Data Recovery and Subspace Clustering from Quantized and Corrupted Measurements

Ren Wang, *Student Member, IEEE,* Meng Wang, *Member, IEEE*, Jinjun Xiong

***Abstract*—Quantized low-rank matrix recovery estimates the original matrix from its entry-wise quantized measurements. Subspace clustering divides data points belonging to the union of subspaces (UoS) into the respective subspaces. Generalizing from both quantized matrix recovery and subspace clustering, this paper for the first time studies the problem of combined data recovery and subspace clustering based on the quantized measurements of data points following the UoS model. The recovery and clustering is achieved simultaneously by solving a nonconvex constrained maximum likelihood problem. The relative recovery error is proved to diminish to zero as the matrix size increases. A Sparse Alternative Proximal Algorithm (Sparse-APA) with a convergence guarantee is proposed to solve the nonconvex problem. The proposed method is evaluated numerically on synthetic and Extended Yale Face B Datasets.**

***Index Terms*—quantization, union of subspaces, subspace clustering, matrix recovery.**

I. Introduction

In applications like image denoising [2], sensor network location [3], and collaborative filtering [12], the obtained measurements are not only noisy but discretized to binary or multi-values. For example, some images have low resolutions. The user responses and ratings are selected from a few values. Moreover, adding noise and then applying quantization to the raw data can mask the information and enhance the data privacy of individual users. This data privacy enhancement through quantizing measurements have been exploited in sensor networks [49] and smart meters [43], and synchrophasor data management in power systems [19].

The problem of recovering the original data from quantized measurements has been studied under the assumption that the ground-truth data before quantization can be modeled by a low-rank matrix [4], [8], [9], [12], [17], [25], [27]–[29]. A matrix $M \in \mathbb{R}^{m\times n}$ is low-rank if its rank r is much less than m and n. Refs. [19], [28] further consider the case that the data matrix before quantization is the sum of a low-rank matrix and a sparse matrix, where the sparse matrix models the significant errors in the measurements. Ref. [19] shows that when the number of corruptions per column is bounded, the relative recovery error is $\mathcal{O}(\sqrt{\frac{r}{\min(m,n)}})$, which diminishes to zero asymptotically when the matrix dimensions m and n both increase to infinity.

In many applications, the data matrix is no longer low-rank, but each column of the matrix belongs to one of p low-dimension subspaces. This so-called Union-of-Subspaces (UoS) model is a generalization of low-rank matrices. For example, feature trajectories of a rigidly moving object in a video [33], [47], face images of a subject under varying illumination [1], and time series by sensors measuring the same event [17] all belong to a low-dimensional subspace of the ambient space. Subspace clustering [13] groups the data points in the UoS model to their respective subspaces and finds applications in anomaly detection and localization [17], [52], image classification [10], [26], social network analysis [51] computer vision [31], [48]. Since data in a subspace are often distributed arbitrarily, standard clustering methods such as k-means [21] that rely on the spatial proximity of the data in each cluster do not provide meaningful clusters. Among various subspace clustering methods (e.g., [13], [14], [32], [34], [40], [42], [45], [46]), the spectral-based methods [13], [32], [46] have high and provable performances. These methods find a sparse representation of each data point using other points in the same subspace. Spectral clustering [37] is then applied to the similarity graph built on the sparse coefficients to cluster the data. The computation of the sparse coefficients usually requires solving convex optimization problems [13], [32] and is time-consuming for large datasets. Moreover, the existing subspace clustering methods do not consider quantization errors and degrade significantly when the measurements are highly quantized. Since quantized measurements can be viewed as nonlinear functions of the ground-truth data, clustering from quantized measurements is related to a recent line of work that consider nonlinear measurements following the assumption that after a nonlinear mapping from the measurements to a high-dimensional space, the resulting lifted points belong to the UoS. The nonlinear mapping is characterized through kernels [23], [39] or learned from deep learning [24], [41]. The kernel approach requires the selection of appropriate kernels, and the deep-learning approach requires a very large training set. Moreover, these clustering methods have no analytical guarantees.

