Compressive PCA for Low-Rank Matrices on Graphs

Nauman Shahid, Nathanael Perraudin, Gilles Puy, Pierre Vandergheynst

Compressive PCA on Graphs

Nauman Shahid NAUMAN.SHAHID@EPFL.CH
Signal Processing Laboratory 2 (LTS2), EPFL STI IEL, Lausanne, CH-1015, Switzerland

Nathanael Perraudin NATHANAEL.PERRAUDIN@EPFL.CH
Signal Processing Laboratory 2 (LTS2), EPFL STI IEL, Lausanne, CH-1015, Switzerland

Gilles Puy GILLES.PUY@INRIA.FR
INRIA Rennes - Bretagne Atlantique, Campus de Beaulieu, FR-35042 Rennes Cedex, France

Pierre Vandergheynst PIERRE.VANDERGHEYNST@EPFL.CH
Signal Processing Laboratory 2 (LTS2), EPFL STI IEL, Lausanne, CH-1015, Switzerland

Abstract

Randomized algorithms reduce the complexity of low-rank recovery methods only w.r.t dimension p of a big dataset $Y \in \Re^{p\times n}$. However, the case of large n is cumbersome to tackle without sacrificing the recovery. The recently introduced Fast Robust PCA on Graphs (FRPCAG) approximates a recovery method for matrices which are low-rank on graphs constructed between their rows and columns. In this paper we provide a novel framework, Compressive PCA on Graphs (CPCA) for an approximate recovery of such data matrices from sampled measurements. We introduce a RIP condition for low-rank matrices on graphs which enables efficient sampling of the rows and columns to perform FRPCAG on the sampled matrix. Several efficient, parallel and parameter-free decoders are presented along with their theoretical analysis for the low-rank recovery and clustering applications of PCA. On a single core machine, CPCA gains a speed up of p/k over FRPCAG, where $k \ll p$ is the subspace dimension. Numerically, CPCA can efficiently cluster 70,000 MNIST digits in less than a minute and recover a low-rank matrix of size 10304×1000 in 15 secs, which is 6 and 100 times faster than FRPCAG and exact recovery.

1. Introduction

Exact low-rank recovery methods like Robust PCA (Candès et al., 2011) do not scale for big datasets $Y \in \Re^{p\times n}$ (large p and large n). Randomized techniques are used to effectively deal with this problem associated with high dimensional data (the case of large p) (Tropp, 2008; Boutsidis et al., 2009; Witten & Candes, 2013; Li & Haupt, 2015; Halko et al., 2011; Oh et al., 2015; Rahmani & Atia, 2015a;b; Ha & Barber, 2015) using the tools of compression (Davenport et al., 2010). These works improve upon the computational complexity by reducing only the *data dimension* p but still scale in the same manner w.r.t n. Factorized methods (Jiang et al., 2013; Zhang & Zhao, 2013) are faster and scale well but this comes at the price of the loss of convexity. Thus, the case of large n remains unresolved.

For many machine learning applications involving big data, such as clustering, an approximate low-rank representation might suffice. The recently introduced Fast Robust PCA on Graphs (FRPCAG) (Shahid et al., 2015b) approximates a recovery method for clusterable matrices which are low-rank on *graphs* constructed between their rows and columns. As shown in (Shahid et al., 2015b), many real world data matrices can be considered to be low-rank on graphs due to an underlying stationarity assumption (Perraudin & Vandergheynst, 2016).

FRPCAG does not require an SVD and scales linearly with n. However, it still suffers from 1) memory requirements 2) non-parallel implementation 3) cost of k-means for clustering and 4) the cost of parameter tuning for large p and large n. To solve the above mentioned problems with FRPCAG we propose to perform a sampling of the data matrices along rows and columns inspired by the recent work of (Puy et al., 2015) and work with the sampled data.

In this work we present a restricted isometry property (RIP) for low-rank matrices on graphs and relate it to the cumulative coherence of the graph eigenvectors. The proposed row and column sampling strategy is used to solve

.

the low-rank recovery task on the sampled data by using FRPCAG. Finally, we present 3 convex, efficient, parallel, low-cost and parameter-free decoders with theoretical guarantees for two potential applications of PCA: low-rank recovery and clustering. We call our proposed framework "Compressive PCA on graphs" (CPCA). The computational complexity of CPCA is only dominated by the graph construction ($\mathcal{O}(np\log(n))$) using FLANN (Muja & Lowe, 2014) which supports a high degree of parallelism. A speed-up of p/k (where $k \ll p$ is the rank of the subspace or number of clusters in Y) is obtained over FRPCAG solely for low-rank recovery and clustering tasks.

2. Compressive PCA

2.1. Graph Nomenclature

For a matrix $Y \in \Re^{p\times n}$, a $\mathcal{K}$-nearest neighbor undirected graph between the rows or columns of Y is denoted as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is the set of edges and $\mathcal{V}$ is the set of vertices. The first step in the construction of G consists of connecting each y_i to its $\mathcal{K}$ nearest neighbors y_j (using Euclidean distance), resulting in $|\mathcal{E}|$ connections. The $\mathcal{K}$-nearest neighbors are non-symmetric but a symmetric weighted adjacency matrix W is computed via a Gaussian kernel as $W_{ij} = \exp(-\|(y_i - y_j)\|_2^2/\sigma^2)$ if y_j is connected to y_i and 0 otherwise. Let D be the diagonal degree matrix of G which is given as: $D_{ii} = \sum_j W_{ij}$. Then, the combinatorial Laplacian that characterizes the graph G is defined as $\mathcal{L} = D - W$ and its normalized form as $\mathcal{L}_n = D^{-1/2}(D - W)D^{-1/2}$. Throughout this work we use the approximate nearest neighbor algorithm (FLANN (Muja & Lowe, 2014)) for graph construction whose complexity is $\mathcal{O}(np\log(n))$ for $p \ll n$ (Sankaranarayanan et al., 2007) (and it can be performed in parallel).

2.2. Low-rank matrices on graphs

Let $\mathcal{L}_c \in \mathbb{R}^{n\times n}$ be the Laplacian of the graph G_c connecting the different columns of Y and $\mathcal{L}_r \in \mathbb{R}^{p\times p}$ the Laplacian of the graph G_r that connects the rows of Y. Furthermore, let $\mathcal{L}_c = Q\Lambda_c Q^\top = Q_{k_c}\Lambda_{ck_c}Q_{k_c}^\top + \bar{Q}_{k_c}\bar{\Lambda}_{ck_c}\bar{Q}_{k_c}^\top$, where $\Lambda_{k_c} \in \Re^{k_c\times k_c}$ is a diagonal matrix of lower eigenvalues and $\bar{\Lambda}_{k_c} \in \Re^{(n-k_c)\times(n-k_c)}$ is a diagonal matrix of higher graph eigenvalues. Similarly, let $\mathcal{L}_r = P\Lambda_r P^\top = P_{k_r}\Lambda_{rk_r}P_{k_r}^\top + \bar{P}_{k_r}\bar{\Lambda}_{rk_r}\bar{P}_{k_r}^\top$. All the values in Λ_r and Λ_c are sorted in increasing order. For a $\mathcal{K}$-nearest neighbors graph constructed from k_c-clusterable data (along columns) one can expect $\lambda_{k_c}/\lambda_{k_c+1} \approx 0$ as $\lambda_{k_c} \approx 0$ and $\lambda_{k_c} \ll \lambda_{k_c+1}$. We refer to the ratio $\lambda_{k_c}/\lambda_{k_c+1}$ as the spectral gap of $\mathcal{L}_c$. The same holds for the Laplacian $\mathcal{L}_r$. Then, low-rank matrices on graphs can be defined as following and recovered by solving FRPCAG (Shahid et al., 2015b).

Definition 1. *A matrix Y^* is (k_r, k_c)-low-rank on the graphs $\mathcal{L}_r$ and $\mathcal{L}_c$ if $(Y^*)_j \in \text{span}(P_{k_r})$ for all $j = 1,\ldots,n$ and $(Y^*)_i^\top \in \text{span}(Q_{k_c})$ for all $i = 1,\ldots,p$. The set of (k_r, k_c)-low-rank matrices on the graphs $\mathcal{L}_r$ and $\mathcal{L}_c$ is denoted by $\mathcal{LR}(P_{k_r}, Q_{k_c})$.*

2.3. Proposed scheme & organization of the paper

Given a data matrix $Y \in \Re^{p\times n} = \bar{X} + E$, where $\bar{X} \in \mathcal{LR}(P_{k_r}, Q_{k_c})$, the goal is to develop a method to efficiently recover $\bar{X}$. We propose to: 1) Construct Laplacians $\mathcal{L}_r$ and $\mathcal{L}_c$ between the rows and columns of Y using the scheme of Section 2.1. 2) Sample the rows and columns of Y to get a subsampled matrix $\tilde{Y}$ using the sampling scheme of Section 3. 3) Construct the compressed Laplacians $\tilde{\mathcal{L}}_r, \tilde{\mathcal{L}}_c$ from $\mathcal{L}_r, \mathcal{L}_c$ (Section 4). 4) Determine a low-rank $\tilde{X}$ for $\tilde{Y}$ with $\tilde{\mathcal{L}}_r, \tilde{\mathcal{L}}_c$ in algorithm 1 of FRPCAG:

$$\min_{\tilde{X}} \|\tilde{Y} - \tilde{X}\|_1 + \gamma_c \,\text{tr}(\tilde{X}\tilde{\mathcal{L}}_c\tilde{X}^\top) + \gamma_r \,\text{tr}(\tilde{X}^\top\tilde{\mathcal{L}}_r\tilde{X}), \tag{1}$$

where $\tilde{X} = M_r\bar{X}M_c + \tilde{E}$ and $\tilde{E}$ models the errors in the recovery of the subsampled matrix $\tilde{X}$. 5) Use the decoders presented in Section 5 to decode the low-rank X (for the unsampled Y) on graphs $\mathcal{L}_r, \mathcal{L}_c$ if the task is low-rank recovery, or 6) perform k-means on $\tilde{X}$ to get cluster labels $\tilde{C}$ and use the semi-supervised label propagation (presented in Section 6) to get the cluster labels C for the full X.

We directly state the computational complexities of several problems due to the space constraints. The details of calculations are presented in Appendix A.8.

3. How to sample? RIP for low-rank matrices on graphs

Let $M_r \in \Re^{\rho_r\times p}$ be the subsampling matrix for sampling the rows and $M_c \in \Re^{n\times\rho_c}$ for sampling the columns of Y. M_c and M_r are constructed by drawing ρ_c and ρ_r indices $\Omega_c = \{\omega_1\cdots\omega_{\rho_c}\}$ and $\Omega_r = \{\omega_1\cdots\omega_{\rho_r}\}$ uniformly without replacement from the sets $\{1, 2, \cdots, n\}$ and $\{1, 2, \cdots, p\}$ and satisfy:

$$M_c^{ij} = \begin{cases} 1 & \text{if } i = \omega_j \\ 0 & \text{otherwise} \end{cases} \quad M_r^{ij} = \begin{cases} 1 & \text{if } j = \omega_i \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Now, the subsampled data matrix $\widetilde{Y} \in \Re^{\rho_c\times\rho_r}$ can be written as $\widetilde{Y} = M_rYM_c$. CPCA requires M_r and M_c to be constructed such that the "low-rankness" property of the data Y is preserved under sampling. Before discussing this, we introduce a few basic definitions in the context of graphs G_c and G_r.

Definition 2. ***(Graph cumulative coherence).*** *The cumulative coherence of order k_c, k_r of G_c and G_r is:*

$$\nu_{k_c} = \max_{1\le i\le n} \sqrt{n}\|Q_{k_c}^\top\Delta_i\|_2 \;\&\; \nu_{k_r} = \max_{1\le j\le p} \sqrt{p}\|P_{k_r}^\top\Delta_j\|_2,$$

where Δ_i is 1 for a node i and 0 otherwise, i.e, Δ_i is the localized node of the graph. In the above equations $Q_{k_c}^\top\Delta_i$ and $P_{k_r}^\top\Delta_j$ characterize the first k_c and k_r fourier modes

(Shuman et al., 2013) of Δ_i, Δ_j on the graphs G_c and G_r respectively. Thus, the cumulative coherence is a measure of how well the energy of the (k_r, k_c) low-rank matrices spreads over the nodes of the graphs. These quantities exactly control the number of vertices ρ_c and ρ_r that need to be sampled from the graphs G_r and G_c such that the properties of the graphs are preserved (Puy et al., 2015). As we are concerned about data matrices $Y \in \mathcal{LR}(P_{k_r}, Q_{k_c})$, the coherence for the graphs directly imply the coherence conditions on Y.

