HIGH DIMENSIONAL DECOMPOSITION OF COHERENT/STRUCTURED MATRICES VIA SEQUENTIAL COLUMN/ROW SAMPLING

Mostafa Rahmani and George Atia

Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA
mostafa@knights.ucf.edu, george.atia@ucf.edu

ABSTRACT

This paper focuses on the low rank plus sparse matrix decomposition problem in big data settings. Conventional algorithms solve high-dimensional optimization problems that scale with the data dimension, which limits their scalability. In addition, existing randomized approaches mostly rely on blind random sampling. In this paper, the drawbacks of random sampling from coherent/structured data matrices are analyzed showing that random sampling cannot provide efficient descriptive sketches of coherent data. In addition, a column/row subspace pursuit algorithm which recovers the low rank component via a small set of informative columns/rows of the data is proposed. The obtained column and row spaces are updated in each iteration to converge to the column and row spaces of the low rank matrix. The informative columns are located using the information embedded in the row space while the informative rows are identified using the information embedded in the column space.

***Index Terms*—** Robust PCA, Randomized Method, Subspace Recovery, Low Rank Matrix, Matrix Decomposition

1. INTRODUCTION

There are many important applications in which the given data matrix $\mathbf{D} \in \mathbb{R}^{N_1 \times N_2}$ can be naturally modeled as

$$\mathbf{D} = \mathbf{L} + \mathbf{S}, \tag{1}$$

where $\mathbf{L}$ is a low rank (LR) matrix and $\mathbf{S}$ a sparse matrix with arbitrary unknown support [1–7]. The following convex program was proposed in [1, 2] to directly decompose the matrix $\mathbf{D}$ into its LR and sparse components

$$\min_{\dot{\mathbf{L}},\dot{\mathbf{S}}} \lambda\|\dot{\mathbf{S}}\|_1 + \|\dot{\mathbf{L}}\|_* \quad \text{s. t.} \quad \dot{\mathbf{L}} + \dot{\mathbf{S}} = \mathbf{D}\,, \tag{2}$$

where $\|.\|_1$ is the ℓ_1-norm, $\|.\|_*$ the nuclear norm and λ a parameter that determines the trade-off between the sparse and LR components [1]. The optimal point of (2) was shown to yield exact decomposition if the columns and rows subspaces of $\mathbf{L}$ are sufficiently incoherent with the standard basis and the non-zero elements of $\mathbf{S}$ are sufficiently diffused [1]. The decomposition algorithm (2) requires saving $N_1 \times N_2$ dimensional matrices in the working memory and the complexity of solving (2) is $\mathcal{O}(rN_1N_2)$ per iteration (r is the rank of $\mathbf{L}$), which is prohibitive for high dimensional data [8, 9].

1.1. Randomized approaches

An effective idea to develop scalable decomposition algorithms is to exploit the low dimensional structure of the LR matrix [10–15]. The idea is to sample a set of columns of $\mathbf{D}$ whose LR component can span the column space (CS) of $\mathbf{L}$. These sampled columns are decomposed using (2) to learn the CS of $\mathbf{L}$. Similarly, the row space (RS) of $\mathbf{L}$ is obtained by decomposing a set of randomly sampled rows. Finally, the LR matrix is recovered using the learned CS and RS. Thus, instead of decomposing the full scale data, one decomposes small sketches constructed from subsets of the data columns and rows [12, 14]. To the best of our knowledge, all current randomized decomposition algorithms rely on uniform random sampling for column/row sampling. However, uniform sampling is not efficient when the CS/RS of $\mathbf{L}$ are coherent with the standard basis and we might not be able to evade the high dimensionality of the data with uniform random sampling. This motivates the work of this paper which focuses on matrix decomposition using adaptive sampling from structured and coherent data.

1.2. Definitions

The row space of a matrix $\mathbf{L}$ with rank r is said to be incoherent with parameter μ_v, if $\max_i \|\mathbf{V}^T\mathbf{e}_i\|_2^2 \leq \frac{r\mu_v}{N_2}$, where $\mathbf{V}$ is an orthonormal basis for the RS of $\mathbf{L}$. Similarly, the column space of $\mathbf{L}$ is said to be incoherent with parameter μ_u, if $\max_i \|\mathbf{U}^T\mathbf{e}_i\|_2^2 \leq \frac{r\mu_u}{N_1}$, where $\mathbf{U}$ is an orthonormal basis for the CS of $\mathbf{L}$ [16, 17].