This paper studies the problem of data recovery and data clustering from quantized and partially corrupted measurements when the data satisfies the UoS model. It for the first time connects quantized matrix recovery and subspace clustering. On the one hand, it generalizes quantized matrix recovery from low-rank matrices to the UoS model. On the other hand, it extends subspace clustering to the cases with highly quantized measurements. This paper proposes to solve a nonconvex constrained maximum log-likelihood problem so as to recover the data and cluster the data points into respective subspaces simultaneously. The recovery error of

R. Wang and M. Wang are with the Dept. of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY. Email: {wangr8, wangm7}@rpi.edu. J. Xiong is with IBM Thomas J. Watson Research Center, Yorktown Heights, NY, Email: jinjun@us.ibm.com.

our proposed data recovery method is $\mathcal{O}(\sqrt{\frac{d}{\min(m,n)}})$, where d is the dimension of each subspace. When subspaces are independent, this error bound is reduced by a factor of $\sqrt{1/p}$ compared with the result by directly applying the existing methods for quantized low-rank matrix recovery, where p is the number of subspaces. A sparse alternative proximal algorithm (Sparse-APA) is developed to solve the nonconvex problem approximately. Every iterates generated by Sparse-APA is proved to converge to a critical point of the nonconvex problem.

The rest of the paper is organized as follows. The problem formulation and related work are introduced in Section II. Section III presents the theoretical analyses of our proposed data recovery method. Section IV introduces the Sparse-APA algorithm with its convergence analysis. Section VI records the numerical experiments. Section VII concludes the paper. Supporting lemmas are deferred to the Appendix.

II. Problem Formulation and Related Work

A. Mathematical model

Let $L^* \in \mathbb{R}^{m\times n}$ contain the actual data before quantization. The column vectors of L^* are in $\mathbb{R}^m$, and each vector belongs to one of p different d-dimensional subspaces in $\mathbb{R}^m$ ($d \ll m$). Let r denote the rank of L^*, then $r \le pd$. Let $[n]$ denote the set $\{1, ..., n\}$. Let S_i ($i \in [p]$) denote the ith subspace, and let L_i^* denote the submatrix of L^* that contains all the columns in L^* that belong to S_i. Let n_i denote the number of columns in L_i^*. We assume $m \le n_i \le \xi n/p$ for all i and some positive constant ξ. That means the number of points in each subspace is larger than the ambient dimension m, and each subspace has at most of a constant fraction of the columns in L^*.

Then there exists a coefficient matrix $C^* \in \mathbb{R}^{n\times n}$ such that $L^* = L^*C^*$, $C_{ii} = 0$ for all $i \in [n]$, and C_{ij}^* is zero if the ith column and the j column of L^* do not belong to the same subspace. Let c_j^* denote the jth column of C^*. Since the dimension is d for every S_i, c_j^* contains at most d nonzero entries for all $j \in [n]$. Following the terminologies in [13] one can say that L^* satisfies the *self-expressive property*, and C^* satisfies the *subspace-preserving property*. We summarize them as follows.

Definition 1. *[13] A matrix $L \in \mathbb{R}^{m\times n}$ has the* **self-expressive property** *if $L = LC$ for some $C \in \mathbb{R}^{n\times n}$, and $C_{ii} = 0$ for all $i \in [n]$. Moreover, C has the* **subspace-preserving property** *of L if $C_{ij} = 0$ for columns i and j of L belonging to different subspaces.*

Let $E^* \in \mathbb{R}^{m\times n}$ contain additive errors. E^* is sparse with the number of nonzero entries s much smaller than mn. The partially corrupted measurements can be represented by $X^* = L^* + E^*$. We assume $\|L\|_\infty \le \alpha_1$ and $\|E\|_\infty \le \alpha_2$, where the infinity norm $\|\cdot\|_\infty$ measures the maximum absolute value. Let $N \in \mathbb{R}^{m\times n}$ denote the noise matrix with i.i.d. entries drawn from a known cumulative distribution function $\Phi(x)$. Two common choices of $\Phi(x)$ are (i) Probit model with $\Phi(x) = \Phi_{\text{norm}}(x/\sigma)$, $\sigma > 0$, where Φ_{norm} is the cumulative distribution function of the standard normal distribution $\mathcal{N}(0,1)$; (ii) Logistic model with $\Phi(x) = \Phi_{log}(x/\sigma) = \frac{1}{1+e^{-x/\sigma}}$.