Theorem 1. ***(Restricted-isometry property (RIP) for low-rank matrices on graphs)*** *Let M_c and M_r be two random subsampling matrices as constructed in* (2)*. For any $\delta, \epsilon \in (0, 1)$, with probability at least $1 - \epsilon$,*

$$(1-\delta)\|Y\|_F^2 \leq \frac{np}{\rho_r \rho_c}\|M_r Y M_c\|_F^2 \leq (1+\delta)\|Y\|_F^2 \quad (3)$$

for all $Y \in \mathcal{LR}(P_{k_r}, Q_{k_c})$ provided that

$$\rho_c \geq \frac{27}{\delta^2}\nu_{k_c}^2 \log\left(\frac{4k_c}{\epsilon}\right) \;\&\; \rho_r \geq \frac{27}{\delta^2}\nu_{k_r}^2 \log\left(\frac{4k_r}{\epsilon}\right), \quad (4)$$

where ν_{k_c}, ν_{k_r} characterize the graph cumulative coherence as in Definition 2 and $\frac{np}{\rho_c \rho_r}$ is just a normalization constant which quantifies the norm conservation in (3)*.*

Proof. Please refer to Appendix A.1. □

Theorem 1 is a direct extension of the RIP for k-bandlimited signals on one graph (Puy et al., 2015). It states that the information in $Y \in \mathcal{LR}(P_{k_r}, Q_{k_c})$ is preserved with overwhelming probability if the sampling matrices (2) are constructed with a uniform sampling strategy satisfying (4). Note that ρ_r and ρ_c depend on the cumulative coherence of the graph eigenvectors. The better spread the eigenvectors are, the smaller is the number of vertices that need to be sampled. As proved in (Puy et al., 2015), we have $v_{k_r}^2 \geq k_r$ and $v_{k_c}^2 \geq k_c$. Therefore, k_c and k_r is the minimum number of columns and rows that we need to sample. Our proposed framework in this paper builds on the case of uniform sampling but it can be easily extended for other sampling schemes (Puy et al., 2015).

4. Graphs for Compressed data

To ensure the preservation of algebraic and spectral properties one can construct the compressed Laplacians $\tilde{\mathcal{L}}_r \in \Re^{\rho_r \times \rho_r}$ and $\tilde{\mathcal{L}}_c \in \Re^{\rho_c \times \rho_c}$ from the kron reduction of $\mathcal{L}_r$ and $\mathcal{L}_c$ (Dorfler & Bullo, 2013). Let Ω be the set of sampled nodes and $\bar{\Omega}$ the complement set and let $\mathcal{L}(A_r, A_c)$ denote the (row, column) sampling of $\mathcal{L}$ w.r.t sets A_r, A_c then the Laplacian $\tilde{\mathcal{L}}_c$ for the columns of compressed matrix $\tilde{Y}$ is:

$$\tilde{\mathcal{L}}_c = \mathcal{L}_c(\Omega, \Omega) - \mathcal{L}_c(\Omega, \bar{\Omega})\mathcal{L}_c^{-1}(\bar{\Omega}, \bar{\Omega})\mathcal{L}_c(\bar{\Omega}, \Omega).$$

Let $\mathcal{L}_c$ has k_c connected components or $\lambda_{k_c}/\lambda_{k_c+1} \approx 0$. Then, as argued in theorem $III.4$ of (Dorfler & Bullo, 2013) two nodes α, β are not connected in $\tilde{\mathcal{L}}_c$ if there is no path between them in $\mathcal{L}_c$ via $\bar{\Omega}$. Thus, if the sampling is done uniformly then one can expect $\tilde{\mathcal{L}}_c$ to have k_c connected components as well. The same holds for $\tilde{\mathcal{L}}_r$ as well. This method involves the multiplication of 3 sparse matrices. The only expensive operation above is the inverse of $\mathcal{L}(\bar{\Omega}, \bar{\Omega})$ which can be performed with $\mathcal{O}(O_l \mathcal{K} n)$ cost using the Lancoz method (Susnjara et al., 2015), where O_l is the number of iterations for Lancoz approximation.

5. Decoders for low-rank recovery

Let $\tilde{X} \in \Re^{\rho_r \times \rho_c}$ be the low-rank solution of (1) with the compressed graph Laplacians $\tilde{\mathcal{L}}_r, \tilde{\mathcal{L}}_c$ and sampled data $\tilde{Y}$. The goal is to decode the low-rank $X \in \Re^{p \times n}$ for the full Y. We assume that $\tilde{X} = M_r \bar{X} M_c + \tilde{E}$, where $\tilde{E} \in \Re^{\rho_r \times \rho_c}$ models the noise incurred by (1).

5.1. Ideal Decoder

A straight-forward way to decode X on the original graphs $\mathcal{L}_r$ and $\mathcal{L}_c$, when one knows the basis P_{k_r}, Q_{k_c} involves solving the following optimization problem:

$$\min_X \|M_r X M_c - \tilde{X}\|_F^2$$
$$\text{s.t: } (X)_i \in span(P_{k_r}),\ (X^\top)_j \in span(Q_{k_c}). \quad (5)$$

Theorem 2. *Let M_r and M_c be such that* (3) *holds and X^* be the solution of* (5) *with $\tilde{X} = M_r \bar{X} M_c + \tilde{E}$, where $\bar{X} \in \mathcal{LR}(P_{k_r}, Q_{k_c})$ and $\tilde{E} \in \Re^{\rho_r \times \rho_c}$. We have:*

$$\|X^* - \bar{X}\|_F \leq 2\sqrt{\frac{np}{\rho_c \rho_r (1-\delta)}}\|\tilde{E}\|_F, \quad (6)$$

where $\sqrt{np/\rho_c \rho_r (1-\delta)}$ is a constant resulting from the norm preservation in (3)*.*

Proof. Please refer to Appendix A.2. □

Thus, the error of the ideal decoder is only bounded by the error $\tilde{E}$ in the low-rank $\tilde{X}$ obtained by solving (1). In fact $\tilde{E}$ depends on the spectral gaps of $\tilde{\mathcal{L}}_c, \tilde{\mathcal{L}}_r$, as explained in Theorem 2 of (Shahid et al., 2015b). Hence, the ideal decoder itself does not introduce any error in the decode stage. The solution for this decoder requires projecting over the eigenvectors P and Q of $\mathcal{L}_r$ and $\mathcal{L}_c$. This is computationally expensive because diagonalization of $\mathcal{L}_r$ and $\mathcal{L}_c$ cost $\mathcal{O}(p^3)$ and $\mathcal{O}(n^3)$ respectively. Moreover, the constants k_r, k_c are not known beforehand and require tuning.

5.2. Alternate Decoder

As the ideal decoder is computationally costly, we propose to decode X from $\tilde{X}$ by using a convex and computationally tractable problem which involves the minimization of graph dirichlet energies.

$$\min_X \|M_r X M_c - \tilde{X}\|_F^2 + \bar{\gamma_c}\,\mathrm{tr}(X \mathcal{L}_c X^\top) + \bar{\gamma_r}\,\mathrm{tr}(X^\top \mathcal{L}_r X). \quad (7)$$

Theorem 3. *Let M_r and M_c be such that* (3) *holds and $\gamma > 0$. Let also X^* be the solution of* (7) *with $\bar{\gamma_c} =$*

γ/λ_{k_c+1}, $\bar{\gamma}_r = \gamma/\lambda_{k_r+1}$, *and* $\tilde{X} = M_r\bar{X}M_c + \tilde{E}$, *where* $\bar{X} \in \mathcal{LR}(P_{k_r}, Q_{k_c})$ *and* $\tilde{E} \in \Re^{\rho_r \times \rho_c}$. *We have:*

$$\|\bar{X}^* - \bar{X}\|_F \leq \sqrt{\frac{np}{\rho_c\rho_r(1-\delta)}}\Bigg[\Big(2 + \frac{1}{\sqrt{2\gamma}}\Big)\|\tilde{E}\|_F + (\frac{1}{\sqrt{2}} + \sqrt{\gamma})\sqrt{\Big(\frac{\lambda_{k_c}}{\lambda_{k_c+1}} + \frac{\lambda_{k_r}}{\lambda_{k_r+1}}\Big)}\|\bar{X}\|_F\Bigg], \quad \textit{and}$$

$$\|E^*\|_F \leq \frac{\|\tilde{E}\|_F}{\sqrt{2\gamma}} + \frac{1}{\sqrt{2}}\sqrt{\Big(\frac{\lambda_{k_c}}{\lambda_{k_c+1}} + \frac{\lambda_{k_r}}{\lambda_{k_r+1}}\Big)}\|\bar{X}\|_F, \tag{8}$$

where $\bar{X}^* = \mathrm{Proj}_{\mathcal{LR}(P_{k_r}, Q_{k_c})}(X)$ *and* $E^* = X^* - \bar{X}^*$. $\mathrm{Proj}_{\mathcal{LR}(P_{k_r}, Q_{k_c})}(.)$ *denotes the orthogonal projection onto* $\mathcal{LR}(P_{k_r}, Q_{k_c})$ *and* γ *depends on the signal to noise ratio.*

Proof. Please refer to Appendix A.3 □

Theorem 3 states that in addition to the error $\tilde{E}$ in $\tilde{X}$ incurred by (1), the error of the alternate decoder (7) also depends on the spectral gaps of the Laplacians $\mathcal{L}_r$ and $\mathcal{L}_c$ respectively. This is the price that one has to pay in order to avoid the expensive ideal decoder. For a k_r, k_c clusterable data Y across the rows and columns, one can expect $\lambda_{k_r}/\lambda_{k_r+1} \approx 0$ and $\lambda_{k_c}/\lambda_{k_c+1} \approx 0$ and the solution is as good as the ideal decoder. Nevertheless, it is possible to reduce this error by using graph filters g such that the ratios $g(\lambda_{k_c})/g(\lambda_{k_c+1})$ and $g(\lambda_{k_r})/g(\lambda_{k_r+1})$ approach zero. However, we do not discuss this approach in our work. It is trivial to solve (7) using a conjugate gradient scheme that costs $\mathcal{O}(InpK)$, where I is the number of iterations for the algorithm to converge.

5.3. Approximate Decoder 1: The subspace upsampling scheme

The alternate decoder proposed above is 1) almost as computationally expensive as FRPCAG and 2) requires tuning two model parameters. In this section we propose an approximate decoder which overcomes these limitations.

5.3.1. Step 1: Splitting the alternate decoder

Using $X = U\Sigma V^\top$ and $\tilde{X} = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$ and the invariance property of the trace under cyclic permutations, we can replace (7) by:

$$\min_{U,V}\|M_rU\Sigma V^\top M_c - \tilde{U}\tilde{\Sigma}\tilde{V}^\top\|_F^2 + \bar{\gamma}_c\,\mathrm{tr}(\Sigma^2V^\top\mathcal{L}_cV) + \bar{\gamma}_r\,\mathrm{tr}(U^\top\mathcal{L}_rU\Sigma^2) \quad \text{s.t: } U^\top U = I_k,\ V^\top V = I_k.$$

$\tilde{\Sigma}$ can be determined by one inexpensive SVD ($\mathcal{O}(\rho_r^2\rho_c)$ for $\rho_r < \rho_c$) of $\widetilde{X}$ and k can be set equal to the number of entries in $\tilde{\Sigma}$ which are above a threshold. If (8) holds for the alternate decoder then $\|\bar{\Sigma}^* - \bar{\Sigma}\|_F$ (where $\bar{\Sigma}^*, \bar{\Sigma}$ are the singular values of $\bar{X}^*, \bar{X}$) is also bounded as argued in the discussion of Appendix A.3. Thus, the singular values Σ and $\tilde{\Sigma}$ of X and $\tilde{X}$ differ approximately by the normalization constant of theorem 1 .