In this paper, it is assumed that the support of $\mathbf{S}$ follows the Bernoulli model with parameter ρ, i.e., each element of $\mathbf{S}$ is equal to zero independently with probability $1 - \rho$.

This work was supported by NSF CAREER Award CCF-1552497 and NSF Grant CCF-1320547.

2. NON-UNIFORM DATA DISTRIBUTION

In this section, we provide a theoretical study of the shortcomings of random sampling with coherent data matrices. First, we show that the coherency of the RS/CS increases when the distributions of the columns/rows of $\mathbf{L}$ are less uniform within the CS/RS. Subsequently, it is shown that random sampling does not yield efficient descriptive data sketches if the CS/RS of $\mathbf{L}$ are highly coherent. The proofs of the presented theoretical results are available in [14].

2.1. Studying the coherency of L

In this section, we present results establishing that the coherency of the RS of $\mathbf{L}$ with the standard basis increases if the distribution of the columns of $\mathbf{L}$ is less uniform in the CS of $\mathbf{L}$. We assume that the columns follow a subspace clustering structure [18] as per the following assumption.

Assumption 1 *The matrix* $\mathbf{L}$ *can be represented as* $\mathbf{L} = [\mathbf{U}_1\mathbf{Q}_1 \ ... \ \mathbf{U}_n\mathbf{Q}_n]$. *The CS of* $\{\mathbf{U}_i \in \mathbb{R}^{N_1 \times r/n}\}_{i=1}^{n}$ *are random* r/n*-dimensional subspaces in* $\mathbb{R}^{N_1}$. *The RS of* $\{\mathbf{Q}_i \in \mathbb{R}^{r/n \times n_i}\}_{i=1}^{n}$ *are random* r/n*-dimensional subspaces in* $\{\mathbb{R}^{n_i}\}_{i=1}^{n}$, *respectively,* $\sum_{i=1}^{n} n_i = N_2$, *and* $\min_i n_i \gg r/n$.

The following lemma establishes a lower bound for the RS coherency of $\mathbf{L}$.

Lemma 1 *If* $\mathbf{L}$ *follows Assumption 1, the rank of* $\mathbf{L}$ *is equal to* r, $r/n \geq 18 \log \max_i n_i$ *and* $n_i \geq 96 \frac{r}{n} \log n_i, 1 \leq i \leq n$, *then*

$$\mathbb{P}\left[\max_i \|\mathbf{V}^T \mathbf{e}_i\|_2^2 < \frac{0.5\, r}{N_2}\left(\frac{1}{n}\frac{N_2}{\min_i n_i}\right)\right] \leq 2\sum_{i=1}^{n} n_i^{-5}\,. \quad (3)$$

According to Lemma 1, the RS coherency of $\mathbf{L}$ is linear in $\frac{N_2}{\min_i n_i}$. The factor $\frac{N_2}{\min_i n_i}$ is an excellent measure of the columns distribution. For instance, suppose $n = 2$ and $r = 2$, i.e., the columns lie in a union of two one-dimensional subspaces. A large value of $\frac{N_2}{\min_i n_i}$ implies that most of the data is aligned along a given direction, i.e., the distribution of the columns is highly non-uniform. Accordingly, if the distribution of the columns/rows of $\mathbf{L}$ is less uniform, the coherency of the RS/CS of $\mathbf{L}$ increases.

2.2. Sampling from coherent data

In this subsection, we highlight two drawbacks of random sampling from coherent data.

I. Random sampling cannot capture the CS/RS efficiently: When the distribution of the columns/rows of $\mathbf{L}$ is non-uniform (the coherency of RS/CS is high), we cannot capture the CS/RS using a small set of randomly sampled columns/rows. To show this fact, we provide a numerical example and confirm our analysis by Lemma 2.

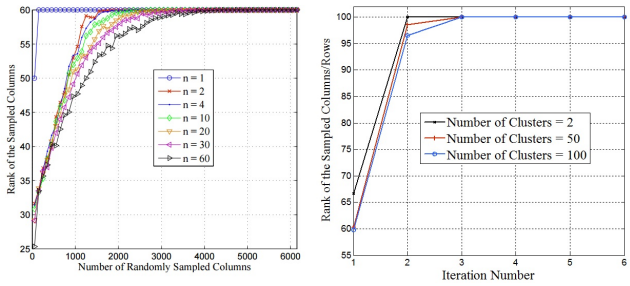

Fig. 1. (Left) The rank of a set of uniformly sampled columns for different number of clusters. (Right) The rank of the low rank component of sampled columns after each iteration by Algorithm 1.