The quantized operator $\mathcal{Q}$ maps a real number to one of the K labels. Given the quantization boundaries $\omega_0 < \omega_1 < ... < \omega_K$, we have

$$\mathcal{Q}(x) = l \quad \text{if } \omega_{l-1} < x \le \omega_l,\ l \in [K]. \tag{1}$$

A K-level noisy and quantized measurement Y_{ij} based on L_{ij}^* satisfies

$$Y_{ij} = \mathcal{Q}(L_{ij}^* + E_{ij}^* + N_{ij}),\ \forall (i,j). \tag{2}$$

One can check that

$$Y_{ij} = l \text{ with probability } f_l(X_{ij}^*),\ \forall (i,j), \tag{3}$$

where $\sum_{l=1}^K f_l(X_{ij}^*) = 1$, and

$$f_l(X_{ij}^*) = P(Y_{ij} = l | X_{ij}^*) = \Phi(\omega_l - X_{ij}^*) - \Phi(\omega_{l-1} - X_{ij}^*). \tag{4}$$

The process is visualized in Fig. 1.

We assume that the selected cumulative distribution function Φ is monotonously increasing. Since $|X_{ij}^*| = |L_{ij}^* + E_{ij}^*| \le \alpha_1 + \alpha_2$, $\Phi(\omega_l - X_{ij}^*) \ge \Phi(\omega_{l-1} - X_{ij}^*) + \beta$ holds, where β is a positive number depending on quantization boundaries and the range of X^*. Then, $1 \ge f_l \ge \beta > 0$.

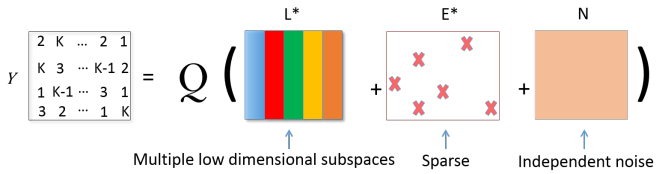


Fig. 1. Quantized measurements with corruptions.

This paper solves the following problem:
(P1) *Given observations Y, known boundaries $\omega_0 < \omega_1 < ... < \omega_K$ and noise distribution Φ, can we recover L^* and cluster the data into the corresponding subspaces S_i's simultaneously?*

B. Applications of the proposed problem

(P1) finds applications in different domains.

Image recovery and image clustering. Image recovery from low-resolution measurements is a long-standing problem, where the low resolution can result from either sensor qualities or communication constraints. One important problem in computer vision is to separate the images of different subjects from a mixture of images. The images of the same person with varying illumination belong to the same low-dimensional subspace [1]. Every column of L^* represents one vectorized image of a person, and L^* contains images from p persons under different illuminations. E^* represents significant measurement errors.

Video recovery and motion segmentation. Motion segmentation segments a video sequence of multiple moving objects into multiple spatial-temporal regions, each of which corresponds to a motion in the scene [13]. Each row of L^* corresponds one frame of the video. Each column of L^*

corresponds to the trajectory of a feature point. The feature trajectories of the same motion belong to a low-dimensional subspace. E^* corresponds to measurement errors, and Y contains the low-resolution measurements.

Event location in power systems. Each column of L^* corresponds to the voltage or current phasor measurements collected by a Phasor Measurement Unit (PMU). Time series provided by PMUs at different locations belong to the same low-dimensional subspace if these measurements are affected by the same event [17], [18]. The clustering results can be used to locate the impact regions of different events. Since the PMU data are transmitted from PMUs across the system to the operator, Ref. [19] proposes to add noise and apply quantization to enhance the data privacy of the measurements and reduce the communication rate. (P1) models the operator's problem of data recovery and event location.

C. Related work

When $p = 1$, L^* reduces to a low-rank matrix. Our problem reduces to the problem of low-rank matrix recovery from quantized measurements, which has been studied by [4], [8], [9], [12], [17], [25], [27]–[29], with motivating applications in collaborative filtering [12], image processing [2], and sensor networks [3]. The above references assume E^* is zero, while Refs. [19], [28] further consider the case that E^* is nonzero, i.e., the data matrix before quantization is partially corrupted.