Assuming $\tilde{U} = M_r\bar{U} + \tilde{E}^u$ and $\tilde{V} = \bar{V}M_c + \tilde{E}^v$, where $(\bar{U})_i \in span(P_{k_r})$, $i = 1, \cdots, p$, $(\bar{V}^\top)_j \in span(Q_{k_c})$, $j = 1, \cdots, n$ and $\tilde{E}^u \in \Re^{\rho_r \times \rho_r}$, $\tilde{E}^v \in \Re^{\rho_c \times \rho_c}$ and $\bar{X} = \bar{U}\bar{\Sigma}\bar{V}^\top$, where $\bar{X} \in \mathcal{LR}(P_{k_r}, Q_{k_c})$, we can propose an approximate decoder which separately solves the subspace (U and V) learning problems.

$$\min_U\|M_rU - \tilde{U}\|_F^2 + \gamma^{'}{}_r\,\mathrm{tr}(U^\top\mathcal{L}_rU) \quad \text{s.t:} \quad U^\top U = I_k,$$
$$\min_V\|V^\top M_c - \tilde{V}\|_F^2 + \gamma^{'}{}_c\,\mathrm{tr}(V^\top\mathcal{L}_cV) \quad \text{s.t:} \quad V^\top V = I_k. \tag{9}$$

Now using $X = U\tilde{\Sigma}V^\top\sqrt{np/\rho_r\rho_c(1-\delta)}$ gives a good approximate solution. Solving (9) is as expensive as (7) due to the orthonormality constraints (as explained in appendix A.4). Therefore, we drop the constraints and get

$$\min_U\|M_rU - \tilde{U}\|_F^2 + \gamma^{'}{}_r\,\mathrm{tr}(U^\top\mathcal{L}_rU), \tag{10}$$
$$\min_V\|V^\top M_c - \tilde{V}\|_F^2 + \gamma^{'}{}_c\,\mathrm{tr}(V^\top\mathcal{L}_cV). \tag{11}$$

The solution to (10) & (11) is not orthonormal anymore. The deviation from the orthonormality depends on the constants $\gamma_r^{'}$ and $\gamma_c^{'}$, but $X = U\tilde{\Sigma}V^\top\sqrt{np/\rho_r\rho_c(1-\delta)}$ is still low-rank. The above two problems can be solved using traditional iterative methods for linear systems. In fact, the closed form solutions can be written as:

$$U = (M_r^\top M_r + \gamma_r^{'}\mathcal{L}_r)^{-1}M_r^\top\tilde{U}, \tag{12}$$
$$V = (M_cM_c^\top + \gamma_c^{'}\mathcal{L}_c)^{-1}M_c\tilde{V}. \tag{13}$$

Thus, problems (10) & (11) decode the subspaces U and V such that they are smooth on their respective graphs $\mathcal{L}_r$ and $\mathcal{L}_c$. This can also be referred to as a simultaneous decoding and subspace denoising stage. The columns of U and V are not normalized with the above solution, therefore, a unit norm normalization step is needed at the end.

Theorem 4. *Let* M_r *and* M_c *be such that* (3) *holds and* $\gamma_r', \gamma_c' > 0$. *Let also* U^* *and* V^* *be respectively the solutions of* (10) *and* (11) *with* $\tilde{U} = M_r\bar{U} + \tilde{E}^u$ *and* $\tilde{V} = \bar{V}M_c + \tilde{E}^v$, *where* $(\bar{U})_i \in span(P_{k_r})$, $i = 1, \cdots, p$, $(\bar{V}^\top)_j \in span(Q_{k_c})$, $j = 1, \cdots, n$, $\tilde{E}^u \in \Re^{\rho_r \times \rho_r}$, $\tilde{E}^v \in \Re^{\rho_c \times \rho_c}$. *We have:*

$$\|\bar{U}^* - \bar{U}\|_F \leq \sqrt{\frac{2p}{\rho_r(1-\delta)}}\Bigg[\left(2 + \frac{1}{\sqrt{\gamma_r'\lambda_{k_r+1}}}\right)\|\tilde{E}^u\|_F + \left(\sqrt{\frac{\lambda_{k_r}}{\lambda_{k_r+1}}} + \sqrt{\gamma_r'\lambda_{k_r}}\right)\|\bar{U}\|_F\Bigg], \quad \textit{and}$$

$$\|E^*\|_F \leq \sqrt{\frac{2}{\gamma_r'\lambda_{k_r+1}}}\|\tilde{E}^u\|_F + \sqrt{2\frac{\lambda_{k_r}}{\lambda_{k_r+1}}}\|\bar{U}\|_F.$$

where $\bar{U}^* = P_{k_r}P_{k_r}^\top X$ *and* $E^* = U^* - \bar{U}^*$. *The same inequalities with slight modification also hold for* V^*, *which we omit because of space constraints.*

Table 1. Summary of CPCA and its computational complexity for a dataset $Y \in \Re^{p\times n}$. Throughout we assume that $\mathcal{K}, k, \rho_r, \rho_c, p \ll n$.

Steps	Procedure	Complexity
1	Construct graph Laplacians between the rows $\mathcal{L}_r$ and columns $\mathcal{L}_c$ of Y using Section 2.1.	$\mathcal{O}(np\log(n))$
2	Construct row and column sampling matrices $M_r \in \Re^{\rho_r\times p}$ and $M_c \in \Re^{n\times\rho_c}$ satisfying (2) and theorem 1	–
3	Sample the data matrix Y as $\tilde{Y} = M_r Y M_c$	–
4	Construct the new graph Laplacians between the rows $\tilde{\mathcal{L}}_r$ and columns $\tilde{\mathcal{L}}_c$ of $\tilde{Y}$ using Section 4.	$\mathcal{O}(O_l\mathcal{K}n)$
5	Solve FRPCAG (1) using algorithm 1 in (Shahid et al., 2015b) to get the low-rank $\tilde{X}$	$\mathcal{O}(I\rho_r\rho_c\mathcal{K})$
6	**For clustering**: Run k-means on $\tilde{X}$ to get the cluster labels $\tilde{C}$ Decode the cluster labels C for X on graphs $\mathcal{L}_r, \mathcal{L}_c$ using the semi-supervised label propagation (Section 6)	$\mathcal{O}(I\rho_r\rho_c k)$ $\mathcal{O}(O_l nk)$
7	**For low-rank recovery**: Decode X from $\tilde{X}$ using the approximate decoder (eq. (16) of Section 5.3)	$\mathcal{O}(O_l nk\mathcal{K})$

Proof. Please refer to Appendix A.5. □

As $\bar{X} = \bar{U}\bar{\Sigma}\bar{V}^\top$, we can say that the error with this approximate decoder is upper bounded by the product of the errors of the individual subspace decoders. Also note that the error again depends on the spectral gaps defined by the ratios $\lambda_{k_c}/\lambda_{k_c+1}$ and $\lambda_{k_r}/\lambda_{k_r+1}$. This decoder is less expensive as compared to the alternate decoder and costs $\mathcal{O}(I\mathcal{K}kn)$. Furthermore, each of the vectors of the subspaces U and V can be determined in parallel.

5.3.2. Step 2: Subspace Upsampling

To avoid tuning two parameters $\gamma_r^{'}$ and $\gamma_c^{'}$ we propose to re-formulate the approximate decoder as follows:

$$\min_U \operatorname{tr}(U^\top \mathcal{L}_r U) \quad \text{and} \quad \min_V \operatorname{tr}(V^\top \mathcal{L}_c V)$$
$$\text{s.t: } M_r U = \tilde{U}, \qquad \text{s.t: } M_c^\top V = \tilde{V}. \tag{14}$$

The solution to the above problems is simply a graph upsampling operation as explained in Lemma 1. Since our subspaces $\tilde{U}$ and $\tilde{V}$ are determined by the SVD of $\tilde{X}$ which is noise and outlier free, we can directly upsample them without needing a margin for the noise. Note that now we have a parameter-free decode stage.

Lemma 1. *Let $S \in \Re^{c\times r}$ and $R \in \Re^{d\times r}$ be the two matrices such that $d < r$ and $d < c$. Furthermore, let $M \in \Re^{d\times c}$ be a sampling matrix as constructed in* (2) *and $\mathcal{L} \in \Re^{c\times c}$ be a symmetric positive semi-definite matrix. We can write $S = [S_a^\top | S_b^\top]^\top$, where $S_b \in \Re^{d\times r}$ and $S_a \in \Re^{(c-d)\times r}$ are the known and unknown submatrices of S. Then the exact and unique solution to the following problem:*

$$\min_{S_a} \operatorname{tr}(S^\top \mathcal{L} S), \quad \textit{s.t:} \quad MS = R \tag{15}$$

is given by $S_a = -\mathcal{L}_{aa}^{-1}\mathcal{L}_{ab}R$.

Proof. Please refer to Appendix A.6. □

Using Lemma 1 and the notation of Section 4 we can write:

$$U = \begin{bmatrix} -\mathcal{L}_r^{-1}(\bar{\Omega}_r, \bar{\Omega}_r)\mathcal{L}_r(\bar{\Omega}_r, \Omega_r)\tilde{U} \\ \tilde{U} \end{bmatrix}$$
$$V = \begin{bmatrix} -\mathcal{L}_c^{-1}(\bar{\Omega}_c, \bar{\Omega}_c)\mathcal{L}_c(\bar{\Omega}_c, \Omega_c)\tilde{V} \\ \tilde{V} \end{bmatrix}. \tag{16}$$

Eqs. (16) involve solving a large sparse linear system. If each connected component of the graph has at least one labeled element, $\mathcal{L}_r(\bar{\Omega}_r, \bar{\Omega}_r)$ is full rank and invertible. However, for stability reasons, we prefer a pseudo inversion. Since $\mathcal{L}_r(\bar{\Omega}_r, \bar{\Omega}_r)$ is symmetric, we can use a Lancoz based approximation (Susnjara et al., 2015). The idea is based on the fact that

$$\mathcal{L}_a^\dagger R = g(\mathcal{L}_a)R, \quad \text{where} \quad g(x) = \begin{cases} \frac{1}{x} & x > 10^{-8} \\ 0 & \text{otherwise.} \end{cases}$$

Note that the eqs. (16) can be implemented in parallel for every column of U and V. This gives a significant advantage over the alternate decoder in terms of computation time. The cost of this decoder is $\mathcal{O}(O_l\mathcal{K}kn)$ where O_l is the number of iterations for the Lancoz approximation method. Two other schemes for approximate decoders are presented in Appendix A.7.

6. Decoder for clustering: Semi-supervised Label Propagation

For the clustering application we do not need the full low-rank matrix X. Thus, we propose to do k-means on the low-rank representation of the sampled data $\tilde{X}$ obtained using (1), extract the cluster labels $\tilde{C}$ and then decode the cluster labels C for X on the graphs $\mathcal{L}_r$ and $\mathcal{L}_c$. This scheme is similar to the standard semi-supervised label propagation.

Let $\tilde{C} \in \{0,1\}^{\rho_c\times k}$ be the cluster labels of $\tilde{X}$ (for k clusters) which are obtained by performing k-means. Then $\tilde{C}_{ij} = 1$ if $\tilde{x}_i \in \ j^{th}\ cluster$ and 0 otherwise. We use the same strategy as for the approximate low-rank decoder and propose to solve the following problem:

$$\min_C \operatorname{tr}(C^\top \mathcal{L}_c C) \quad \text{s.t:} \quad M_c^\top C = \tilde{C}. \tag{17}$$

According to Lemma 1, the solution is given by:

$$C = \begin{bmatrix} -\mathcal{L}_c^{-1}(\bar{\Omega}_c, \bar{\Omega}_c)\mathcal{L}_c(\bar{\Omega}_c, \Omega_c)\tilde{C} \\ \tilde{V} \end{bmatrix}. \tag{18}$$

The solution C obtained by solving the above problem is not binary and some maximum pooling thresholding needs

to be done to get the cluster labels, i.e,

$$C_{ij} \leftarrow \begin{cases} 1 & \text{if } C_{ij} = \max\{C_{ij}\ \forall\ j = 1 \cdots k\} \\ 0 & \text{otherwise.} \end{cases}$$

7. Computational Complexity

A summary of all the decoders and their computational complexities is presented in Table 6 of Appendix A.8. The complete CPCA algorithm and the computational complexities of different steps are presented in Table 1. For $\mathcal{K}, k, \rho_r, \rho_c, p \ll n$ CPCA scales as $\mathcal{O}(np\log(n))$ which is in fact the complexity of graph construction using FLANN (Muja & Lowe, 2014). Note that as compared to FRPCAG ($\mathcal{O}(Inp\mathcal{K})$), the complexity of CPCA for low-rank recovery (excluding the construction of graphs G_r, G_c) is $O(O_l nk\mathcal{K})$. Thus a speed-up of p/k is obtained as compared to FRPCAG. A detailed explanation regarding the calculation of complexities of CPCA and other models is presented in Table 7 and Appendix A.8.

8. Experimental Results

We perform two types of experiments corresponding to two applications of PCA 1) Data clustering and 2) Low-rank recovery using two open-source toolboxes: the UNLocBoX (Perraudin et al., 2014a) and the GSPBox (Perraudin et al., 2014b). Due to space constraints some of the results are presented in Appendix A.8.