Numerical example: Suppose $\mathbf{L}$ follows Assumption 1, $N_1 = 200$, $N_2 = 6150$, $r = 60$, $\{n_i\}_{i=1}^{n/2} = \frac{5r}{n}$, and $\{n_i\}_{i=n/2+1}^{n} = \frac{200r}{n}$. Thus, the distribution of the columns of $\mathbf{L}$ is strongly structured and highly non-uniform ($\frac{N_2}{\min_i n_i} = 1230$ when $n = 60$). The left plot of Fig. 1 illustrates the rank of the randomly sampled columns versus the number of sampled columns for different number of clusters n. When $n = 60$, it turns out that we need to sample more than half of the columns to span the CS. As such, we cannot evade high-dimensionality with uniform random column/row sampling.

The following lemma confirms our analysis. It shows that the sufficient number of randomly sampled columns scales linearly with the RS coherency, which can be quite high if the columns are distributed non-uniformly (c.f. Lemma 1).

Lemma 2 *Suppose* m_1 *columns are sampled uniformly at random with replacement from the matrix* $\mathbf{L}$ *with rank* r. *If* $m_1 \geq 10\mu_v r \log \frac{2r}{\delta}$, *where* μ_v *is the row space coherency defined in Sec. 1.2, then the selected columns of the matrix* $\mathbf{L}$ *span the CS of* $\mathbf{L}$ *with probability at least* $(1 - \delta)$.

II. Random sampling requires too many data points to yield correct decomposition: Suppose $\mathbf{D}_c$ is the matrix of sampled columns and $\mathbf{L}_c$ its LR component. The following lemma shows that the sufficient number of randomly sampled columns to ensure correct decomposition of $\mathbf{D}_c$ is linear in the coherency parameter even if $\mathbf{L}_c$ has the CS of $\mathbf{L}$.

Lemma 3 *Suppose the CS of* $\mathbf{L}$ *is a random* r*-dimensional subspace,* $\mathbf{L}_c$ *has the same CS of* $\mathbf{L}$ *and the support set of* $\mathbf{S}$ *follows the Bernoulli model with parameter* ρ. *In addition, assume that the columns of* $\mathbf{D}_c$ *were sampled uniformly at random. If* $m_1 \geq \frac{r}{c_4}\mu'(\log N_1)^2$ *and* $\rho \leq \rho_s$, *then*

$$\min_{\dot{\mathbf{L}}_c, \dot{\mathbf{S}}_c} \lambda\|\dot{\mathbf{S}}_c\|_1 + \|\dot{\mathbf{L}}_c\|_* \quad \textit{subject to} \quad \dot{\mathbf{L}}_c + \dot{\mathbf{S}}_c = \mathbf{D}_c\,, \quad (4)$$

yields the exact decomposition with probability at least $1 - c_1 N_1^{-3}$, *where* $\mu' = \max\left(\frac{c_2 \max(r, \log N_1)}{r}, 6\mu_v, \mu_v(c_3 \log N_1)^2\right)$ *and* $\{c_i\}_{i=1}^{4}$ *are constant numbers.*