When no quantization is applied, our problem reduces to the subspace clustering problem that has been studied in [13], [14], [32], [40], [42], [45], [46]. These methods first solve convex optimization problems to compute C^* and then apply spectral clustering [37] to the similarity graph built based on C^*. If C^* is subspace preserving, it is possible to correctly cluster the data points. Specifically, two subspaces S_i and S_j ($i \neq j$) are *independent* of each other if S_i and S_j intersects only at $\mathbf{0}$. We say that L^* is *subspace independent* if for every $i \in [p]$, S_i is independent of the sum of all S_j for $j \neq i$, $j \in [p]$. This indicates that a data point can only be linearly expressed by data points from the same subspace. If every point in a subspace-independent dataset L^* is directly measured, one can estimate a subspace-preserving coefficient matrix C^* by minimizing the ℓ_0-norm of all matrices C such that $L^* = L^*C$ [13]. Since ℓ_0-norm is nonconvex, Ref. [13] replaces ℓ_0-norm with the convex ℓ_1-norm, and the resulting sparse subspace clustering (SSC) method computes the coefficient matrix by solving

$$\min_{C \in \mathbb{R}^{n \times n}} \|C\|_1 \quad \text{s.t. } L^* = L^*C \tag{5}$$

and applies spectral clustering [37] on to the solution to (5) to obtain the clustering results. Solving large-scale convex optimization problems is computationally expensive. Moreover, these clustering methods do not consider quantization errors and perform poorly with highly quantized measurements.

This paper connects quantized data recovery and subspace clustering for the first time. It develops the first unified approach that recovers and clusters the data simultaneously. Our method outperforms a naive approach of first recovering the data from quantized measurements and then applying SSC methods.

III. Quantized Matrix Recovery and Subspace Clustering

To solve (P1), we propose to estimate L^*, C^*, and E^* by the solution $(\hat{L}, \hat{E}, \hat{C})$ to the following nonconvex optimization problem,

$$\min_{L,E \in \mathbb{R}^{m \times n}, C \in \mathbb{R}^{n \times n}} F(L, E) \quad \text{s.t. } (L, E, C) \in \mathcal{S}_f, \tag{6}$$

where

$$F(L, E) = -\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{K} \mathbf{1}_{[Y_{ij}=l]} \log(f_l(L_{ij} + E_{ij})), \tag{7}$$

$$\begin{aligned} \mathcal{S}_f = \{(L, E, C) : L = LC, \text{rank}(L) \leq r, \|L\|_\infty \leq \alpha_1, \\ \|E\|_\infty \leq \alpha_2, \|E\|_0 \leq s, \|c_i\|_0 \leq d, C_{ii} = 0, \forall i \in [n]\}. \end{aligned} \tag{8}$$

$\|\cdot\|_0$ measures the number of nonzero entries of a vector or matrix. c_i is the ith column of the coefficient matrix C. $\mathbf{1}_{[A]}$ denotes the indicator function that takes value 1 if A is true and value 0 otherwise.

(7) is a constrained maximum log-likelihood estimation problem. The feasible set $\mathcal{S}_f$ includes all self-expressive matrices L that are at most rank r and the corresponding subspace-preserving matrix C with at most d nonzero entries per column. We impose the constraint of $\|c_i\|_0 \leq d$ because with all subspaces S_i's being d-dimensional, each point can be presented by a linear combination of at most d points in the same subspace. The error matrix E contains at most s nonzeros. L and E have bounded infinity norms. Among all matrices in the feasible set $\mathcal{S}_f$, (7) returns a self-expressive $\hat{L}$ with the corresponding subspace-preserving $\hat{C}$ and the estimated error matrix $\hat{E}$ such that with $\hat{L}$ and $\hat{E}$, the likelihood of obtaining Y is maximized. Then one can apply spectral clustering [37] to $\hat{C}$ to separate data points into different clusters. Thus, the data recovery and subspace clustering are achieved simultaneously.

Note that (6) is nonconvex due to the nonconvexity of the feasible set (8). We first analyze the recovery and clustering performance assuming that a solver for (6) exists. We defer the algorithm to solve (6) in Section IV.