8.1. Clustering

Datasets: We perform our clustering experiments on 5 benchmark databases (as in (Shahid et al., 2015a;b)): CMU PIE, ORL, YALE, MNIST and USPS. For the USPS and ORL dataset, we further run two types of experiments 1) on subset of datasets and 2) on full datasets. The experiments on the subsets of the datasets take less time so they are used to show the efficiency of our model for a wide variety of noise types. The details of all datasets used are provided in Table 8 of Appendix A.8.

Noise & Errors: To evaluate the robustness of CPCA to corruptions we add 3 different types of noise in all the samples of datasets in different experiments: 1) Gaussian noise and 2) Laplacian noise with standard deviation ranging from 5% to 20% of the original data 3) Sparse noise (randomly corrupted pixels) occupying 5% to 20% of each data sample.

Comparison with other methods: We compare the clustering performance of CPCA with 11 other models including: 1) k-means on original data 2) Laplacian Eigenmaps (LE) (Belkin & Niyogi, 2003) 3) Locally Linear Embedding (LLE) (Roweis & Saul, 2000) 4) Standard PCA 5) Graph Laplacian PCA (GLPCA) (Jiang et al., 2013) 6) Manifold Regularized Matrix Factorization (MMF) (Zhang & Zhao, 2013) 7) Non-negative Matrix Factorization (NMF) (Lee & Seung, 1999) 8) Graph Regularized Non-negative Matrix Factorization (GNMF) (Cai et al., 2011) 9) Robust PCA (RPCA) (Candès et al., 2011) 10) Robust PCA on Graphs (RPCAG) (Shahid et al., 2015a) and 11) Fast Robust PCA on Graphs (FRPCAG) (Shahid et al., 2015b). RPCA and RPCAG are not used for the evaluation of MNIST, USPS large and ORL large datasets due to computational complexity of these models.

Pre-processing: All datasets are transformed to zero-mean and unit standard deviation along the features / rows. For MMF the samples are additionally normalized to unit-norm. For NMF and GNMF only the unit-norm normalization is applied to all the samples of the dataset as NMF based models can only work with non-negative data.

Evaluation Metric: We use *clustering error* as a metric to compare the clustering performance of various models. The clustering error for LE, PCA, GLPCA, MMF, NMF and GNMF is evaluated by performing k-means on the principal components V (note that these models explicitly learn V, where $X = U\Sigma V^\top$). The clustering error for RPCA, RPCAG and FRPCAG is determined by performing k-means directly on the low-rank X. For our CPCA method, k-means is performed on the small low-rank $\tilde{X}$ and then the labels for full X are decoded using the strategy of Section 6.

Parameter Selection: To perform a fair validation for each of the models we use a range of values for the model parameters as presented in Table 9 of Appendix A.8. For a given dataset, each of the models is run for each of the parameter tuples in this table and the parameters corresponding to minimum clustering error are selected for testing purpose. Furthermore, PCA, GLPCA, MMF, NMF and GNMF are non-convex models so they are run 10 times for each of the parameter tuple. RPCA, RPCAG, FRPCAG and CPCA based models are convex so they are run only once. For our proposed CPCA, we use a convention $CPCA(a, b)$, where a and b denote the downsampling factors on the columns and rows respectively. A uniform sampling strategy is always used for CPCA.

The graphs G_r, G_c are constructed using FLANN (Muja & Lowe, 2009) as discussed in Section 2.1. The small graphs $\tilde{G}_r, \tilde{G}_c$ can also be constructed using FLANN or the strategy of Section 4. For all the experiments reported in this paper we use $\mathcal{K}$-nearest neighbors = 10 and Gaussian kernel for the adjacency matrices W. The smoothing parameters σ^2 for the Gaussian kernels are automatically set to the average distance of the $\mathcal{K}$-nearest neighbors.

Discussion: Tables 2 & 3 (and 10 & 11 in Appendix A.8) present the clustering results for USPS small, USPS large, MNIST, ORL small, ORL large, CMU PIE and YALE datasets. Note that not all the models are run for all the datasets due to computational constraints. The best results

Table 2. Clustering error of USPS datasets for different PCA based models. The best results per column are highlighted in bold and the 2nd best in blue. NMF and GNMF require non-negative data so they are not evaluated for USPS because USPS is also negative.

Dataset	Model	no noise	Gaussian noise				Laplacian noise				Sparse noise			
			5%	10%	15%	20%	5%	10%	15%	20%	5%	10%	15%	20%
USPS small ($n = 3500$ $p = 256$)	k-means	0.31	0.31	0.31	0.33	0.32	0.32	0.30	0.36	0.37	0.40	0.45	0.53	0.73
	LLE	0.40	0.34	0.32	0.35	0.24	0.40	0.40	0.33	0.36	0.23	0.30	0.33	0.37
	LE	0.38	0.38	0.38	0.36	0.35	0.38	0.38	0.38	0.38	0.32	0.33	0.36	0.48
	PCA	0.27	0.29	0.25	0.28	0.26	0.29	0.29	0.28	0.24	0.29	0.26	0.26	0.28
	MMF	0.21	**0.20**	0.21	0.22	0.21	0.21	0.21	0.22	0.21	0.27	0.23	0.25	0.27
	GLPCA	**0.20**	**0.20**	0.21	0.23	0.23	0.21	0.21	0.22	0.21	0.26	0.24	0.24	0.28
	RPCA	0.26	0.25	0.23	0.24	0.22	0.26	0.26	0.25	0.24	0.26	0.24	0.23	0.30
	RPCAG	**0.20**	**0.20**	0.21	0.20	0.21	**0.20**	0.21	0.21	0.21	**0.21**	**0.22**	0.23	0.25
	FRPCAG	**0.20**	**0.20**	**0.20**	**0.19**	**0.20**	**0.20**	**0.19**	**0.17**	**0.17**	**0.21**	**0.22**	**0.22**	**0.23**
	CPCA (2,1)	**0.20**	**0.20**	0.21	0.20	0.22	**0.20**	0.22	0.22	0.21	0.23	0.23	0.25	0.28
USPS large ($n = 10000$ $p = 256$)	k-means	0.26	0.26	0.26	0.26	0.28	0.27	0.26	0.26	0.26	0.26	0.25	0.34	0.30
	LLE	0.51	0.29	0.22	0.21	0.22	0.22	0.22	0.22	0.21	0.22	**0.19**	0.26	0.31
	LE	0.33	0.32	0.32	0.27	0.27	0.32	0.34	0.31	0.34	0.35	0.44	0.49	0.53
	PCA	0.21	0.21	0.21	0.22	0.21	0.21	0.21	0.22	0.21	0.22	0.23	0.23	0.23
	MMF	0.24	0.23	0.23	0.24	0.24	0.19	0.23	0.22	0.23	0.24	0.25	0.26	0.26
	GLPCA	0.16	0.16	0.17	0.17	0.16	0.17	0.15	0.15	0.17	**0.18**	**0.19**	**0.21**	0.23
	FRPCAG	**0.15**	0.16	0.17	**0.15**	**0.14**	0.16	0.16	0.16	0.17	**0.18**	0.21	**0.21**	**0.21**
	CPCA (10,2)	**0.15**	**0.14**	**0.14**	**0.15**	**0.14**	**0.14**	**0.14**	**0.13**	**0.14**	**0.18**	**0.19**	0.22	0.24

are highlighted in bold and the second best in blue. From Tables 2 & 3 it is clear that our proposed CPCA model attains comparable clustering results to the state-of-the-art RPCAG and FRPCAG models and better than the others in most of the cases.

It is interesting to compare 1) the time needed for FRPCAG and CPCA to perform clustering 2) the corresponding clustering error and 3) the sub-sampling rates in CPCA. Table 3 shows such a comparison for 70,000 digits of MNIST with (10, 2) times downsampling on the (columns, rows) respectively for CPCA. The time needed by CPCA is an order of magnitude lower than FRPCAG. Surprisingly, the error of CPCA is also lower than FRPCAG. Such cases can also be observed in USPS dataset (Table 2). Downsampling removes spurious samples sometimes and the voting scheme (Section 6) becomes robust to these samples which lie on the cluster borders. Note that the time reported here does not include the construction of graphs G_r, G_c as both methods use the same graphs. Furthermore, these graphs can be constructed in the order of a few seconds if parallel processing is used. The time for CPCA includes steps 2 to 5 and 7 of Table 1.

Table 3. Clustering error and computational times of FRPCAG and CPCA on MNIST dataset ($784 \times$ 70,000). For CPCA the columns and rows of MNIST dataset were downsampled by 10 and 2 respectively. The time reported here corresponds to steps 2 to 5 and 7 of Table 1 and algorithm 1 of (Shahid et al., 2015b) for FRPCAG excluding the construction of graphs G_r, G_c.

Model	FRPCAG	CPCA (10,2)
Error	0.25	0.24
time (secs)	350	58

Table 4 shows the variation of clustering error of CPCA with different downsampling factors across rows and columns of the USPS dataset ($256 \times 10,000$). Obviously, higher downsampling results in an increase in the clustering error. However, note that we can downsample the samples (columns) by a factor of 5 without observing an error increase. The downsampling of features results in an error increase because the number of features for this dataset is only 256 and downsampling results in a loss of information. Similar behavior can also be observed for the ORL small and ORL large datasets in Table 10 where the performance of CPCA is slightly worse than FRPCAG because the number of samples n for ORL is only 400.

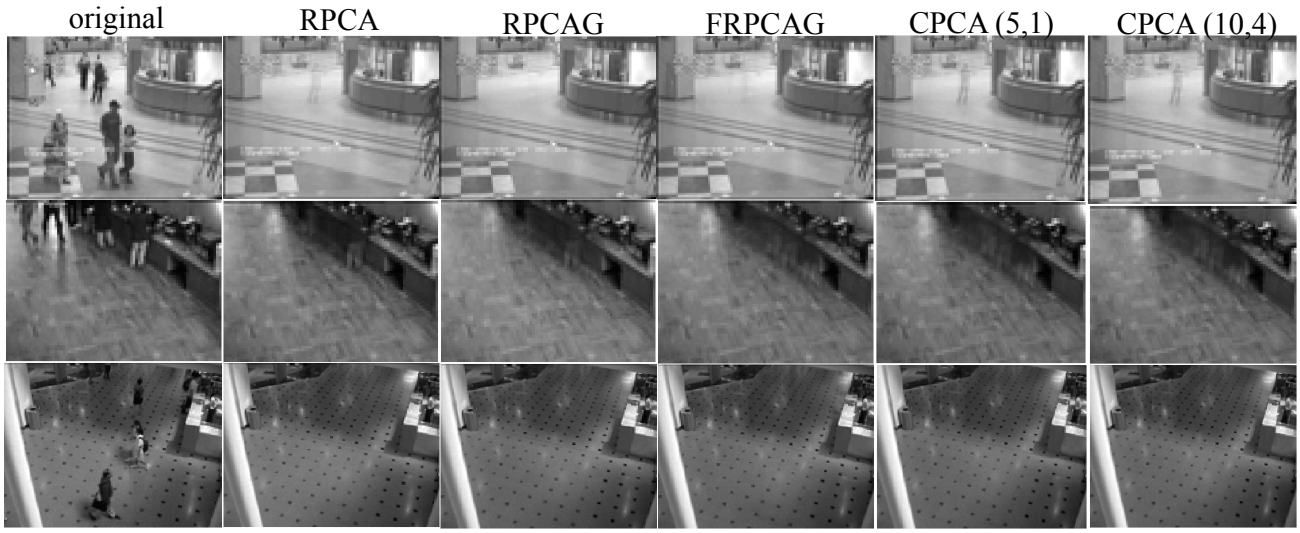
Table 4. Variation of clustering error of CPCA with different uniform downsampling schemes / factors across rows and columns of the USPS dataset ($256 \times 10,000$).

downsampling (rows / cols)	1	5	10	15	20
1	0.16	0.16	0.21	0.21	0.21
2	0.21	0.23	0.23	0.24	0.25
4	0.26	0.30	0.31	0.31	0.31

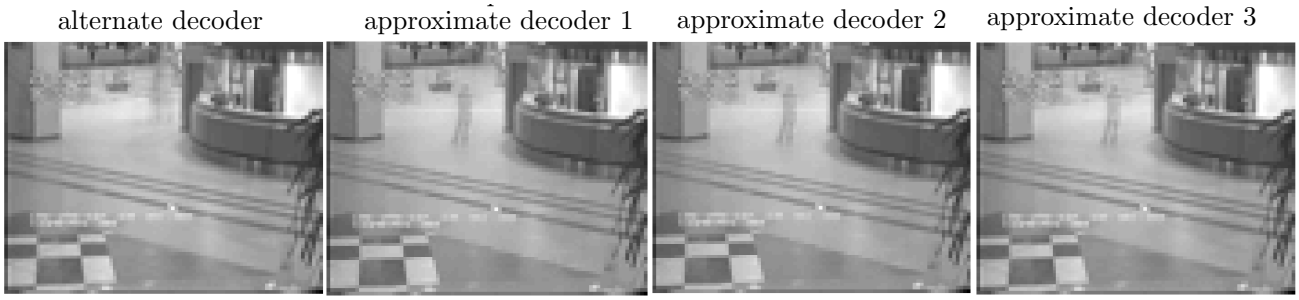
Table 12 in Appendix A.8 shows that the rank of the data is preserved under the proposed downsampling framework even in the presence of noise. For clustering experiments, the lowest error with CPCA occurs when the rank $\approx$ number of clusters, which is an underlying assumption for low-rank matrices on graphs.

