Online Low-Rank + Sparse Structure Learning for Dynamic Network Tracking

Alp Ozdemir and Selin Aviyente

ozdemira@msu.edu, aviyente@egr.msu.edu

Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48824, USA.

***Abstract*—Recent developments in information technology have enabled us to collect and analyze high dimensional and higher order data such as tensors. High dimensional data usually lies in a lower dimensional subspace and identifying this low-dimensional structure is important in many signal and information processing applications. Traditional subspace estimation approaches have been limited to vector-type data and cannot effectively deal with these high order datasets. Moreover, most of the existing methods are batch algorithms which can't handle streaming data. In this paper, we propose a new tensor subspace tracking approach to identify changes in dynamic networks. The proposed approach recursively estimates low-rank subspace of higher order data and decomposes it into low-rank and sparse components. The proposed approach is evaluated on both simulated and real dynamic networks.**

***Index Terms*—Tensor algebra, robust principal component analysis, low-rank tensor approximation, tensor subspace tracking.**

I. Introduction

High dimensional data mostly lies in a lower dimensional subspace and principal component analysis (PCA) is the most widely used technique to identify this lower dimensional subspace. However, PCA suffers much from corrupted data and may find a completely wrong principal subspace in the presence of even a few outliers. These drawbacks have forced researchers to develop more robust subspace estimation techniques, e.g. Robust PCA (RPCA) which is a significantly more difficult problem than standard PCA [1], [2].

Since the recent work by Candes et al. and Chandrasekaran et al. [3], [4], the general problem of separating a sparse matrix and a low-rank matrix from their sum has received a lot of attention. The final goal usually is to either find the column span of the low-rank matrix or the support of the sparse one. This is now commonly referred to as the "low-rank + sparse recovery" problem. There has been a large amount of recent work on batch methods for low-rank + sparse recovery and its various extensions such as Principal Component Pursuit, Outlier Pursuit and Low-Leverage Decomposition [5]–[18].

With the increase in the amount of streaming data, approaches developed for static data become limited. For streaming data, using recursive approaches are more advantageous to reduce the computational complexity and to achieve real-time subspace tracking. Various online approaches have been proposed to solve the RPCA problem, i.e. GRASTA, PETRELS and REPROCS [19]–[27]. These approaches first identify the subspace that the low rank data lies in, then recovers incoming low-rank measurement vectors from missing entries by considering this subspace information. To address this issue, GRASTA performs incremental gradient descent on Grassman manifolds to estimate and track the subspace, while PETRELS uses least squares estimation recursively. REPROCS on the other hand applies sparse recovery to provide better estimation for the sparse noise at each time point, then subtracts the sparse component from the measurement vector to obtain an estimate of the low-rank part. All of these methods are designed for vector type measurements, and cannot be applied directly to higher order datasets such as tensors.

Recently, some computationally efficient approaches have been proposed to track and update subspace information for dynamic tensors such as dynamic tensor analysis, streaming tensor analysis and window based tensor analysis [28], [29]. However, the main focus of these approaches is analyzing streaming datasets efficiently without addressing the robustness of the subspace estimates to outliers. Moreover, these methods do not necessarily recover a low-rank structure and require updates at each time point. More recently, Mardani et al. [30] proposed an online subspace learning method based on nuclear norm minimization and extended this approach for matrices and higher order datasets. Extension of this algorithm to tensors takes advantage of parallel factor analysis (PARAFAC) model to minimize tensor rank and considers temporal information as one of the tensor modes.

In this paper, we propose a low-rank plus sparse structure learning algorithm for higher order datasets such as matrices and tensors. The proposed approach relies on recovering a low-rank subspace estimate along each mode of the data where the rank is defined through the Tucker rank. This rank definition is directly related to the community structure in network type data. In this paper, we adapted a recently introduced vector based online algorithm REPROCS [24] to 2-way tensors (matrices) in order to track dynamic networks. The goal of the proposed approach is to separate the low-rank part of the data from sparse noise components in real time by updating the estimates of low-rank subspaces. We evaluate the proposed framework by applying it to a set of simulated and real networks.

This work was in part supported by NSF CCF-1422262 and NSF CCF-1218377.

II. Background

A. Recursive Projected Compressive Sensing (REPROCS)

Let $\mathbf{M}_t \in \mathbb{R}^{n\times 1}$ be a time-series of measurement vectors written as $\mathbf{M}_t = \mathbf{L}_t + \mathbf{S}_t$ where $\mathbf{L}_t$ is the low-rank component which lies in a subspace spanned by $\mathbf{P}_t$ and $\mathbf{S}_t$ is the sparse noise vector. Let $\hat{\mathbf{P}}_t$ be an accurate estimate of the r-dimensional subspace $\mathbf{P}_t$ at time t and $\hat{\mathbf{P}}_{t,\perp}$ be the orthogonal complement of $\hat{\mathbf{P}}_t$. Let $\alpha_t := \hat{\mathbf{P}}_t'\mathbf{L}_t$ be the projection of $\mathbf{L}_t$ onto $\hat{\mathbf{P}}_t$ and $\beta_t := (\hat{\mathbf{P}}_{t,\perp})'\mathbf{L}_t$ be a projection of $\mathbf{L}_t$ onto $\hat{\mathbf{P}}_{t,\perp}$. Then, $\mathbf{M}_t$ can be rewritten as $\mathbf{M}_t = \hat{\mathbf{P}}_t\alpha_t + \hat{\mathbf{P}}_{t,\perp}\beta_t + \mathbf{S}_t$.