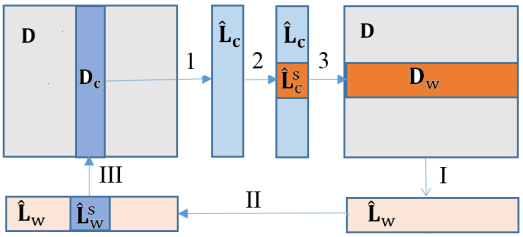

Fig. 2. Visualization of Algorithm 1. We run few cycles of the algorithm and stop when the rank of the LR component does not change over T consecutive steps. One cycle of the algorithm starts from the point marked "I" and proceeds as follows. **I**: Matrix $\mathbf{D}_w$ is decomposed and $\hat{\mathbf{L}}_w$ is the obtained LR component of $\mathbf{D}_w$. **II**: Algorithm 2 is applied to $\hat{\mathbf{L}}_w$ to select the informative columns of $\hat{\mathbf{L}}_w$. $\hat{\mathbf{L}}_w^s$ is the matrix of columns selected from $\hat{\mathbf{L}}_w$. **III**: Matrix $\mathbf{D}_c$ is formed from the columns of $\mathbf{D}$ that correspond to the columns of $\hat{\mathbf{L}}_w^s$. **1**: Matrix $\mathbf{D}_c$ is decomposed and $\hat{\mathbf{L}}_c$ is the obtained LR component of $\mathbf{D}_c$. **2**: Algorithm 2 is applied to $\hat{\mathbf{L}}_c^T$ to select the informative rows of $\hat{\mathbf{L}}_c$. $\hat{\mathbf{L}}_c^s$ is the matrix of rows selected from $\hat{\mathbf{L}}_c$. **3**: Matrix $\mathbf{D}_w$ is formed as the rows of $\mathbf{D}$ corresponding to the rows used to form $\hat{\mathbf{L}}_c^s$.

Algorithm 1 Column/Row Subspace Pursuit Algorithm

1. Initialization

Form $\mathbf{D}_w \in \mathbb{R}^{C_r\hat{r}\times N_2}$ by randomly choosing $C_r\hat{r}$ rows of $\mathbf{D}$. Initialize $k=1$ and set T equal to an integer greater than 1.

2. While $k > 0$

2.1 Sample the most informative columns

2.1.1 Obtain $\hat{\mathbf{L}}_w$ via (2) as the LR component of $\mathbf{D}_w$.

2.1.2 Apply Algorithm 2 to $\hat{\mathbf{L}}_w$ with $C = C_r$.

2.1.3 Form the matrix $\mathbf{D}_c$ from the columns of $\mathbf{D}$ corresponding to the sampled columns of $\hat{\mathbf{L}}_w$.

2.2 Sample the most informative rows

2.2.1 Obtain $\hat{\mathbf{L}}_c$ via (2) as the LR component of $\mathbf{D}_c$.

2.2.2 Apply Algorithm 2 to $\hat{\mathbf{L}}_c^T$ with $C = C_r$.

2.2.3 Form the matrix $\mathbf{D}_w$ from the rows of $\mathbf{D}$ corresponding to the sampled rows of $\hat{\mathbf{L}}_c$.

2.3 If the dimension of the RS of $\hat{\mathbf{L}}_w$ does not increase in T consecutive iterations, set $k=0$ to stop the algorithm.

2. End While

Output: The matrices $\hat{\mathbf{L}}_c$ and $\hat{\mathbf{L}}_w$.

3. COLUMN-ROW SUBSPACE PURSUIT SAMPLING ALGORITHM

In this algorithm, we address the worst case scenario in which the distribution of both columns and rows of $\mathbf{L}$ can be highly non-uniform. Thus, a small set of randomly sampled columns/rows does not span the CS/RS with high probability. It is assumed that we know an upper bound $\hat{r}$ on the rank of $\mathbf{L}$. Such knowledge is often available as side information depending on the particular application. For instance, facial images under varying illumination and facial expressions are known to lie on a special low-dimensional subspace [19]. In the proposed method, we define $\mathbf{D}_c$ and $\mathbf{D}_w$ as the matrices of sampled columns and sampled rows, respectively. The matrices $(\mathbf{L}_c, \mathbf{S}_c)$ and $(\mathbf{L}_w, \mathbf{S}_w)$ are the LR and sparse components of $\mathbf{D}_c$ and $\mathbf{D}_w$, respectively. In addition, matrices $\hat{\mathbf{L}}_c$ and $\hat{\mathbf{L}}_w$ are the obtained LR component of $\mathbf{D}_c$ and $\mathbf{D}_w$ via (2), respectively.

Before presenting the proposed algorithm (Algorithm 1), we explain Algorithm 2, which is used as a subroutine in the proposed algorithm. Algorithm 2 is a column sampling algorithm which locates the informative columns of a given LR rank matrix. In each iteration, *the algorithm finds a set of the columns of the given matrix that span its CS*. Since it repeats the sampling process C times (without replacement), in the end Cr non-repeated columns are sampled. C is chosen sufficiently large so that the sampled columns form a LR matrix. We refer the reader to [20, 21] and references therein for further information on efficient methods for column sampling from LR matrices.