8.2. Low-rank recovery

In order to demonstrate the effectiveness of our model to recover low-rank static background from videos we perform experiments on 1000 frames of 3 videos available online. All the frames are vectorized and arranged in a data matrix Y whose columns correspond to frames. The graph G_c is



(a) Static background separation from videos using different PCA based models. The first row corresponds to a frame from the video of a shopping mall lobby, the second row to a restaurant food counter and the third row to an airport lobby. The leftmost plot in each row shows the actual frame, the other 5 show the recovered low-rank using RPCA, RPCAG, FRPCAG and CPCA with two different uniform downsampling schemes.



(b) A quality comparison of various low-rank decoders discussed in this work.

constructed between the columns of Y and the graph G_r is constructed between the rows of Y following the methodology of Section 2.1. Fig. 1a shows the recovery of low-rank frames for one actual frame of each of the videos. The leftmost plot in each row shows the actual frame, the other 5 show the recovered low-rank representations using RPCA, RPCAG, FRPCAG and CPCA with two different uniform downsampling rates. For CPCA the approximate decoder (16) of Section 5.3 is used and k is set such that $\tilde{\Sigma}_{k,k}/\tilde{\Sigma}_{1,1} < 0.1$. For the 2nd and 3rd rows of Fig. 1a it can be seen that our proposed model is able to separate the static backgrounds very accurately from the moving people which do not belong to the static ground truth. However, the quality is slightly compromised in the 1st row where the shadow of the person appears in the low-rank frames recovered with CPCA. In fact, this person remains static for a long time in the video and the uniform sampling compromises the quality slightly. The speed-up observed for these experiments from Table 5 is 10 times over FRPCAG and 1000 times over RPCA and RPCAG.

Fig. 1b presents a comparison of the quality of the low-rank static background extracted using the alternate (7) and approximate decoders discussed in (16) and Appendix A.7 for a video (1st row) of Fig. 1a. Clearly, the alternate decoder performs slightly better than the approximate decoders but at the price of tuning of two model parameters. Fig. 13 in Appendix A.8 presents a comparison of the quality of low-rank for the same video extracted using the approximate decoder (16) using different downsampling factors on the pixels and frames. It is obvious that the quality of low-rank remains intact even with higher downsampling factors.

Table 5. Computational time in seconds of RPCA, RPCAG, FRPCAG and CPCA for low-rank recovery of different videos in Fig. 1a. The time reported here corresponds to steps 2 to 5 and 7 of Table 1, algorithm 1 of (Shahid et al., 2015b) for FRPCAG, (Candès et al., 2011) for RPCA and (Shahid et al., 2015a) for RPCAG, excluding the construction of graphs G_r, G_c.

Videos	RPCA	RPCAG	FRPCAG	CPCA (5,1)	CPCA (10,4)
1	2700	3550	120	21	8
2	1650	2130	85	15	6
3	3650	4100	152	32	11

9. Conclusion

We present Compressive PCA on Graphs (CPCA) which approximates a recovery of low-rank matrices on graphs from their sampled measurements. It is supported by the proposed restricted isometry property (RIP) which is related to the coherence of the eigenvectors of graphs between the rows and columns of the data matrix. Accompanied with several efficient, parallel, parameter free and low-cost decoders, the presented framework gains a several orders of magnitude speed-up over the low-rank recovery methods like Robust PCA. Our theoretical analysis reveals that CPCA targets exact recovery for low-rank matrices which are clusterable across the rows and columns. Extensive clustering experiments on 5 datasets with various types of noise and comparison with 11 state-of-the-art methods reveal the efficiency of our model. CPCA also achieves state-of-the-art results for background separation from videos. The overall complexity is dominated only by the construction of graphs which can be done in parallel using approximate nearest neighbors search.

References

Belkin, Mikhail and Niyogi, Partha. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Boutsidis, Christos, Mahoney, Michael W, and Drineas, Petros. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 968–977. Society for Industrial and Applied Mathematics, 2009.

Cai, Deng, He, Xiaofei, Han, Jiawei, and Huang, Thomas S. Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1548–1560, 2011.

Candès, Emmanuel J, Li, Xiaodong, Ma, Yi, and Wright, John. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

Davenport, Mark, Boufounos, Petros T, Wakin, Michael B, Baraniuk, Richard G, et al. Signal processing with compressive measurements. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):445–460, 2010.

Dorfler, Florian and Bullo, Francesco. Kron reduction of graphs with applications to electrical networks. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 60(1):150–163, 2013.

Elkan, Charles. Using the triangle inequality to accelerate k-means. In *ICML*, volume 3, pp. 147–153, 2003.

Ha, Wooseok and Barber, Rina Foygel. Robust pca with compressed data. In *Advances in Neural Information Processing Systems*, pp. 1927–1935, 2015.

Halko, Nathan, Martinsson, Per-Gunnar, and Tropp, Joel A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Jiang, Bo, Ding, Chris, and Tang, Jin. Graph-laplacian pca: Closed-form solution and robustness. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3492–3498. IEEE, 2013.

Lee, Daniel D and Seung, H Sebastian. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Li, Xingguo and Haupt, Jarvis. Identifying outliers in large matrices via randomized adaptive compressive sampling. *Signal Processing, IEEE Transactions on*, 63 (7):1792–1807, 2015.

Muja, Marius and Lowe, David. Scalable nearest neighbour algorithms for high dimensional data. 2014.

Muja, Marius and Lowe, David G. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2, 2009.

Oh, Tae-Hyun, Matsushita, Yasuyuki, Tai, Yu-Wing, and Kweon, In So. Fast randomized singular value thresholding for nuclear norm minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4484–4493, 2015.

Perraudin, N. and Vandergheynst, P. Stationary signal processing on graphs. *ArXiv e-prints*, January 2016.

Perraudin, N., Shuman, D., Puy, G., and Vandergheynst, P. UNLocBoX A matlab convex optimization toolbox using proximal splitting methods. *ArXiv e-prints*, February 2014a.

Perraudin, Nathanaël, Paratte, Johan, Shuman, David, Kalofolias, Vassilis, Vandergheynst, Pierre, and Hammond, David K. GSPBOX: A toolbox for signal processing on graphs. *ArXiv e-prints*, August 2014b.

Puy, Gilles, Tremblay, Nicolas, Gribonval, Rémi, and Vandergheynst, Pierre. Random sampling of bandlimited signals on graphs. *arXiv preprint arXiv:1511.05118*, 2015.

Rahmani, Mostafa and Atia, George. High dimensional low rank plus sparse matrix decomposition. *arXiv preprint arXiv:1502.00182*, 2015a.

Rahmani, Mostafa and Atia, George K. Randomized robust subspace recovery for big data. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pp. 1–6. IEEE, 2015b.

Roweis, Sam T and Saul, Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Sankaranarayanan, Jagan, Samet, Hanan, and Varshney, Amitabh. A fast all nearest neighbor algorithm for applications involving large point-clouds. *Computers & Graphics*, 31(2):157–174, 2007.

Shahid, Nauman, Kalofolias, Vassilis, Bresson, Xavier, Bronstein, Michael, and Vandergheynst, Pierre. Robust principal component analysis on graphs. *arXiv preprint arXiv:1504.06151*, 2015a.

Shahid, Nauman, Perraudin, Nathanael, Kalofolias, Vassilis, and Vandergheynst, Pierre. Fast robust pca on graphs. *arXiv preprint arXiv:1507.08173*, 2015b.

Shuman, David I, Narang, Sunil K, Frossard, Pascal, Ortega, Antonio, and Vandergheynst, Pierre. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *Signal Processing Magazine, IEEE*, 30 (3):83–98, 2013.

Susnjara, Ana, Perraudin, Nathanael, Kressner, Daniel, and Vandergheynst, Pierre. Accelerated filtering on graphs using lanczos method. *arXiv preprint arXiv:1509.04537*, 2015.

Tropp, Joel A. On the conditioning of random subdictionaries. *Applied and Computational Harmonic Analysis*, 25 (1):1–24, 2008.

Witten, Rafi and Candes, Emmanuel. Randomized algorithms for low-rank matrix factorizations: sharp performance bounds. *Algorithmica*, pp. 1–18, 2013.

Zhang, Zhenyue and Zhao, Keke. Low-rank matrix approximation with manifold regularization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(7): 1717–1729, 2013.

A. Appendices

A.1. Proof of theorem 1

Proof. We start with the sampling of the rows. Theorem 5 in (Puy et al., 2015) shows that for any $\delta_r, \epsilon_r \in (0,1)$, with probability at least $1-\epsilon_r$,

$$(1-\delta_r)\|z\|_2^2 \leq \frac{p}{\rho_r}\|M_r z\|_2^2 \leq (1+\delta_r)\|z\|_2^2$$

for all $z \in span(P_{k_r})$ provided that

$$\rho_r \geq \frac{3}{\delta_r^2}\nu_{k_r}^2 \log\left(\frac{2k_r}{\epsilon_r}\right). \tag{19}$$

Notice that Theorem 5 in (Puy et al., 2015) is a uniform result. As a consequence, with probability at least $1-\epsilon_r$,

$$(1-\delta_r)\|y_i\|_2^2 \leq \frac{p}{\rho_r}\|M_r y_i\|_2^2 \leq (1+\delta_r)\|y_i\|_2^2, \quad i=1,\dots,n, \tag{20}$$

for all $y_1,\dots,y_n \in span(P_{k_r})$ provided that (19) holds. Summing the previous inequalities over all i shows that, with probability at least $1-\epsilon_r$,

$$(1-\delta_r)\|Y\|_F^2 \leq \frac{p}{\rho_r}\|M_r Y\|_F^2 \leq (1+\delta_r)\|Y\|_F^2, \tag{21}$$

for all $Y \in \Re^{p\times n}$ with column-vectors in $span(P_{k_r})$.

Let us continue with the sampling of the columns. Again, Theorem 5 in (Puy et al., 2015) shows that for any $\delta_c, \epsilon_c \in (0,1)$, with probability at least $1-\epsilon_c$,

$$(1-\delta_c)\|w\|_2^2 \leq \frac{n}{\rho_c}\|w^\top M_c\|_2^2 \leq (1+\delta_c)\|w\|_2^2$$

for all $w \in span(Q_{k_c})$ provided that

$$\rho_c \geq \frac{3}{\delta_c^2}\nu_{k_c}^2 \log\left(\frac{2k_c}{\epsilon_c}\right). \tag{22}$$

As a consequence, with probability at least $1-\epsilon_c$,

$$(1-\delta_c)\|z_i\|_2^2 \leq \frac{n}{\rho_c}\|z_i^\top M_c\|_2^2 \leq (1+\delta_c)\|z_i\|_2^2, \quad i=1,\dots,\rho_r, \tag{23}$$

for all $z_1,\dots,z_{\rho_r} \in span(Q_{k_c})$ provided that (22) holds. Summing the previous inequalities over all i shows that, with probability at least $1-\epsilon_c$,

$$(1-\delta_c)\|Z\|_F^2 \leq \frac{n}{\rho_c}\|Z M_c\|_F^2 \leq (1+\delta_c)\|Z\|_F^2 \tag{24}$$

for all $Z \in \Re^{\rho_r\times n}$ with row-vectors in $span(Q_{k_c})$. In particular, this property holds, with at least the same probability, for all matrices Z of the form $M_r Y$ where $Y \in \Re^{p\times n}$ is a matrix with row-vectors in $span(Q_{k_c})$.