REPROCS first projects the measurement vector onto $\hat{\mathbf{P}}_{t,\perp}$ to approximately nullify the low-rank component $\mathbf{L}_t$. Let $\mathbf{y}_t := (\hat{\mathbf{P}}_{t,\perp}')\mathbf{M}_t$, then $\mathbf{y}_t$ can be rewritten as $\mathbf{y}_t = (\hat{\mathbf{P}}_{t,\perp}')\mathbf{S}_t + \beta_t$, and the dimension of the projected data vector reduces to $n-r$. Since $\hat{\mathbf{P}}_t$ is an estimate of $\mathbf{P}_t$, β_t can be interpreted as small noise. Therefore, solving for n-dimensional $\mathbf{S}_t$ from $(n-r)$-dimensional $\mathbf{y_t}$ becomes a traditional sparse recovery problem. Once $\hat{\mathbf{S}}_t$ is recovered, $\mathbf{L}_t$ can be estimated

as $\hat{\mathbf{L}}_t = \mathbf{M}_t - \hat{\mathbf{S}}_t$. Performance of this algorithm highly depends on the correctness of the estimated low-rank subspace and the slowly changing subspace assumption [24]–[27].

B. Compressive Sensing of Matrices

In this section, we briefly review matrix recovery from compressed measurements presented in [31], [32]. Let $\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}$ be an s-sparse matrix with $\| \mathbf{X} \|_0 \leq s$ and $\mathbf{U}_i \in \mathbb{R}^{m_i \times N_i}$ be measurement matrices which satisfy RIP_{2s} property for some $\delta_{2s} \in (0, \sqrt{2}-1)$. Define

$$\mathbf{Y} = \mathbf{U}_1 \mathbf{X} \mathbf{U}_2^\top + \mathbf{E},\ \mathbf{Y} \in \mathbb{R}^{m_1 \times m_2}, \tag{1}$$

where $\mathbf{E} = [\mathbf{e}_1, ...\mathbf{e}_{m_2}]$ denotes the noise matrix. Suppose that the entries of $\mathbf{E}$ are independent and identically distributed (i.i.d.), and assume that $\| \mathbf{e}_i \| \leq \frac{\varepsilon}{\sqrt{m_2}}$ which implies $\| \mathbf{E} \|_F \leq \varepsilon$ for some real nonnegative number ε. $\mathbf{X}$ can be recovered as follows. Let $\mathbf{y}_1, ..., \mathbf{y}_{m2}$ denote the columns of $\mathbf{Y}$ and $\hat{\mathbf{z}}_i \in \mathbb{R}^{N_1}$ be a solution of

$$\hat{\mathbf{z}}_i = \min \| \mathbf{z}_i \|_1, \| \mathbf{y}_i - \mathbf{U}_1 \mathbf{z}_i \|_2 \leq \frac{\varepsilon}{\sqrt{m_2}},\ i \in \{1,\ 2, ...,\ m_2\}. \tag{2}$$

Then each $\hat{\mathbf{z}}_i$ is unique and s-sparse. Let $\mathbf{Z}$ be $\mathbf{X}\mathbf{U}_2^\top$ and $\hat{\mathbf{Z}} \in \mathbb{R}^{N1 \times m_2}$ be the matrix whose columns are $\hat{\mathbf{z}}_1, ..., \hat{\mathbf{z}}_{m_2}$, then $\| \hat{\mathbf{z}}_i - \mathbf{z}_i \|_2 \leq C_2 \frac{\varepsilon}{\sqrt{m_2}}$ implies $\| \hat{\mathbf{Z}} - \mathbf{Z} \|_2 \leq C_2 \varepsilon$ [32]. Let $\mathbf{v}_1, ...\mathbf{v}_N$ be the rows of $\hat{\mathbf{Z}}$. Then $\hat{\mathbf{w}}_j \in \mathbb{R}^{1 \times N2}$, the j^{th} row of $\mathbf{X}$ can be recovered similar to eqn. 2, $\hat{\mathbf{w}}_j^\top = \min \| \mathbf{w}_j^\top \|_1$ subject to $\| \mathbf{v}_j^\top - U_2 \mathbf{w}_j^\top \|_2 \leq C_2 \frac{\varepsilon}{\sqrt{N_1}}$, $j \in \{1,\ 2, ...,\ N_1\}$ yielding $\| \hat{\mathbf{X}} - \mathbf{X} \|_F \leq C_2^2 \varepsilon$.

III. Recursive Low-Rank + Sparse Structure Identification for Matrices

A. Problem Definition & Approach

In this paper, we will state the problem and its solution for a matrix at each time point by considering matrices as 2-way tensors. The developed solutions can be easily extended to higher order tensors.

Let $\mathscr{M}_t \in \mathbb{R}^{N_1 \times N_2}$ be a 2-way tensor representing a network structure at time t. We represent $\mathscr{M}_t$ as:

$$\mathscr{M}_t = \mathscr{L}_t + \mathscr{S}_t \tag{3}$$

where $\mathscr{S}_t$ is the s-sparse tensor and $\mathscr{L}_t$ is the dense component which is low rank along each mode with $rank(\mathbf{L}_t^{(i)}) \ll min(N_i, \prod_{k=1, k \neq i}^{2} N_k)$.

