sparse principal component analysis.,” *J. Mach. Learn. Res.*, vol. 11, no. 2, pp. 517–553, 2010.

[13] Z. Ma, “Sparse principal component analysis and iterative thresholding,” *Ann. Stat.*, vol. 41, no. 2, pp. 772–801, 2013.

[14] T. Cai, Z. Ren, and H. H. Zhou, “Estimating structured high-dimensional covariance and precision matrices: Optimal rates and adaptive estimation,” *Electron. J. Stat.*, vol. 10, no. 1, pp. 1–59, 2016.

[15] N. Lassami, K. Abed-Meraim, and A. Aïssa-El-Bey, “Low cost subspace tracking algorithms for sparse systems,” in *Eur. Signal Process. Conf.*, 2017, pp. 1400–1404.

[16] N. Lassami, A. Aïssa-El-Bey, and K. Abed-Meraim, “Fast sparse subspace tracking algorithm based on shear and givens rotations,” in *Asilomar Conf. Signals Syst. Comput.*, 2019, pp. 1667–1671.

[17] N. Lassami, A. Aïssa-El-Bey, and K. Abed-Meraim, “Low cost sparse subspace tracking algorithms,” *Signal Process.*, vol. 173, pp. 107522, 2020.

[18] X. Yang, Y. Sun, T. Zeng, T. Long, and T. K. Sarkar, “Fast STAP method based on PAST with sparse constraint for airborne phased array radar,” *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4550–4561, 2016.

[19] P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, and K. D. Koutroumbas, “Online sparse and low-rank subspace learning from incomplete data: A Bayesian view,” *Signal Process.*, vol. 137, pp. 199–212, 2017.

[20] C. Wang and Y. M. Lu, “Online learning for sparse PCA in high dimensions: Exact dynamics and phase transitions,” in *IEEE Inf. Theory Works.*, 2016, pp. 186–190.

[21] W. Yang and H. Xu, “Streaming sparse principal component analysis,” in *Int. Conf. Mach. Learn.*, 2015, pp. 494–503.

[22] L. Van Der Maaten, E. Postma, and J. Van den Herik, “Dimensionality reduction: A comparative review,” *J. Mach. Learn. Res.*, vol. 10, no. 66-71, pp. 13, 2009.

[23] N. V. Dung, K. Abed-Meraim, N. L. Trung, and R. Weber, “Generalized minimum noise subspace for array processing,” *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3789–3802, 2017.

[24] A. J. Rothman, E. Levina, and J. Zhu, “Generalized thresholding of large covariance matrices,” *J. Am. Stat. Assoc.*, vol. 104, no. 485, pp. 177–186, 2009.

[25] T. Cai and M. Yuan, “Adaptive covariance matrix estimation through block thresholding,” *Ann. Stat.*, vol. 40, no. 4, pp. 2014–2042, 2012.

[26] D. A. Spielman, H. Wang, and J. Wright, “Exact recovery of sparsely-used dictionaries,” in *PMLR Ann. Conf. Learn. Theory*, 2012, vol. 23, pp. 37.1–37.18.

[27] L. T. Thanh, N. V. Dung, N. L. Trung, and K. Abed-Meraim, “Robust subspace tracking with missing data and outliers: Novel algorithm with convergence guarantee,” *IEEE Trans. Signal Process.*, vol. 69, pp. 2070–2085, 2021.

[28] M. Hardt and E. Price, “The noisy power method: A meta algorithm with applications,” in *Adv. Neural Inf. Process. Syst.*, 2014, vol. 27, pp. 2861–2869.

[29] Z. Allen-Zhu and Y. Li, “First efficient convergence for streaming k-PCA: A global, gap-free, and near-optimal rate,” in *IEEE Ann. Symp. Found. Comput. Sci.*, 2017, pp. 487–492.

[30] N. V. Dung, K. Abed-Meraim, and N. L. Trung, “Second-order optimization based adaptive PARAFAC decomposition of three-way tensors,” *Digit. Signal Process.*, vol. 63, pp. 100–111, 2017.

[31] S. Zhou, N. X. Vinh, J. Bailey, Y. Jia, and I. Davidson, “Accelerating online CP decompositions for higher order tensors,” in *ACM Int. Conf. Knowl. Discover. Data Min.*, 2016, pp. 1375–1384.

[32] H. Kasai, “Fast online low-rank tensor subspace tracking by CP decomposition using recursive least squares from incomplete observations,” *Neurocomput.*, vol. 347, pp. 177 – 190, 2019.

[33] L. T. Thanh, K. Abed-Meraim, N. L. Trung, and A. Hafiane, “A fast randomized adaptive CP decomposition for streaming tensors,” in *IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 2910–2914.