We now continue by combining (21) and (24). We obtain that

$$(1-\delta_c)(1-\delta_r)\|Y\|_F^2 \leq \frac{np}{\rho_c\rho_r}\|M_r Y M_c\|_F^2 \leq (1+\delta_c)(1-\delta_r)\|Y\|_F^2 \tag{25}$$

for all $Y \in \mathbb{R}^{p\times n}$ with column-vectors in $span(P_{k_r})$ and row-vectors in $span(Q_{k_c})$, provided that (19) and (22) hold. It remains to compute the probability that (25) holds. Property (25) does not hold if (21) or (24) do not hold. Using the union bound, (25) does not hold with probability at most $\epsilon_r+\epsilon_c$. To finish the proof, one just need to choose $\epsilon_r=\epsilon_c=\epsilon/2$ and $\delta_r=\delta_c=\delta/3$, and notice that $(1+\delta/3)^2 \leq 1+\delta$ and $(1-\delta/3)^2 \geq 1-\delta$ for $\delta \in (0,1)$. □

A.2. Proof of Theorem 2

Proof. Using the optimality condition we have, for any $Z \in \mathbb{R}^{p\times n}$,

$$\|M_r X^* M_c - \tilde{X}\|_F \leq \|M_r Z M_c - \tilde{X}\|_F.$$

For $Z = \bar{X}$, we have

$$\|M_r X^* M_c - \tilde{X}\|_F \leq \|M_r \bar{X} M_c - \tilde{X}\|_F,$$

which gives

$$\|M_r X^* M_c - M_r \bar{X} M_c - \tilde{E}\|_F \leq \|\tilde{E}\|_F.$$

As (3) holds, we have

$$\begin{aligned}\|M_r X^* M_c - M_r \bar{X} M_c - \tilde{E}\|_F &\geq \|M_r(X^* - \bar{X})M_c\|_F - \|\tilde{E}\|_F \\ &\geq \sqrt{\frac{\rho_r \rho_c (1-\delta)}{np}}\|X^* - \bar{X}\|_F - \|\tilde{E}\|_F.\end{aligned}$$

Therefore, we get

$$\|X^* - \bar{X}\|_F \leq 2\sqrt{\frac{np}{\rho_r \rho_c (1-\delta)}}\|\tilde{E}\|_F.$$

□

A.3. Proof of Theorem 3

Proof. Using the optimality condition we have for any $Z \in \Re^{p\times n}$ and optimal solution $X^* = \bar{X}^* + E^*$:

$$\|M_r X^* M_c - \tilde{X}\|_F^2 + \bar{\gamma_c}\,\mathrm{tr}(X^* \mathcal{L}_c X^{*\top}) + \bar{\gamma_r}\,\mathrm{tr}(X^{*\top}\mathcal{L}_r X^*) \leq \|M_r Z M_c - \tilde{X}\|_F^2 + \bar{\gamma_c}\,\mathrm{tr}(Z\mathcal{L}_c Z^\top) + \bar{\gamma_r}\,\mathrm{tr}(Z^\top \mathcal{L}_r Z) \quad (26)$$

using $Z = \bar{X} = P_{k_r} Y_b Q_{k_c}^\top$ as in the proof of theorem 2 in (Shahid et al., 2015b), where $Y_b \in \Re^{k_r \times k_c}$ and it is not necessarily diagonal. Note that $\|Y_b\|_F = \|\bar{X}\|_F$, $Q_{k_c}^\top Q_{k_c} = I_{k_c}$, $P_{k_r}^\top P_{k_r} = I_{k_r}$, $\bar{Q}_{k_c}^\top Q_{k_c} = 0$, $\bar{P}_{k_r}^\top P_{k_r} = 0$. From the proof of theorem 2 in (Shahid et al., 2015b) we also know that:

$$\begin{aligned}\mathrm{tr}(\bar{X}\mathcal{L}_c(\bar{X})^\top) &\leq \lambda_{k_c}\|\bar{X}\|_F^2 \\ \mathrm{tr}(\bar{X}^\top \mathcal{L}_r(\bar{X})) &\leq \lambda_{k_r}\|\bar{X}\|_F^2 \\ \mathrm{tr}(X^*\mathcal{L}_c(X^*)^\top) &\geq \lambda_{k_c+1}\|X^*\bar{Q}_{k_c}\|_F^2 \\ \mathrm{tr}(X^*\mathcal{L}_r(X^*)^\top) &\geq \lambda_{k_r+1}\|\bar{P}_{k_r}^\top X^*\|_F^2\end{aligned}$$

Now using all this information in (26) we get

$$\|M_r X^* M_c - \tilde{X}\|_F^2 + \bar{\gamma}_c\lambda_{k_c+1}\|X^*\bar{Q}_{k_c}\|_F^2 + \bar{\gamma}_r\lambda_{k_r+1}\|\bar{P}_{k_r}^\top X^*\|_F^2 \leq \|\tilde{E}\|_F^2 + (\bar{\gamma}_c\lambda_{k_c} + \bar{\gamma}_r\lambda_{k_r})\|\bar{X}\|_F^2$$

From above we have:

$$\|M_r X^* M_c - \tilde{X}\|_F \leq \|\tilde{E}\|_F + \sqrt{(\bar{\gamma}_c\lambda_{k_c} + \bar{\gamma}_r\lambda_{k_r})}\|\bar{X}\|_F \quad (27)$$

and

$$\sqrt{(\bar{\gamma}_c\lambda_{k_c+1}\|X^*\bar{Q}_{k_c}\|_F^2 + \bar{\gamma}_r\lambda_{k_r+1}\|\bar{P}_{k_r}^\top X^*\|_F^2)} \leq \|\tilde{E}\|_F + \sqrt{(\bar{\gamma}_c\lambda_{k_c} + \bar{\gamma}_r\lambda_{k_r})}\|\bar{X}\|_F \quad (28)$$

using

$$\bar{\gamma}_c = \gamma\frac{1}{\lambda_{k_c+1}} \quad \text{and} \quad \bar{\gamma}_r = \gamma\frac{1}{\lambda_{k_r+1}},$$

and

$$\|E^*\|_F^2 = \|X^*\bar{Q}_{k_c}\|_F^2 = \|\bar{P}_{k_r}^\top X^*\|_F^2$$

we get:

$$\|M_r X^* M_c - \tilde{X}\|_F \leq \|\tilde{E}\|_F + \sqrt{\gamma\Big(\frac{\lambda_{k_c}}{\lambda_{k_c+1}} + \frac{\lambda_{k_r}}{\lambda_{k_r+1}}\Big)}\|\bar{X}\|_F \tag{29}$$

and

$$\sqrt{2\gamma}\|E^*\|_F \leq \|\tilde{E}\|_F + \sqrt{\gamma\Big(\frac{\lambda_{k_c}}{\lambda_{k_c+1}} + \frac{\lambda_{k_r}}{\lambda_{k_r+1}}\Big)}\|\bar{X}\|_F \tag{30}$$

which implies

$$\|E^*\|_F \leq \frac{\|\tilde{E}\|_F}{\sqrt{2\gamma}} + \frac{1}{\sqrt{2}}\sqrt{\gamma\Big(\frac{\lambda_{k_c}}{\lambda_{k_c+1}} + \frac{\lambda_{k_r}}{\lambda_{k_r+1}}\Big)}\|\bar{X}\|_F \tag{31}$$

Focus on $\|M_r X^* M_c - \tilde{X}\|_F^2$ now. As M_r, M_c are constructed with a sampling without replacement, we have $\|M_r E^* M_c\|_F \leq \|E^*\|_F$. Now using the above facts and the RIP we get:

$$\begin{aligned}\|M_r X^* M_c - \tilde{X}\|_F &= \|M_r(\bar{X}^* + E^*)M_c - M_r\bar{X}M_c - \tilde{E}\|_F \\ &\geq \sqrt{\frac{\rho_r\rho_c(1-\delta)}{np}}\|\bar{X}^* - \bar{X}\|_F - \|\tilde{E}\|_F - \|E^*\|_F\end{aligned}$$

this implies

$$\|\bar{X}^* - \bar{X}\|_F \leq \sqrt{\frac{np}{\rho_c\rho_r(1-\delta)}}\left[\Big(2 + \frac{1}{\sqrt{2\gamma}}\Big)\|\tilde{E}\|_F + (\frac{1}{\sqrt{2}} + \sqrt{\gamma})\sqrt{\Big(\frac{\lambda_{k_c}}{\lambda_{k_c+1}} + \frac{\lambda_{k_r}}{\lambda_{k_r+1}}\Big)}\|\bar{X}\|_F\right]$$

□

Discussion Let $A_1, A_2 \in \Re^{p\times n}$ and $A_1 = U_1 S_1 V_1^T$, $A_2 = U_2 S_2 V_2^T$ then if $\|A_1 - A_2\|_F^2 \to 0$, then $S_1 \to S_2$.

We observe that

$$\|A_1 - A_2\|_F^2 = \|U_1 S_1 V_1^T - U_2 S_2 V_2^T\|_F^2 = \|U_2^T U_1 S_1 V_1^T V_2 - S_2\|_F^2$$

which implies that $U_2^T U_1 S_1 V_1^T V_2 \approx S_2$. This is equivalent to saying that for the significant values of S_2, the orthonormal matrices $U_2^T U_1$ and $V_1^T V_2$ have to be almost diagonal. As a result, for the significant values of S_2, U_2 and V_2 have to be aligned with U_1 and V_1. The same reason also implies that $S_1 \approx S_2$.

A.4. Solution of eq. (9)

Let us examine how to solve (9). The problem can be reformulated as:

$$\min_U \operatorname{tr}(U^\top \mathcal{L}_r U) \quad \text{s.t:} \quad U^\top U = I_k,\ \|M_r U - \tilde{U}\|_F^2 < \epsilon$$

Let $U^{'}$ is the zero appended matrix of $\tilde{U}$, then we can re-write it as:

$$\min_U \operatorname{tr}(U^\top \mathcal{L}_r U) \quad \text{s.t:} \quad U^\top U = I_k,\ \|M_r(U - U^{'})\|_F^2 < \epsilon$$

The above problem is equivalent to (9), as the term $\|M_r(U - U^{'})\|_F^2$ has been removed from the objective and introduced as a constraint. Note that the constant γ_r is not needed anymore. The new model parameter ϵ controls the radius of the L_2 ball $\|M_r(U - U^{'})\|_F^2$. In simple words it controls how much noise is tolerated by the projection of U on the ball that is centered at $U^{'}$. To solve the above problem one needs to split it down into two sub-problems and solve iteratively between:

1. The optimization $\min_U \operatorname{tr}(U^\top \mathcal{L}_r U)$ s.t: $U^\top U = I_k$. The solution to this problem is given by the lowest k eigenvectors of $\mathcal{L}_r$. Thus it requires a complexity of $\mathcal{O}((n+p)k^2)$ for solving both problems (9).
2. The projection on the L_2 ball $\|M_r(U - U^{'})\|_F^2$ whose complexity is $\mathcal{O}(\rho_c + \rho_r)$.

Thus the solution requires a double iteration with a complexity of $\mathcal{O}(Ink^2)$ and is almost as expensive as FRPCAG.

A.5. Proof of Theorem 4

Proof. We can write (10) and (11) as following:

$$\min_{u_1 \cdots u_p} \sum_{i=1}^{p} \left[\|M_r u_i - \tilde{u}_i\|_2^2 + \gamma^{'}{}_r u_i^\top \mathcal{L}_r u_i \right] \tag{32}$$

$$\min_{v_1 \cdots v_n} \sum_{i=1}^{n} \left[\|M_c^\top v_i - \tilde{v}_i\|_2^2 + \gamma^{'}{}_c v_i^\top \mathcal{L}_c v_i \right] \tag{33}$$

In this proof, we only treat Problem (32) and the recovery of $\bar{U}$. The proof for Problem (11) and the recovery of $\bar{V}$ is identical. The above two problems can be solved independently for every i. From theorem 3.2 of (Puy et al., 2015) we obtain:

$$\|\bar{u}_i^* - \bar{u}_i\|_2 \leq \sqrt{\frac{p}{\rho_r(1-\delta)}} \left[\left(2 + \frac{1}{\sqrt{\gamma_r' \lambda_{k_r+1}}} \right) \|\tilde{e}_i^u\|_2 + \left(\sqrt{\frac{\lambda_{k_r}}{\lambda_{k_r+1}}} + \sqrt{\gamma_r' \lambda_{k_r}} \right) \|\bar{u}_i\|_2 \right],$$

and

$$\|e_i^*\|_2 \leq \frac{1}{\sqrt{\gamma_r' \lambda_{k_r+1}}} \|\tilde{e}_i^u\|_2 + \sqrt{\frac{\lambda_{k_r}}{\lambda_{k_r+1}}} \|\bar{u}_i\|_2,$$

which implies

$$\|\bar{u}_i^* - \bar{u}_i\|_2^2 \leq 2\frac{p}{\rho_r(1-\delta)} \left[\left(2 + \frac{1}{\sqrt{\gamma_r' \lambda_{k_r+1}}} \right)^2 \|\tilde{e}_i^u\|_2^2 + \left(\sqrt{\frac{\lambda_{k_r}}{\lambda_{k_r+1}}} + \sqrt{\gamma_r' \lambda_{k_r}} \right)^2 \|\bar{u}_i\|_2^2 \right],$$

and

$$\|e_i^*\|_2^2 \leq \frac{2}{\gamma_r' \lambda_{k_r+1}} \|\tilde{e}_i^u\|_2^2 + 2\frac{\lambda_{k_r}}{\lambda_{k_r+1}} \|\bar{u}_i\|_2^2.$$