Suppose that we have a sequence of training matrices defined as $\mathscr{M}_{train}$ which do not contain any sparse information and are used for the initial estimate of the subspace in which each mode of $\mathscr{L}_t$ lies in. $\mathscr{M}_{train} \in \mathbb{R}^{N_1 \times N_2 \times t_{train}}$ can be considered as a 3-way tensor and its full Tucker decomposition [33], [34] is

$$\mathscr{M}_{train} = \mathscr{C} \times_1 \mathbf{P}_0^{(1)} \times_2 \mathbf{P}_0^{(2)} \times_3 \mathbf{P}_0^{(3)} \tag{4}$$

where $\mathbf{P}_0^{(1)}$, $\mathbf{P}_0^{(2)}$ and $\mathbf{P}_0^{(3)}$ are the basis matrices along each mode with $\mathbf{P}_0^{(i)} \in \mathbb{R}^{N_i \times N_i}$. Let $\hat{\mathbf{P}}_0^{(i)}$s be the truncated version of $\mathbf{P}_0^{(i)}$s by keeping the columns with the singular values higher than σ_{min}. $\hat{\mathbf{P}}_0^{(i)} \in \mathbb{R}^{N_i \times r_0^{(i)}}$s where $i \in \{1,\ 2\}$ give the initial subspace information for $\mathscr{L}_t$ and $r_0^{(i)}$ is the rank of $\hat{\mathbf{P}}_0^{(i)}$. The goal is to estimate $\mathscr{L}_t$ and $\mathscr{S}_t$ for each $t > t_{train}$ by recursively updating $\mathbf{P}_t^{(i)}$s. The $\mathscr{L}_t$'s are assumed to satisfy a slowly changing low-rank subspace model which will be detailed in section III-B.

Let $\mathbf{P}_t$ be the set of projection matrices which form the basis for the subspaces in which each mode of $\mathscr{L}_t$ lies in $\mathbf{P}_t = \{\mathbf{P}_t^{(1)}, \mathbf{P}_t^{(2)}\}$. Assume $\mathbf{P}_t$ has been accurately predicted using past estimates of $\mathscr{L}_t$ and we have $\hat{\mathbf{P}}_{t-1}^{(i)}$'s with small $\left\| \left(\mathbf{I} - \hat{\mathbf{P}}_{t-1}^{(i)} (\hat{\mathbf{P}}_{t-1}^{(i)})^\top \right) \mathbf{P}_t^{(i)} \right\|_2$. Then $\mathscr{M}_t$ is projected to the space orthogonal to $\hat{\mathbf{P}}_{t-1}$ defined through the projection operators $\boldsymbol{\phi}_t^{(i)} = \mathbf{I} - \hat{\mathbf{P}}_{t-1}^{(i)} (\hat{\mathbf{P}}_{t-1}^{(i)})^\top$ to obtain $\mathscr{Y}_t$ as $\mathscr{Y}_t = \mathscr{M}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)}$, which can be rewritten as:

$$\begin{aligned} \mathscr{Y}_t &= (\mathscr{L}_t + \mathscr{S}_t) \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)}, \\ \mathscr{Y}_t &= \beta_t + \mathscr{S}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)}, \end{aligned} \tag{5}$$

where $\beta_t = \mathscr{L}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)}$. Since $\| \boldsymbol{\phi}_t^{(i)} \mathbf{P}_t^{(i)} \|_2$ is small, the projection of $\mathscr{L}_t$ to $\boldsymbol{\phi}_t^{(i)}$s will yield small $\| \beta_t \|_F$ (see Appendix A). Notice that, although the projection matrices $\boldsymbol{\phi}_t^{(i)}$'s are of size $N_i \times N_i$, they have rank $N_i - rank(\hat{\mathbf{P}}_t^{(i)})$. Therefore, obtaining $\mathscr{S}_t$ from $\mathscr{Y}_t$ can be represented as sparse recovery problem in small noise. Since $\hat{\mathbf{P}}_t^{(i)}$'s are dense and restricted isometry constants (RIC) of measurement matrices ($\boldsymbol{\phi}_t^{(i)}$) are small [26], we can accurately recover $\mathscr{S}_t$ from $\mathscr{Y}_t$. To recover $\mathscr{S}_t$ from $\mathscr{Y}_t$, we apply serial recovery procedure for compressed matrices presented in [35] and rewieved in Section II-B. Once $\hat{\mathscr{S}}_t$ is recovered, $\mathscr{L}_t$ can be estimated as $\hat{\mathscr{L}}_t = \mathscr{M}_t - \hat{\mathscr{S}}_t$.