One side remark is that in the special case that the measurements do not contain significant errors, one can drop E in the objective function and the constraints and solve the resulting simplified problem:

$$\min_{(L,C) \in \mathcal{S}_f} F(L) = -\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{K} \mathbf{1}_{[Y_{ij}=l]} \log(f_l(L_{ij})), \tag{9}$$

$$\begin{aligned} \mathcal{S}_f = \{(L, C) : L = LC, \text{rank}(L) \leq r, \|L\|_\infty \leq \alpha_1, \\ \|c_i\|_0 \leq d, C_{ii} = 0, \forall i \in [n]\}, \end{aligned} \tag{10}$$

The above recovery and clustering problem from quantized measurements has not been addressed before, even without corruptions. Here we focus on the general problem (6), and the results are applicable to the special case (9), simply by setting s to zero in the analysis and the algorithm.

We first define two constants γ_α and L_α needed for the recovery analysis,

$$\gamma_\alpha = \min_{l \in [K]} \inf_{|x| \leq \alpha_1 + \alpha_2} \{\frac{\dot{f}_l^2(x)}{f_l^2(x)} - \frac{\ddot{f}_l(x)}{f_l(x)}\}, \tag{11}$$

$$L_\alpha = \max_{l\in[K]} \sup_{|x|\le\alpha_1+\alpha_2} \{|\dot{f}_l(x)|/f_l(x)\}, \tag{12}$$

where $\dot{f}_l(x)$ and $\ddot{f}_l(x)$ are the first and second order derivatives with respect to x. These definitions are proposed in [4]. γ_α and L_α depend on α, Φ, and ω_i's ($i \in [K]$) but are independent of m and n. Since $f_l(x)$ is log-concave if and only if $(\dot{f}_l(x))^2 \ge \ddot{f}_l(x) f_l(x)$ [7], $\gamma_\alpha \ge 0$ for a log-concave f_l and $\gamma_\alpha > 0$ for a strictly log-concave f_l. One can check that $\gamma_\alpha > 0$ for logistic and Gaussian noises [2].

The objective includes both data clustering and data recovery. Since the clustering performance and the recovery performance are coupled with each other, we analyze the data recovery performance assuming that the clustering is not "arbitrarily bad." Roughly speaking, we assume that the recovered data $\hat{L}$ contains points from p subspaces, but the division of data points can be different from that of L^*. To formalize the result, we need the following assumption:

Assumption 1: Columns of $\hat{L}$ belong to p subspaces, each of which has a dimension smaller or equal to d. The number of columns in each subspace is lower and upper bounded by m and $\xi_2 n/p$, where ξ_2 is a constant smaller than p. We then have $\xi \ge \max(\xi_1, \xi_2)$.

One sufficient condition for the first sentence of Assumption 1 to hold is that $\hat{C}$ is a p-block diagonal matrix. Here, a matrix is called p-block diagonal if by only permuting rows, it can be transformed to a new matrix with p square submatrices along its diagonal. All elements are zero except the elements in those square submatrices [15]. This indicates that some columns of $\hat{L}$ can be represented by each other if their column indices correspond to row indices of nonzero entries in the same block of $\hat{C}$. Note that $\|c_i\|_0 \le d$ from the feasibility constraints. Then one column of $\hat{L}$ can be represented by at most d other columns corresponding to the same block. Therefore, data points in $\hat{L}$ belong to p subspaces with dimension of each subspace smaller or equal to d. For noiseless data belonging to independent subspaces, [36] introduces sufficient conditions for the block diagonal assumption. [35] proposes a regularizer to the objective function to promote a solution with a block diagonal structure.