The table of Algorithm 1, Fig. 2 and its caption provide the details of the proposed sampling approach and the definitions of the used matrices. We start the cycle from the position marked "I" in Fig. 2 with $\mathbf{D}_w$ formed using $C_r\hat{r}$ randomly sampled rows. The constant C_r is chosen large enough such that $\mathbf{L}_w$ is a LR matrix. For ease of exposition, assume that $\hat{\mathbf{L}}_w = \mathbf{L}_w$ and $\hat{\mathbf{L}}_c = \mathbf{L}_c$, i.e., $\mathbf{D}_w$ and $\mathbf{D}_c$ are decomposed correctly. The matrix $\hat{\mathbf{L}}_w^s$ is the informative columns of $\hat{\mathbf{L}}_w$. Thus, the rank of $\hat{\mathbf{L}}_w^s$ is equal to the rank of $\hat{\mathbf{L}}_w$. Since $\hat{\mathbf{L}}_w = \mathbf{L}_w$, $\hat{\mathbf{L}}_w^s$ is a subset of the rows of $\mathbf{L}_c$. If the rows of $\mathbf{L}$ exhibit a clustering structure (or their distribution is not uniform), it is likely that $\text{rank}(\hat{\mathbf{L}}_w^s) < \text{rank}(\mathbf{L}_c)$. Thus, $\text{rank}(\mathbf{L}_w) < \text{rank}(\mathbf{L}_c)$. We continue one cycle of the algorithm by going through steps 1, 2 and 3 of Fig. 2 to update $\mathbf{D}_w$. Using a similar argument, we see that the rank of an updated $\mathbf{L}_w$ will be greater than the rank of $\mathbf{L}_c$. Thus, if we run more cycles of the algorithm – each time updating $\mathbf{D}_w$ and $\mathbf{D}_c$ – the rank of $\mathbf{L}_w$ and $\mathbf{L}_c$ will increase. As detailed in the table of Algorithm 1, we stop if the dimension of the span of the obtained LR component does not change in T consecutive iterations. While there is no guarantee that the rank of $\mathbf{L}_w$ will converge to r (it can converge to a value smaller than r), our investigations have shown that Algorithm 1 performs quite well and the RS of $\mathbf{L}_w$ converges to the RS of $\mathbf{L}$ in few steps. In addition, the proposed method can be used independently as a robust column/feature sampling algorithm [22, 23].

3.1. Recovering the LR matrix

Once the sampling algorithm converges, $\mathbf{L}$ can be easily recovered using the LR components of the matrices $\mathbf{D}_c$ and $\mathbf{D}_w$. Define $\mathbf{H} \in \mathbb{R}^{m_2\times m_1}$ as the matrix formed from the intersection of $\hat{\mathbf{L}}_w$ and $\hat{\mathbf{L}}_c$, where m_1 is the number of columns of $\hat{\mathbf{L}}_c$ and m_2 the number of rows of $\hat{\mathbf{L}}_w$. In addition, define $\mathbf{U}_h \in \mathbb{R}^{m_2\times\dot{r}}$, $\mathbf{V}_h \in \mathbb{R}^{m_1\times\dot{r}}$, $\mathbf{\Sigma}_h \in \mathbb{R}^{\dot{r}\times\dot{r}}$ as the matrix of left singular vectors, the matrix of right singular vectors,

and the diagonal matrix of singular values of $\mathbf{H}$, respectively, where $\dot{r}$ is its rank. We make use of the generalized Nyström method to form the LR matrix as follows [24]

$$\hat{\mathbf{L}} = \hat{\mathbf{L}}_c \mathbf{V}_h \mathbf{\Sigma}_h^{+} \mathbf{U}_h^T \hat{\mathbf{L}}_w \ . \tag{5}$$

In addition, The sparse component can be obtained as $\hat{\mathbf{S}} = \mathbf{D} - \hat{\mathbf{L}}$.

Remark 1 *The number of rows/columns of $\mathbf{D}_w$/$\mathbf{D}_c$ is of order $\mathcal{O}(r)$. Thus, the complexity of Algorithm 1 is roughly $\mathcal{O}(\max(N_1, N_2) r^2 T_1 T_2)$, where T_1 is the number of iterations to decompose $\mathbf{D}_w$/$\mathbf{D}_c$ and T_2 the number of iterations of Algorithm 1. According to our investigations, few iterations (less than 4) of Algorithm 1 are usually sufficient. Thus, the overall complexity is of order $\mathcal{O}(\max(N_1, N_2) r^2 T_1)$.*

Algorithm 2 Informative Column Sampling from LR Matrices

Input: Matrix $\mathbf{A}$.