Algorithm 1 Recursive Projected Compressive Sensing for Matrices

Input: $\mathscr{M}_t$, $\hat{\mathbf{P}}_0^{(i)}s$
Output: $\hat{\mathscr{L}}_t$, $\hat{\mathscr{S}}_t$
for $t > 0$ **do**
  for i=1:2 **do**
    $\boldsymbol{\phi}_t^{(i)} = \mathbf{I} - \hat{\mathbf{P}}_{t-1}^{(i)} (\hat{\mathbf{P}}_{t-1}^{(i)})^\top$
  end for
  $\mathscr{Y}_t = \mathscr{M}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)}$
  Recover $\hat{\mathscr{S}}_t$ from $\mathscr{Y}_t$ by using CSM-S algorithm.
  Estimate $\hat{\mathscr{L}}_t \leftarrow \mathscr{M}_t - \hat{\mathscr{S}}_t$
  if $mod(t - \hat{t}_j + 1, \alpha) = 0$ **then**
    for i=1:2 **do**
      $\mathbf{D}^{(i)} = \left[\hat{\mathbf{L}}_{t_j + (k-1)\alpha}^{(i)} \cdots \hat{\mathbf{L}}_{t_j + k\alpha - 1}^{(i)} \right]$
      $\hat{\mathbf{P}}_{(t)}^{(i)} = deleteDirection(\mathbf{D}, \hat{\mathbf{P}}_{(t-1)}^{(i)})$
      $\hat{\mathbf{P}}_{(t)}^{(i)} = addDirection(\mathbf{D}, \hat{\mathbf{P}}_{(t)}^{(i)})$
    end for
    if $\hat{\mathbf{P}}_{(t)}^{(1)} \neq \hat{\mathbf{P}}_{(t-1)}^{(1)}$ or $\hat{\mathbf{P}}_{(t)}^{(2)} \neq \hat{\mathbf{P}}_{(t-1)}^{(2)}$ **then**
      $j \leftarrow j+1$, $\hat{t}_j \leftarrow t$
      $\hat{\mathbf{P}}_{(j)}^{(1)} \leftarrow \hat{\mathbf{P}}_{(t)}^{(1)}$, $\hat{\mathbf{P}}_{(j)}^{(2)} \leftarrow \hat{\mathbf{P}}_{(t)}^{(2)}$
    end if
  else
    $\hat{\mathbf{P}}_{(t)}^{(1)} \leftarrow \hat{\mathbf{P}}_{(t-1)}^{(1)}$, $\hat{\mathbf{P}}_{(t)}^{(2)} \leftarrow \hat{\mathbf{P}}_{(t-1)}^{(2)}$
  end if
end for

B. Slowly Changing Subspace & Change Points

The following assumptions are made to define slowly changing subspace along each mode of the tensor:

1. Let t_j denote the change points of the low-dimensional subspaces that $\mathbf{L}_t^{(i)}$s are in. Note that the subspaces along each mode can vary independently from the others and as such t_js are the collection of all change points across modes. Assume that for τ large enough, any τ length subsequence of $\mathbf{L}_t^{(i)}$s lies in low-dimensional subspaces, i.e. $max_t rank(\left[\mathbf{L}_{t-\tau+1}^{(i)} ... \mathbf{L}_t^{(i)} \right] \ll \min(\tau, N_i, \prod_{k=1, k \neq i}^{2} N_k)$.

2. $\mathscr{L}_t$ lies in a low dimensional subspace that changes slowly along each mode i.e. $\mathscr{L}_t = \mathscr{A}_t \times_1 \mathbf{P}_t^{(1)} \times_2 \mathbf{P}_t^{(2)}$ with $\mathbf{P}_t^{(i)} = \mathbf{P}_j^{(i)}$ for all $t_j \leq t \leq t_{j+1}$, $j = 1, 2, ...J$ where J is the maximum number of change points. $\mathbf{P}_j^{(i)}$ is an $N_i \times r_j^{(i)}$ basis matrix where $r_j^{(i)} \ll min(N_i, \prod_{k=1, k \neq i}^{2} N_k)$.

3. At the change points, t_j, at least one of the $\mathbf{P}_j^{(i)}$'s

changes as $\mathbf{P}_j^{(i)} = \left[\mathbf{P}_{j-1}^{(i)} \mathbf{P}_{j,add}^{(i)}\right]$, $\mathbf{P}_j^{(i)} = \left[\mathbf{P}_{j-1}^{(i)} \backslash \mathbf{P}_{j,del}^{(i)}\right]$ or $\mathbf{P}_j^{(i)} = \left[(\mathbf{P}_{j-1}^{(i)} \backslash \mathbf{P}_{j,del}^{(i)}), \mathbf{P}_{j,add}^{(i)}\right]$ where $\mathbf{P}_{j,add}^{(i)}$ is a $N_i \times c_{j,add}^{(i)}$ basis matrix with $(\mathbf{P}_{j,add}^{(i)})^\top \mathbf{P}_{j-1}^{(i)} = 0$, i.e., the new directions added to the projection matrix are orthogonal to the previous directions and $\mathbf{P}_{j,del}^{(i)}$ is a $N_i \times c_{j,del}^{(i)}$ matrix of deleted basis columns.

4. There exists constants $c_{max}^{(i)}$ such that $0 \leq c_{j,add}^{(i)} \leq c_{max}^{(i)} < r_0^{(i)}$. $0 \leq \sum_{i=1}^{j}(c_{i,add} - c_{i,del}) \leq c_{dif}^{(i)}$ is required to imply $r_t^{(i)} \leq r_0^{(i)} + c_{dif}^{(i)} := r_{max}^{(i)}$. The number of change points $J \ll \min_i \left((N_i - r_0^{(i)} - c_{dif}^{(i)})/c_{max}^{(i)}\right)$, so $r_{max}^{(i)} + J c_{max}^{(i)} \ll N_i$. Moreover, $(\prod_{k\neq i,k=1}^{2} N_k)(t_{j+1} - t_j) \gg r_0^{(i)} + c_{dif}^{(i)}$ helps to ensure $max_t rank(\left[\mathbf{L}_{t-\tau+1}^{(i)} ... \mathbf{L}_t^{(i)}\right] \ll \min(\tau, N_i, \prod_{k=1,k\neq i}^{2} N_k)$.