Summing the previous inequalities over all i's yields

$$\|\bar{U}^* - \bar{U}\|_F^2 \leq 2\frac{p}{\rho_r(1-\delta)} \left[\left(2 + \frac{1}{\sqrt{\gamma_r' \lambda_{k_r+1}}} \right)^2 \|\tilde{E}^u\|_F^2 + \left(\sqrt{\frac{\lambda_{k_r}}{\lambda_{k_r+1}}} + \sqrt{\gamma_r' \lambda_{k_r}} \right)^2 \|\bar{U}\|_F^2 \right],$$

and

$$\|E^*\|_F^2 \leq \frac{2}{\gamma_r' \lambda_{k_r+1}} \|\tilde{E}^u\|_F^2 + 2\frac{\lambda_{k_r}}{\lambda_{k_r+1}} \|\bar{U}\|_F^2.$$

Taking the square root of both inequalities terminates the proof. Similarly, the expressions for $\bar{V}$ can be derived:

$$\|\bar{V}^* - \bar{V}\|_F \leq \sqrt{\frac{2n}{\rho_c(1-\delta)}} \left[\left(2 + \frac{1}{\sqrt{\gamma_c' \lambda_{k_c+1}}} \right) \|\tilde{E}^v\|_F + \left(\sqrt{\frac{\lambda_{k_c}}{\lambda_{k_c+1}}} + \sqrt{\gamma_c' \lambda_{k_c}} \right) \|\bar{V}\|_F \right]$$

and

$$\|E^*\|_F \leq \sqrt{\frac{2}{\gamma_c' \lambda_{k_c+1}}} \|\tilde{E}^v\|_F + \sqrt{2\frac{\lambda_{k_c}}{\lambda_{k_c+1}}} \|\bar{V}\|_F.$$

□

A.6. Proof of Lemma 1

Proof. Let $S = [S_a^\top | S_b^\top]^\top$. Further we split $\mathcal{L}$ into submatrices as follows:

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_{aa} & \mathcal{L}_{ab} \\ \mathcal{L}_{ba} & \mathcal{L}_{bb} \end{bmatrix}$$

Now (15) can be written as:

$$\begin{aligned} &\min_{S_a} \begin{bmatrix} S_a \\ S_b \end{bmatrix}^\top \begin{bmatrix} \mathcal{L}_{aa} & \mathcal{L}_{ab} \\ \mathcal{L}_{ba} & \mathcal{L}_{bb} \end{bmatrix} \begin{bmatrix} S_a \\ S_b \end{bmatrix} \\ &\text{s.t: } S_b = R \end{aligned}$$

further expanding we get:

$$\min_{S_a} S_a^\top \mathcal{L}_{aa} S_a + S_a^\top \mathcal{L}_{ab} R + R^\top \mathcal{L}_{ba} S_a + R \mathcal{L}_{bb} R$$

using $\nabla S_a = 0$, we get:

$$2\mathcal{L}_{ab} R + 2\mathcal{L}_{aa} S_a = 0$$

$$S_a = -\mathcal{L}_{aa}^{-1} \mathcal{L}_{ab} R$$

□

A.7. Approximate decoders 2 and 3

Approximate decoder 2:

Alternatively, if we have access to the complete data matrix Y then we can reduce the complexity further by performing a graph-upsampling for only one of the two subspaces U and V. Suppose we do the upsampling only for U, then the approximate decoder 2 can be written as:

$$\begin{aligned} &\min_U \operatorname{tr}(U^\top \mathcal{L}_r U) \\ &\text{s.t: } M_r U = \tilde{U}. \end{aligned}$$

The solution for U is given by eq. 16. Then, we can write V as:

$$V = Y^\top U \tilde{\Sigma}^{-1} \sqrt{\frac{\rho_c \rho_r (1-\delta)}{np}}$$

However, we do not need to explicitly determine V here. Instead the low-rank X can be determined directly from U with the projection given below:

$$X = U\tilde{\Sigma}\sqrt{\frac{np}{\rho_c \rho_r (1-\delta)}} V^\top = U\tilde{\Sigma}\sqrt{\frac{np}{\rho_c \rho_r (1-\delta)}} (Y^\top U \tilde{\Sigma}^{-1})^\top \sqrt{\frac{\rho_c \rho_r (1-\delta)}{np}} = UU^\top Y.$$

Approximate decoder 3:

Similar to the approximate decoder 2, we can propose another approximate decoder 3 which performs a graph upsampling on V and then determines U via matrix multiplication operation.

$$\begin{aligned} &\min_V \operatorname{tr}(V^\top \mathcal{L}_c V) \\ &\text{s.t: } M_c^\top V = \tilde{V} \end{aligned}$$

The solution for V is given by eq. 16. Using the similar trick as for the approximate decoder 2, we can compute X without computing U. Therefore,

$$X = YVV^\top$$

For the proposed approximate decoders, we would need to do one SVD to determine the singular values $\tilde{(\Sigma)}$. However, note that this SVD is on the compressed matrix $\tilde{X} \in \Re^{\rho_r \times \rho_c}$. Thus, it is inexpensive $\mathcal{O}(\rho_r^2 \rho_c)$ assuming that $\rho_r < \rho_c$.

A.8. Computational Complexities & Additional Results

We present the computational complexity of all the models considered in this work. For a matrix $X \in \Re^{p \times n}$, let I denote the number of iterations for the algorithms to converge, p is the number of features, n is the number of samples, ρ_r, ρ_c are the number of features and samples for the compressed data $\tilde{Y}$ and satisfy eq. (2) and theorem 1, k is the rank of the low-dimensional space or the number of clusters, $\mathcal{K}$ is the number of nearest neighbors for graph construction, O_l, O_c correspond to the number of iterations in the Lancoz and Chebyshev approximation methods. All the models which use the graph G_c are marked by '+'. The construction of graph G_r is included only in FRPCAG and CPCA. Furthermore,

1. We assume that $\mathcal{K}, k, \rho_r, \rho_c, p << n$ and $n + p + k + \mathcal{K} + \rho_r + \rho_c \approx n$.
2. The complexity of $\|Y - X\|_1$ is $\mathcal{O}(np)$ per iteration and that of $\|\tilde{Y} - \tilde{X}\|_1$ is $\mathcal{O}(\rho_c \rho_r)$.
3. The complexity of the computations corresponding to the graph regularization $\mathrm{tr}(X\mathcal{L}_c X^\top) + \mathrm{tr}(X^\top \mathcal{L}_r X) = \mathcal{O}(p|\mathcal{E}_c| + n|\mathcal{E}_r|) = \mathcal{O}(pn\mathcal{K} + np\mathcal{K})$, where $\mathcal{E}_r, \mathcal{E}_c$ denote the number of non-zeros in $\mathcal{L}_r, \mathcal{L}_c$ respectively. Note that we use the $\mathcal{K}$-nearest neighbors graphs so $\mathcal{E}_r \approx \mathcal{K}p$ and $\mathcal{E}_c \approx \mathcal{K}n$.
4. The complexity for the construction of $\tilde{\mathcal{L}}_c$ and $\tilde{\mathcal{L}}_r$ for compressed data $\tilde{Y}$ is negligible if FLANN is used, i.e, $\mathcal{O}(\rho_c \rho_r \log(\rho_c))$ and $\mathcal{O}(\rho_c \rho_r \log(\rho_r))$. However, if the kron reduction strategy of Section 4 is used then the cost is $\mathcal{O}(\mathcal{K}O_l(n + p)) \approx \mathcal{O}(\mathcal{K}O_l n)$.
5. We use the complexity $\mathcal{O}(np^2)$ for all the SVD computations on the matrix $X \in \Re^{p \times n}$ and $\mathcal{O}(\rho_c \rho_r^2)$ for $\tilde{X} \in \Re^{\rho_c \times \rho_r}$.
6. The complexity of $\|M_r X M_c - \tilde{X}\|_F^2$ is negligible as compared to the graph regularization terms $\mathrm{tr}(X\mathcal{L}_c X^\top) + \mathrm{tr}(X^\top \mathcal{L}_r X)$.
7. We use the approximate decoders for low-rank recovery in the complexity calculations (eq. (16) in Section 5.3). All the decoders for low-rank recovery are summarized in Table 6.
8. The complexity of k-means (Elkan, 2003) is $\mathcal{O}(Inkp)$ for a matrix $X \in \Re^{p \times n}$ and $\mathcal{O}(I\rho_r \rho_c k)$ for a matrix $\tilde{X} \in \Re^{\rho_r \times \rho_c}$.

Table 6. A summary and computational complexities of all the decoders proposed in this work. The Lancoz method used here is presented in (Susnjara et al., 2015).

Type	Low-rank			
	model	**complexity**	**Algo**	**parallel**
ideal	$\min_X \|M_r X M_c - \tilde{X}\|_F^2$ s.t: $X^\top \in span(Q_{k_c})$ $X \in span(P_{k_r})$	$\mathcal{O}(n^3)$	–	–
alter-nate	$\min_X \|M_r X M_c - \tilde{X}\|_F^2$ $+\gamma_c \operatorname{tr}(X\mathcal{L}_c X^\top)$ $+\gamma_r \operatorname{tr}(X^\top \mathcal{L}_r X)$	$\mathcal{O}(Inp\mathcal{K})$	gradient descent	no
approx-imate	$\min_U \|M_r U - \tilde{U}\|_F^2$ $+\gamma^{'}{}_r \operatorname{tr}(U^\top \mathcal{L}_r U)$	$\mathcal{O}(In\mathcal{K})$	gradient descent	yes
	$\min_V \|M_c^\top V - \tilde{V}\|_F^2$ $+\gamma^{'}{}_c \operatorname{tr}(V^\top \mathcal{L}_c V)$	$\mathcal{O}(Ip\mathcal{K})$	gradient descent	yes
	$X = U\tilde{\Sigma}V^\top$	$\mathcal{O}(\rho_r^2 \rho_c)$	SVD	
approx-imate 1	$\min_U \operatorname{tr}(U^\top \mathcal{L}_r U)$ s.t:$M_r U = \tilde{U}$	$\mathcal{O}(pkO_l\mathcal{K})$	Lancoz	
	$\min_V \operatorname{tr}(V^\top \mathcal{L}_c V)$ s.t:$M_c^\top V = \tilde{V}$	$\mathcal{O}(nkO_l\mathcal{K})$	Lancoz	yes
	$X = U\tilde{\Sigma}V^\top$	$\mathcal{O}(\rho_r^2 \rho_c)$	SVD	
approx-imate 2	$\min_U \operatorname{tr}(U^\top \mathcal{L}_r U)$ s.t:$M_r U = \tilde{U}$ $X = UU^\top Y$	$\mathcal{O}(pkO_l\mathcal{K})$	Lancoz	yes
approx-imate 3	$\min_V \operatorname{tr}(V^\top \mathcal{L}_c V)$ s.t:$M_c^\top V = \tilde{V}$ $X = YVV^\top$	$\mathcal{O}(nkO_l\mathcal{K})$	Lancoz	yes

Table 7. Computational complexity of all the models considered in this work

Model	**Complexity** G_c $\mathcal{O}(np\log(n))$	**Complexity** G_r $\mathcal{O}(np\log(p))$	**Complexity Algorithm** for $p \ll n$	**Overall Complexity (low-rank)** **Fast SVD** for $p \ll n$	**Decoder**	**Total**	**Overall Complexity (clustering)** **k-means** (Elkan, 2003)	**Decoder**	**Total**
LE (Belkin & Niyogi, 2003)	+	–	$\mathcal{O}(n^3)$	–	–	–	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(n^3)$
LLE (Roweis & Saul, 2000)	–	–	$\mathcal{O}((p+k)n^2)$	–	–	–	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(pn^2)$
PCA	–	–	$\mathcal{O}(p^2 n)$	–	–	$\mathcal{O}(n(p^2\log(n)+p^2))$	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(np^2\log(n)+np^2)$
GLPCA (Jiang et al., 2013)	+	–	$\mathcal{O}(n^3)$	–	–	$\mathcal{O}(n(p\log(n)+n^2))$	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(n^3))$
NMF (Lee & Seung, 1999)	–	–	$\mathcal{O}(Inpk)$	–	–	$\mathcal{O}(Inpk)$	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(Inp(k+K))$
GNMF (Cai et al., 2011)	+	–	$\mathcal{O}(Inpk)$	–	–	$\mathcal{O}(np\log(n))$	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(np\log(n))$
MMF (Zhang & Zhao, 2013)	+	–	$\mathcal{O}(((p+k)k^2+pk)I)$	–	–	$\mathcal{O}(np\log(n))$	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(np\log(n))$
RPCA (Candès et al., 2011)	–	–	$\mathcal{O}(Inp^2)$	–	–	$\mathcal{O}(np^2I+n\log(n))$	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(np^2I+n\log(n))$
RPCAG (Shahid et al., 2015a)	–	–	$\mathcal{O}(Inp^2)$	–	–	$\mathcal{O}(np^2I+n\log(n))$	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(np^2I+n\log(n))$
FRPCAG (Shahid et al., 2015b)	+	+	$\mathcal{O}(Inp\mathcal{K})$	–	–	$\mathcal{O}(np\log(n))$	$\mathcal{O}(Inkp)$	–	$\mathcal{O}(np\log(n))$
CPCA	+	+	$\mathcal{O}(I\rho_c\rho_r\mathcal{K})$	$\mathcal{O}(\rho_c\rho^2{}_r)$	$\mathcal{O}(nkO_l\mathcal{K})$	$\mathcal{O}(np\log(n))$	$\mathcal{O}(I\rho_c k\rho_r)$	$\mathcal{O}(nkO_l)$	$\mathcal{O}(np\log(n))$

Table 8. Details of the datasets used for clustering experiments in this work.