1. Initialize

1.1 The parameter C is chosen as an integer greater than or equal to one. The algorithm finds C sets of columns, where each set spans the CS of $\mathbf{A}$.

1.2 Set $\mathcal{I} = \emptyset$ as the index set of the sampled columns and set $\mathbf{B} = \mathbf{A}$.

2. Repeat C Times

2.1 Apply the column sampling algorithm presented in [21] to $\mathbf{B}$ to sample $\hat{r}$ columns.

2.2 Store the indices of the sampled columns in set $\mathcal{I}$ and set to zero the columns of $\mathbf{B}$ with indices in $\mathcal{I}$.

Output: The set $\mathcal{I}$ contains the indices of the selected columns.

4. NUMERICAL SIMULATIONS

4.1. Efficient column/row sampling

In this experiment, Algorithm 1 is compared to the randomized decomposition algorithm in [12] which utilizes uniform column/row sampling. It is shown that the proposed sampling strategy can effectively reduce the required number of sampled columns/rows, and makes the proposed method remarkably robust to structures in data. In this experiment the matrix $\mathbf{L}$ follows Assumption 1, $\mathbf{D} \in \mathbb{R}^{2000 \times 4200}$, $r = 60$, $\{n_i\}_{i=1}^{n/2} = \frac{5r}{n}$, and $\{n_i\}_{i=n/2+1}^{n} = \frac{130r}{n}$. The sparse matrix $\mathbf{S}$ follows the Bernoulli model and each element of $\mathbf{S}$ is non-zero with probability 0.02. Also, in this experiment the rows of $\mathbf{L}$ do not exhibit a clustering structure. In addition, the distribution of the rows of $\mathbf{L}$ is random within the RS of $\mathbf{L}$ (since $\mathbf{U}$ spans a random subspace). Thus, if we initiate $\mathbf{D}_w$ with $C_r \hat{r}$ randomly sampled rows where $C_r \geq 2$, the rank of $\mathbf{L}_w$ is equal to r. Accordingly, in this experiment we run Algorithm 1 with only one iteration. We initiate matrix $\mathbf{D}_w$ with 300 randomly sampled rows.

We evaluate the performance of the algorithm for different values of n, i.e., different number of clusters. Fig. 3 shows the performance of the proposed approach and the approach in [12] (which uses uniform random sampling) for different values of m_1 and m_2. For each value of $m_1 = m_2$, we compute the error in LR matrix recovery $\frac{\|\mathbf{L} - \hat{\mathbf{L}}\|_F}{\|\mathbf{L}\|_F}$ averaged over 10 independent runs, and conclude that the algorithm can yield correct decomposition if the average error is less than 0.01. In Fig. 3, the values 0, 1 designate incorrect and correct decomposition, respectively. It can be seen that the presented approach requires a significantly smaller number of samples to yield correct decomposition. This is due to the fact that the randomized algorithm [12] samples both the columns and rows uniformly at random and independently. By contrast, we use $\hat{\mathbf{L}}_w$ to find the most informative columns to form $\mathbf{D}_c$, and also leverage the information embedded in the CS to find the informative rows to update $\mathbf{D}_w$. One can see that when $n = 60$, the algorithm in [12] cannot yield correct decomposition even when $m_1 = m_2 = 1800$.

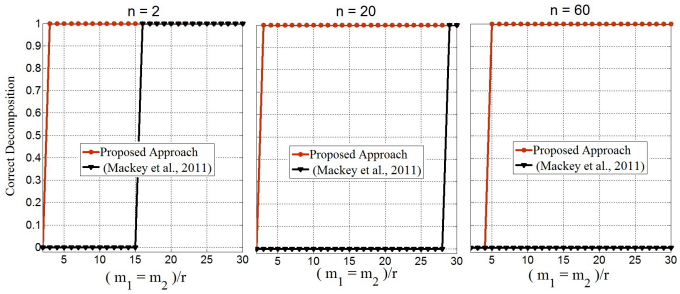

Fig. 3. Performance of the proposed approach and the randomized algorithm in [12]. A value 1 indicates correct decomposition and a value 0 indicates incorrect decomposition.