5. The projection of $\mathscr{L}_t$ along the new added directions, $\mathscr{A}_{t,add} = \mathscr{L}_t \times_1 \mathbf{P}_{j,add}^{\top,(1)} \times_2 \mathbf{P}_{j,add}^{\top,(2)}$ is initially small, i.e $\max_{t_j \leq t \leq t_{j+\alpha}} \| \mathscr{A}_{t,add} \|_\infty \leq \gamma_{add}$ and $\gamma_{add} \ll min(\| \mathscr{L}_t \|_F, \| \mathscr{S}_t \|_F)$, but can increase gradually.

In order to enable a more efficient online implementation, the low-rank subspaces $\mathbf{P}_t^{(i)}$s are estimated and updated every α samples, where α is selected empirically. Similar to the projection PCA (p-PCA) procedure used in [27], mode-i unfoldings $\hat{\mathbf{L}}_t^{(i)}$'s of the last α $\hat{\mathbf{L}}_t$'s are concatenated as $\mathbf{D}^{(i)} = \left[\hat{\mathbf{L}}_{t_j+(k-1)\alpha}^{(i)} \cdots \hat{\mathbf{L}}_{t_j+k\alpha-1}^{(i)}\right]$ with $k \in \{1, 2, ..., K\}$ where K is the maximum number of length α windows and $\mathbf{D}^{(i)}$s are projected onto subspaces which are orthogonal to $\hat{\mathbf{P}}_{(j-1)}^{(i)}$s as follows: $\mathbf{D}_{proj}^{(i)} = (I - \hat{\mathbf{P}}_{(j-1)}^{(i)}(\hat{\mathbf{P}}_{(j-1)}^{(i)})^\top)\mathbf{D}^{(i)}$. Then PCA is applied to find the subspace which spans $\mathbf{D}_{proj}^{(i)}$. Let $\mathbf{P}_{j,add}^{(i)}$ be the truncated basis that spans this subspace obtained by keeping the eigenvectors with eigenvalues greater than σ_{min}. $\mathbf{P}_{j,add}^{(i)}$ and previous subspace estimate $\hat{\mathbf{P}}_{(j-1)}^{(i)}$ together yields new subspace estimate as: $\hat{\mathbf{P}}_{(j)}^{(i)} = [\hat{\mathbf{P}}_{(j-1)}^{(i)} \mathbf{P}_{j,add}^{(i)}]$. During the update step, some of the existing directions can also be deleted from the projection matrix by finding the ones with eigenvalues lower than σ_{min} (see Algorithms 2 and 3). If there is any added or deleted directions, it means that there is a change point.

C. Computational Complexity

Let the two way tensor $\mathscr{M}_t$ be of size $N \times N$. For the time points which do not require subspace update, computational complexity of the proposed approach is equivalent the complexity of l_1 regularization multiplied by the total number of rows and columns to obtain the sparse component $\mathscr{S}_t$ and is $2N \cdot O(N^3) = 2 \cdot O(N^4)$. However, if we use REPROCS after vectorizing the data, complexity for the same operations become $O((N^2)^3) = O(N^6)$. For the time points which require basis update, there is an additional cost of covariance matrix computation and eigenvalue decomposition. For the proposed approach, covariance matrix computations for the two modes have a complexity of $2 \cdot O((\alpha N) \cdot N^2) = 2 \cdot O(\alpha N^3)$ operations whereas eigenvalue decompositions cost $2 \cdot O(N^3)$. However, REPROCS requires $O(\alpha (N^2)^2) = O(\alpha N^4)$ operations for covariance matrix computation and $O((N^2)^3) = O(N^6)$ operations for eigenvalue decomposition. Therefore, proposed approach is computationally more efficient than applying REPROCS to vectorized data.

IV. Results

A. Simulated Data

The proposed framework is first applied to simulated dynamic weighted and directed networks $\mathbf{X}_t \in \mathbb{R}^{64\times64}$ for $t \in \{1, 2, ..., 60\}$ which are generated 100 times. Initially, the networks contain 2 equal size modules. After $t = 20$, one of the modules is slowly divided into two smaller modules of size 16 nodes each. Intra-cluster edge values were selected from $N(0.4,\ 0.1)$ and truncated to the interval $[0, 1]$ while the inter-cluster edge values were selected from $N(0.1,\ 0.1)$. Moreover, these networks were corrupted by a sparse noise matrix $\mathbf{E}_t$ which is 5% sparse and $e_{i,j} \sim beta(3, 2)$. Proposed algorithm is applied with $\alpha = 4$ and $\sigma_{min} = 0.13$. The same algorithm is also applied without sparse recovery step similar to standard PCA. Mean squared error for the recovered low-rank components were computed for both algorithms at each time point (Fig. 1). It can be seen that the proposed approach provides better tracking and better approximation for the low-rank component. It can also be seen that, MSE computed for the proposed algorithm increases after the change point and then decreases after subspace update.

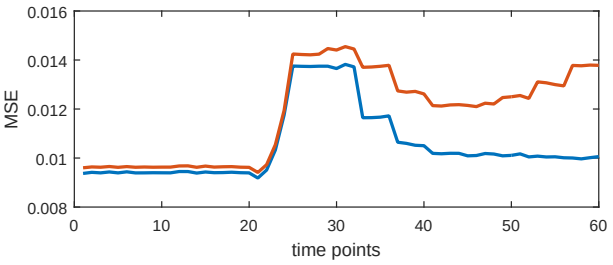

Fig. 1: MSE computed for low-rank components obtained by the proposed approach (blue) and simple subspace tracking algorithm (red).