Dataset	Samples	Dimension	Classes
ORL large	400	56×46	40
ORL small	300	28×23	30
CMU PIE	1200	32×32	30
YALE	165	32×32	11
MNIST	70000	28×28	10
USPS large	10000	16×16	10
USPS small	3500	16×16	10

Table 9. Range of parameter values for each of the models considered in this work. k is the rank or dimension of subspace or the number of clusters, λ is the weight associated with the sparse term for Robust PCA framework (Candès et al., 2011) and γ, α are the parameters associated with the graph regularization term.

Model	Parameters	Parameter Range
LLE (Roweis & Saul, 2000), PCA LE (Belkin & Niyogi, 2003)	k	$k \in \{2^1, 2^2, \cdots, \min(n,p)\}$
GLPCA (Jiang et al., 2013)	k, γ	$k \in \{2^1, 2^2, \cdots, \min(n,p)\}$ $\gamma \implies \beta$ using (Jiang et al., 2013) $\beta \in \{0.1, 0.2, \cdots, 0.9\}$
MMF (Zhang & Zhao, 2013)	k, γ	$k \in \{2^1, 2^2, \cdots, \min(n,p)\}$ $\gamma \in \{2^{-3}, 2^{-2}, \cdots, 2^{10}\}$
NMF (Lee & Seung, 1999)	k	
GNMF (Cai et al., 2011)	k, γ	
RPCA (Candès et al., 2011)	λ	$\lambda \in \{\frac{2^{-3}}{\sqrt{\max(n,p)}} : 0.1 : \frac{2^3}{\sqrt{\max(n,p)}}\}$ $\gamma \in \{2^{-3}, 2^{-2}, \cdots, 2^{10}\}$
RPCAG (Shahid et al., 2015a)	λ, γ	
FRPCAG (Shahid et al., 2015b)	γ_r, γ_c	$\gamma_r, \gamma_c \in (0, 30)$
CPCA	γ_r, γ_c	$\gamma_r, \gamma_c \in (0, 30)$
	k (approximate decoder)	$\tilde{\Sigma}_{k,k}/\tilde{\Sigma}_{1,1} < 0.1$

Table 10. ORL dataset

Data set	Model	no noise	Gaussian noise			
			5%	10%	15%	20%
ORL small	k-means	0.40	0.43	0.43	0.45	0.44
	LLE	0.26	0.26	0.26	0.26	0.19
	LE	0.21	0.18	0.19	0.19	0.19
	PCA	0.30	0.30	0.32	0.34	0.32
	MMF	0.21	0.20	0.18	0.18	0.17
	GLPCA	**0.14**	**0.13**	**0.13**	**0.13**	**0.14**
	NMF	0.31	0.34	0.29	0.31	0.34
	GNMF	0.29	0.29	0.29	0.31	0.29
	RPCA	0.36	0.34	0.33	0.35	0.36
	RPCAG	0.17	0.17	0.17	0.16	0.16
	FRPCAG	0.15	0.15	0.15	0.14	0.16
	CPCA (2,2)	0.23	0.23	0.23	0.24	0.25
	CPCA (2,1)	0.17	0.17	0.17	0.14	0.17
ORL large	k-means	0.49	0.50	0.51	0.51	0.51
	LLE	0.28	0.27	0.27	0.24	0.25
	LE	0.24	0.25	0.25	0.24	0.25
	PCA	0.35	0.34	0.35	0.36	0.36
	MMF	0.23	0.23	0.23	0.24	0.24
	GLPCA	0.18	0.18	0.18	0.19	0.19
	NMF	0.36	0.33	0.36	0.32	0.36
	GNMF	0.34	0.37	0.36	0.39	0.39
	FRPCAG	**0.17**	**0.17**	**0.17**	**0.17**	**0.17**
	CPCA (2,2)	0.21	0.21	0.21	0.23	0.22

Table 11. CMUPIE & YALE datasets

Data set	Model	no noise	Gaussian noise			
			5%	10%	15%	20%
CMUPIE	k-means	0.76	0.76	0.76	0.75	0.76
	LLE	0.47	0.50	0.52	0.50	0.55
	LE	0.60	0.58	0.59	0.58	0.60
	PCA	0.27	0.27	0.27	0.27	0.29
	MMF	0.67	0.67	0.67	0.66	0.67
	GLPCA	0.37	0.39	0.37	0.38	0.38
	NMF	0.24	0.27	**0.24**	0.25	0.27
	GNMF	0.58	0.59	0.56	0.58	0.59
	RPCA	0.39	0.38	0.41	0.41	0.38
	RPCAG	0.24	**0.24**	**0.24**	**0.24**	0.25
	FRPCAG	**0.23**	**0.24**	**0.24**	**0.24**	**0.24**
	CPCA (2,2)	0.26	0.26	0.26	0.28	0.27
	CPCA (2,1)	0.28	0.29	0.29	0.30	0.30
YALE	k-means	0.76	0.76	0.76	0.76	0.77
	LLE	0.51	0.47	0.46	0.49	0.50
	LE	0.52	0.56	0.55	0.54	0.54
	PCA	0.53	0.52	0.53	0.56	0.55
	MMF	0.58	0.59	0.56	0.57	0.58
	GLPCA	0.47	0.46	0.47	0.45	0.48
	NMF	0.57	0.58	0.59	0.57	0.59
	GNMF	0.59	0.59	0.61	0.59	0.60
	RPCA	0.45	0.48	0.46	0.46	0.48
	RPCAG	**0.40**	0.40	0.41	0.41	0.41
	FRPCAG	**0.40**	**0.37**	**0.40**	**0.40**	**0.40**
	CPCA (2,2)	0.43	0.45	0.42	0.46	0.43

Table 12. Preservation of the rank of the datasets in the compressed low-rank $\tilde{X}$ determined by solving FRPCAG (1). For this experiment, we take different datasets and corrupt them with different types of noise and perform cross-validation for clustering using the parameter range for CPCA mentioned in Table 9. Then we report the rank of $\tilde{X}$ for the parameter corresponding to the minimum clustering error. As $\tilde{X}$ is approximately low-rank so we use the following strategy to determine the rank k: $\tilde{\Sigma}_{k,k}/\tilde{\Sigma}_{1,1} < 0.1$. FRPCAG assumes that the number of clusters $\approx$ rank. Our findings show that this assumption is almost satisfied for the sampled matrices even in the presence of various types of noise. Thus, the rank is preserved under the proposed sampling strategy.

Dataset	**downsampling factor**	**actual rank**	**Rank after FRPCAG on sampled matrix**					
	$(columns, rows)$		**Gaussian noise**			**Laplacian noise**		
			5%	**10%**	**15%**	**5%**	**10%**	**15%**
ORL large	(2,1)	40	41	41	41	41	41	42
USPS large	(10,2)	10	10	11	11	11	11	11
MNIST	(10,2)	10	11	11	11	11	11	11
CMU PIE	(2,1)	30	31	31	31	31	31	32
YALE	(2,1)	11	11	11	11	11	11	11

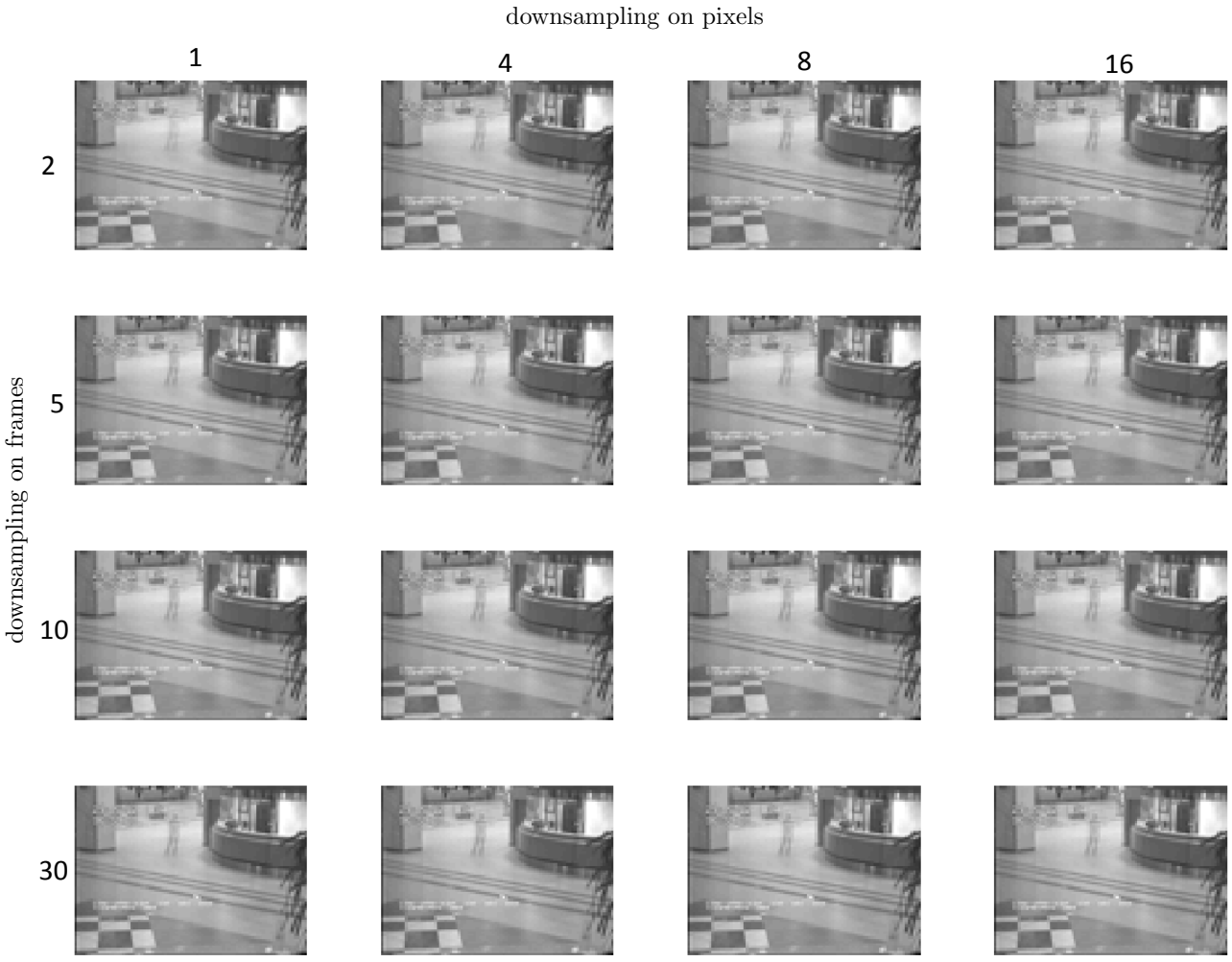


Table 13. A comparison of the quality of low-rank for the shopping mall video (1st row of Fig. 1a) extracted using the approximate decoder (16) for different downsampling factors on the pixels and frames. It is obvious that the quality of low-rank remains intact even with higher downsampling factors.