4.2. Alternating algorithm for column sampling

In this section, we investigate the performance of Algorithm 1 for column sampling. The rank of the selected columns is shown to converge to the rank of $\mathbf{L}$ even when both the rows and columns of $\mathbf{L}$ exhibit a highly structured distribution. To generate $\mathbf{L}$ we first generate a matrix $\mathbf{G}$ as follows:
Matrix $\mathbf{G}$ follows the clustering structure of Assumption 1 with $N_1 = 2000$, $r = 100$, $\{n_i\}_{i=1}^{n/2} = \frac{5r}{n}$, and $\{n_i\}_{i=n/2+1}^{n} = \frac{200r}{n}$. Then, we construct the matrix $\mathbf{U}_g$ from the first r right singular vectors of $\mathbf{G}$. We then generate $\mathbf{G}$ in a similar way and set $\mathbf{V}_g$ equal to the first r right singular vectors of $\mathbf{G}$. Let the matrix $\mathbf{L} = \mathbf{U}_g \mathbf{V}_g^T$. Note that in this simulation we consider a very challenging scenario in which both the columns and rows of $\mathbf{L}$ are highly structured and coherent. The right plot of Fig. 1 shows the rank of the LR component of $\mathbf{D}_c$ after each iteration. The algorithm is shown to converge to the rank of $\mathbf{L}$ in less than 3 iterations even for $n = 100$ clusters. For all values of n, i.e., $n \in \{2, 50, 60\}$, the data is a 10250×10250 matrix.

5. REFERENCES

[1] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[3] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *International Conference on Machine Learning (ICML)*, 2011.

[4] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *Information and Inference*, vol. 2, no. 1, pp. 32–68, 2013.

[5] Q. Ke and T. Kanade, "Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 739–746.

[6] S. Minaee and Y. Wang, "Screen content image segmentation using least absolute deviation fitting," in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 3295–3299.

[7] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma, "Stable principal component pursuit," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2010, pp. 1518–1522.

[8] X. Yuan and J. Yang, "Sparse and low-rank matrix decomposition via alternating direction methods," *Pacific Journal of Optimization*, 2009.

[9] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[10] L. Mackey, A. Talwalkar, and M. I. Jordan, "Distributed matrix completion and robust factorization," *arXiv preprint arXiv:1107.0789*, 2011.

[11] M. Rahmani and G. Atia, "Randomized robust subspace recovery and outlier detection for high dimensional data matrices," *To Appear in the IEEE Transactions on Signal Processing*, 2017, https://arxiv.org/pdf/1505.05901.pdf.

[12] L. W. Mackey, M. I. Jordan, and A. Talwalkar, "Divide-and-conquer matrix factorization," in *Advances in Neural Information Processing Systems*, 2011, pp. 1134–1142.

[13] R. Liu, Z. Lin, S. Wei, and Z. Su, "Solving principal component pursuit in linear time via l_1 filtering," *arXiv preprint arXiv:1108.5359*, 2011.

[14] M. Rahmani and G. Atia, "High dimensional low rank plus sparse matrix decomposition," *To Appear in the IEEE Transactions on Signal Processing*, 2017, https://arxiv.org/pdf/1502.00182.pdf.

[15] X. Li and J. Haupt, "Identifying outliers in large matrices via randomized adaptive compressive sampling," *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1792–1807, 2015.

[16] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse problems*, vol. 23, no. 3, p. 969, 2007.

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

[18] M. Rahmani and G. Atia, "Innovation pursuit: A new approach to subspace clustering," *arXiv preprint arXiv:1512.00907*, 2015.

[19] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[20] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[21] A. Deshpande and L. Rademacher, "Efficient volume sampling for row/column subset selection," in *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010, pp. 329–338.

[22] M. Rahmani and G. Atia, "Robust and scalable column/row sampling from corrupted big data," *arXiv preprint arXiv:1611.05977*, 2016.

[23] A. Rahimpour, A. Taalimi, and H. Qi, "Feature encoding in band-limited distributed surveillance systems," *arXiv preprint arXiv:1612.06423*, 2016.

[24] S. Kumar, M. Mohri, and A. Talwalkar, "On sampling-based approximate spectral decomposition," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 553–560.