B. Reality Mining Dataset

This dataset was collected at MIT Media Laboratory and contains bluetooth interaction of 94 mobile phone users from September 2004 to June 2005 [36]. The participants are categorized as follows: 36 are graduate students who are not in their first year (nodes 1-36), 15 are first year graduate students (nodes 37-51), 6 are staff members (nodes 52-57), 3 are undergraduate students (58,63,66), 6 are freshman (nodes 59-62, 64, 65), one is a professor (node 94) in Media Lab and 27 are students from Sloan Business School (nodes 67-93).

Dynamic 2-way tensor $\mathbf{X}_t \in \mathbb{R}^{91\times91}$ is constructed at each time point $t \in \{1, 2, ..., 46\}$. The first 6 time points which correspond to the time before the Fall term were discarded due to absence of enough data and the networks for the next 4 time points were used for training. The proposed algorithm was applied to the dynamic tensor $\mathbf{X}_t$ with $\alpha = 3$ and $\sigma_{min} = 0.1$ where $t \in \{11, 12, ..., 46\}$. As seen in Fig. 2, the detected change points follow the time points corresponding to the end of the Fall term, the start of the Spring term and the end of the Spring term respectively. Moreover, average of the obtained low rank components were computed for each time period (see Fig. 3). Two dominant clusters which correspond to graduate students, and students from Sloan Business School were observed in the low-rank matrices. The nodes corresponding to Sloan Business School are not very active in the low-rank matrices after the first change point. It can also be seen that most of the nodes are inactive during winter and summer breaks.

V. Conclusions

In this paper, we introduced a new recursive projection based tensor subspace tracking approach to identify changes in dynamic networks. Unlike other tensor subspace learning approaches, we adapted REPROCS to track tensor subspace and decompose the tensor into low-rank and sparse components. The proposed framework is evaluated by applying it to simulated and real networks. Future work will consider

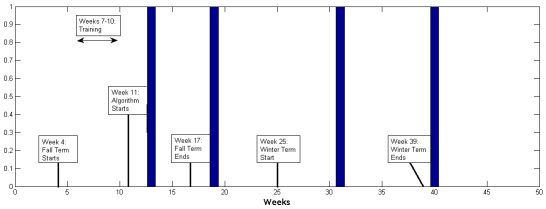

Fig. 2: Detected change points (blue stripes) for Reality Mining Dataset.

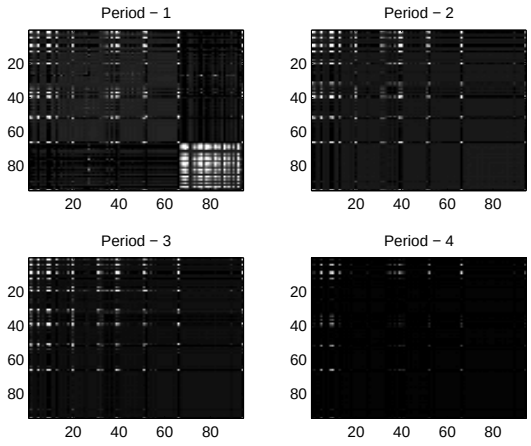

Fig. 3: Average of the low-rank components within each detected time interval on Reality Mining Dataset.

extending the proposed approach to higher order tensors such as dynamic functional connectivity networks of the brain formed across subjects.

VI. Appendix

A. β_t is Small

In this section, we will prove that β_t is small and that eqn (5) can be treated as a sparse recovery in noise problem. Define the subspace estimation error for ith mode as $\text{SE}(\mathbf{P}^{(i)}, \hat{\mathbf{P}}^{(i)}) := \| (I - \hat{\mathbf{P}}^{(i)}\hat{\mathbf{P}}^{\top,(i)})\mathbf{P}^{(i)} \|_F = \varepsilon_i$, where $\mathbf{P}^{(i)}$ and $\hat{\mathbf{P}}^{(i)}$ are true and estimated basis matrices of the ith mode, respectively. $\mathscr{L}_t$ can be written as

$$\begin{aligned}\mathscr{L}_t = \quad & \mathscr{A}_1 \times_1 \mathbf{P}^{(1)}_{j-1} \times_2 \mathbf{P}^{(2)}_{j-1} + \mathscr{A}_2 \times_1 \mathbf{P}^{(1)}_{j,add} \times_2 \mathbf{P}^{(2)}_{j-1} \\ & + \mathscr{A}_3 \times_1 \mathbf{P}^{(1)}_{j-1} \times_2 \mathbf{P}^{(2)}_{j,add} + \mathscr{A}_4 \times_1 \mathbf{P}^{(1)}_{j,add} \times_2 \mathbf{P}^{(2)}_{j,add},\end{aligned} \tag{6}$$

where we redefine the old and new parts of the projection as

$$\begin{aligned}\mathscr{A}_{t,*} = \mathscr{A}_1 = \quad & \mathscr{L}_t \times_1 \mathbf{P}^{\top,(1)}_{j-1} \times_2 \mathbf{P}^{\top,(2)}_{j-1} \\ \mathscr{A}_2 = \quad & \mathscr{L}_t \times_1 \mathbf{P}^{\top,(1)}_{j,add} \times_2 \mathbf{P}^{\top,(2)}_{j-1} \\ \mathscr{A}_3 = \quad & \mathscr{L}_t \times_1 \mathbf{P}^{\top,(1)}_{j-1} \times_2 \mathbf{P}^{\top,(2)}_{j,add} \\ \mathscr{A}_{t,add} = \mathscr{A}_4 = \quad & \mathscr{L}_t \times_1 \mathbf{P}^{\top,(1)}_{j,add} \times_2 \mathbf{P}^{\top,(2)}_{j,add}.\end{aligned} \tag{7}$$

Assumptions:

1) Subspace estimation error $\varepsilon_i = \| (I - \hat{\mathbf{P}}^{(i)}, \hat{\mathbf{P}}^{\top,(i)})\mathbf{P}^{(i)} \|_F \le r_0^{(i)}\zeta$ for $\zeta \ll 1$.
2) Let $\mathbf{l}_k^{(i)}$ be the kth column of $\mathbf{L}_{t,(i)}$ and assume that $\| \mathbf{l}_k^{(i)} \|_F \le \gamma_*^{(i)}$, $\gamma_*^{(i)} \le \frac{1}{\sqrt{\zeta r_J^{(i)}}}$ and $\gamma_{add}^{(i)} << \gamma_*^{(i)}$
3) Assume that $\| \mathscr{L}_t \|_F \le \gamma_*$ where $\gamma_* = min_{i\in\{1,2\}} \left(\gamma_*^{(i)}\sqrt{\prod_{k=1,k\neq i}^2 N_k}\right)$.
4) Assume that $\gamma_{add} = min_{i\in\{1,2\}} \left(\gamma_{add}^{(i)}\sqrt{\prod_{k=1,k\neq i}^2 N_k}\right)$ and $\gamma_{add} \ll \gamma_*$.

The norm of β_t is:

$$\begin{aligned}\| \beta_t \|_F = \quad & \| \mathscr{L}_t \times_1 \boldsymbol{\phi}_t^{(1)} \times_2 \boldsymbol{\phi}_t^{(2)} \|_F \\ \le \quad & \varepsilon_1\varepsilon_2 ||\mathscr{A}_{t,*}||_F + \varepsilon_2 ||\mathscr{A}_2||_F + \varepsilon_1 ||\mathscr{A}_3||_F + ||\mathscr{A}_{t,add}||_F\end{aligned} \tag{8}$$

and,

$$\begin{aligned}\| \mathscr{A}_{t,2} \|_F = \quad & \| \mathscr{L}_t \times_1 \mathbf{P}^{\top,(1)}_{j,add} \times_2 \mathbf{P}^{\top,(2)}_{j-1} \|_F \\ = \quad & \| \mathbf{P}^{\top,(1)}_{j,add} \mathbf{L}_{t,(1)} \mathbf{P}^{(2)}_{j-1} \|_F \\ \le \quad & \| \mathbf{P}^{\top,(1)}_{j,add} \|_F \cdot \| \mathbf{L}_{t,(1)} \|_F \cdot \| \mathbf{P}^{(2)}_{j-1} \|_F \\ = \quad & \gamma_* \sqrt{r^{(1)}_{j,add} r^{(2)}_{j-1}}.\end{aligned} \tag{9}$$

Similarly, $\| \mathscr{A}_{t,3} \|_F \le \gamma_* \sqrt{r^{(1)}_{j-1} r^{(2)}_{j,add}}$.

Let $\bar{N} = max_i(N_i)$, $\bar{r}_J = max_i(r_J^{(i)})$ and $\bar{\gamma}_* = max_i(\gamma_*^{(i)}) \cdot \bar{N}$. Assume that $\bar{\gamma}_{add} = max_k(\gamma_{add}^{(k)})$ and $\bar{\gamma}_{add} \ll \bar{\gamma}_*$.
Therefore;

$$\begin{aligned}\| \beta_t \|_F \le \quad & \bar{N} \cdot \zeta^{3/2} \cdot (\bar{r}_{j-1}) + 2 \cdot \bar{N} \cdot \zeta^{1/2} \cdot \sqrt{\bar{r}_{j-1}\bar{r}_{j,add}} + \bar{N} \cdot \bar{\gamma}_{add}. \\ \| \beta_t \|_F \le \quad & \bar{N} \cdot \zeta^{1/2} \cdot \bar{\gamma}_*^{(-2)} + 2 \cdot \bar{N} \cdot \bar{\gamma}_*^{(-1)} \cdot (\bar{r}_{j,new})^{1/2} + \bar{N} \cdot \bar{\gamma}_{add}.\end{aligned} \tag{10}$$

Since ζ is small and $\bar{\gamma}_*$ is large, the last term is dominant in the upper bound. Thus, β_t can be considered as noise by the slow subspace change assumption $\| \gamma_{add} \|_F \ll \| \mathscr{S}_t \|_F$.

B. Algorithms

Algorithm 2 deleteDirection

1: Input: **D**: data, **P**: input basis matrix
2: Output: **Q**: output basis matrix
3: $\lambda = \frac{1}{m} diag((\mathbf{P}'\mathbf{D}(\mathbf{P}'\mathbf{D})^\top))$ where m is the number of colums of D.
4: i= find($diag(\lambda) < \sigma_{min}$)
5: $\mathbf{Q} = [\mathbf{P} \backslash \mathbf{P}(i,:)]$

Algorithm 3 addDirection

1: Input: **D**: data, **P**: input basis matrix
2: Output: **Q**: output basis matrix
3: Projection: **Projection:** compute $D_{proj} \leftarrow (I - PP')D$
4: PCA: compute $\frac{1}{m} D_{proj} D'_{proj} = U\lambda U'$ where m is the number of columns in D.
5: i= find($diag(\lambda) > \sigma_{min}$)
6: $\mathbf{Q} = [\mathbf{P}\ \mathbf{U}(i,:)]$

References

[1] F.De La Torre and M. J.Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 117–142, 2003.
[2] S.Roweis, "Em algorithms for pca and spca," *Advances in neural information processing systems*, pp. 626–632, 1998.
[3] E. J.Candès *et al.*, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.
[4] V.Chandrasekaran *et al.*, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.
[5] T.Zhang and G.Lerman, "A novel m-estimator for robust pca," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 749–808, 2014.

[6] M.McCoy *et al.*, "Two proposals for robust pca using semidefinite programming," *Electronic Journal of Statistics*, vol. 5, pp. 1123–1160, 2011.

[7] H.Xu *et al.*, "Robust pca via outlier pursuit," in *Advances in Neural Information Processing Systems*, 2010, pp. 2496–2504.

[8] V.Chandrasekaran *et al.*, "The convex geometry of linear inverse problems," *Foundations of Computational mathematics*, vol. 12, no. 6, pp. 805–849, 2012.

[9] A.Ganesh *et al.*, "Principal component pursuit with reduced linear measurements," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 1281–1285.

[10] G.Mateos and G. B.Giannakis, "Robust pca as bilinear decomposition with outlier-sparsity regularization," *Signal Processing, IEEE Transactions on*, vol. 60, no. 10, pp. 5176–5190, 2012.

[11] D.Hsu *et al.*, "Robust matrix decomposition with sparse corruptions," *Information Theory, IEEE Transactions on*, vol. 57, no. 11, pp. 7221–7234, 2011.

[12] J.Wright *et al.*, "Compressive principal component pursuit," *Information and Inference*, vol. 2, no. 1, pp. 32–68, 2013.

[13] M.Tao and X.Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 57–81, 2011.

[14] M. B.McCoy and J. A.Tropp, "Sharp recovery bounds for convex deconvolution, with applications," Tech. Rep., 2012.

[15] M.Mardani *et al.*, "Decentralized sparsity-regularized rank minimization: Algorithms and applications," *Signal Processing, IEEE Transactions on*, vol. 61, no. 21, pp. 5374–5388, 2013.

[16] M.Mardani and G.Giannakis, "Robust network traffic estimation via sparsity and low rank," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4529–4533.

[17] M.Mardani *et al.*, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *Information Theory, IEEE Transactions on*, vol. 59, no. 8, pp. 5186–5205, 2013.

[18] E. J.Candès and B.Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[19] M.Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Computer VisionECCV 2002*. Springer, 2002, pp. 707–720.

[20] D.Skocaj and A.Leonardis, "Weighted and robust incremental method for subspace learning," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1494–1501.

[21] Y.Li *et al.*, "An integrated algorithm of incremental and robust pca," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 1. IEEE, 2003, pp. I–245.

[22] J.He *et al.*, "Online robust subspace tracking from partial information," *arXiv preprint arXiv:1109.3827*, 2011.

[23] Y.Chi *et al.*, "Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations," *Signal Processing, IEEE Transactions on*, vol. 61, no. 23, pp. 5947–5959, 2013.

[24] C.Qiu and N.Vaswani, "Reprocs: A missing link between recursive robust pca and recursive sparse recovery in large but correlated noise," *arXiv preprint arXiv:1106.3286*, 2011.

[25] C.Qiu *et al.*, "Recursive robust pca or recursive sparse recovery in large but structured noise," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5954–5958.

[26] C.Qiu and N.Vaswani, "Recursive sparse recovery in large but structured noisepart 2," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 864–868.

[27] H.Guo *et al.*, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," 2014.

[28] J.Sun *et al.*, "Beyond streams and graphs: dynamic tensor analysis," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 374–383.

[29] J.Sun *et al.*, "Window-based tensor analysis on high-dimensional and multi-aspect streams." in *ICDM*, vol. 6, 2006, pp. 1076–1080.

[30] M.Mardani *et al.*, "Subspace learning and imputation for streaming big data matrices and tensors," *Signal Processing, IEEE Transactions on*, vol. 63, no. 10, pp. 2663–2677, 2015.

[31] S.Friedland *et al.*, "Compressive sensing of sparse tensors," *Image Processing, IEEE Transactions on*, vol. 23, no. 10, pp. 4438–4447, 2014.

[32] S.Friedland *et al.*, "Two algorithms for compressed sensing of sparse tensors," *arXiv preprint arXiv:1404.1506*, 2014.

[33] L. R.Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[34] L.De Lathauwer *et al.*, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[35] S.Friedland *et al.*, "Compressive sensing of sparse tensors," *arXiv preprint arXiv:1305.5777*, 2013.

[36] N.Eagle and A.Pentland, "Reality mining: sensing complex social systems," *Personal and ubiquitous computing*, vol. 10, no. 4, pp. 255–268, 2006.