Theorem 1. *If Assumption 1 holds, with probability at least* $1 - C_1 e^{-2C_2\xi n/p}$*, any global minimizer* $(\hat{L}, \hat{E}, \hat{C})$ *of (6) satisfies*

$$\frac{\|(\hat{L}+\hat{E}) - (L^*+E^*)\|_F}{\sqrt{mn}} \le \min(2\alpha_1 + 2\alpha_2\sqrt{\frac{s}{mn}}, U_\alpha), \tag{13}$$

for some positive constants C_1 *and* C_2*. where*

$$U_\alpha = \max(\frac{16.08 L_\alpha\sqrt{\xi d}}{\gamma_\alpha\sqrt{m}}, \sqrt{\frac{32.16\alpha_2 L_\alpha\sqrt{\xi dns} + 8\alpha_2 s L_\alpha}{\gamma_\alpha mn}}). \tag{14}$$

Furthermore,

$$\frac{\|\hat{L} - L^*\|_F}{\sqrt{mn}} \le \min(2\alpha_1 + 2\alpha_2\sqrt{\frac{s}{mn}}, U_\alpha + 2\alpha_2\sqrt{\frac{s}{mn}}) \tag{15}$$

holds with the same probability.

Proof: The proof follows the same line as the proofs of Theorem 3.1 in [4] and Theorem 1 in [19] that assume L^* is low rank. Ref. [4] considers the cases with no corruptions. Ref. [19] extends the analysis to cases with corruptions. Here we extend the analysis from low-rank matrices to L^* with columns in p low-dimensional subspaces.

The first bound $2\alpha_1 + 2\alpha_2\sqrt{\frac{s}{mn}}$ in (13) follows from the fact that $\hat{L}$, L^*, $\hat{E}$, $E^* \in \mathcal{S}_f$. We discuss the second bound in (13) as follows. Note that the set $\mathcal{S}_f$ is compact, and the objective function $F(X)$ is continuous in X. $F(X)$ then achieves a minimum in $\mathcal{S}_f$. Suppose that $\hat{X} \in \mathcal{S}_f$ minimizes $F(X)$.

Let $\theta = \mathrm{vec}(X) \in \mathbb{R}^{mn}$ and $\mathcal{F}_Y(\theta) = F(X)$. By the second-order Taylor's theorem, we have

$$\begin{aligned}\mathcal{F}_Y(\theta) &= \mathcal{F}_Y(\theta^*) + \langle \nabla_\theta \mathcal{F}_Y(\theta^*), \theta - \theta^* \rangle \\ &+ \frac{1}{2}\left\langle \theta - \theta^*, (\nabla^2_{\theta\theta}\mathcal{F}_Y(\tilde{\theta}))(\theta - \theta^*) \right\rangle,\end{aligned} \tag{16}$$

where $\tilde{\theta} = \theta^* + \eta(\theta - \theta^*)$ for some $\eta \in [0,1]$, with corresponding matrices $\tilde{X} = X^* + \eta(X - X^*)$.

Combining (16), Lemma 2 and Lemma 3, we have that

$$\begin{aligned}F(\hat{X}) \ge F(X^*) &- 4.02 L_\alpha\sqrt{\xi dn}\|\hat{X} - X^*\|_F \\ &- 8.04\alpha_2 L_\alpha\sqrt{\xi dns} - 2\alpha_2 s L_\alpha + \frac{\gamma_\alpha}{2}\|\hat{X} - X^*\|_F^2\end{aligned} \tag{17}$$

holds with probability at least $1 - C_1 e^{-2C_2\xi n/p}$. Note that $F(\hat{X}) \le F(X^*)$. Thus

$$\begin{aligned}\frac{\gamma_\alpha}{2}\|\hat{X} - X^*\|_F^2 \le &4.02 L_\alpha\sqrt{\xi dn}\|\hat{X} - X^*\|_F \\ &+ 8.04\alpha_2 L_\alpha\sqrt{\xi dns} + 2\alpha_2 s L_\alpha\end{aligned} \tag{18}$$

holds with probability at least $1 - C_1 e^{-2C_2\xi n/p}$. From $x < a + b \le \max(2a, 2b)$, it holds with the same probability that

$$\|\hat{X} - X^*\|_F/\sqrt{mn} \le U_\alpha. \tag{19}$$

■

Theorem 1 provides the upper bound of the recovery error when partial measurements are corrupted. We interpret the significance of Theorem 1 from the following aspects.

(1) Correction of corrupted measurements. In the special case that there is no corruption, the error bound is

$$\frac{\|\hat{L} - L^*\|_F}{\sqrt{mn}} \le \min(2\alpha_1, \frac{8.04 L_\alpha\sqrt{\xi d}}{\gamma_\alpha\sqrt{m}}), \tag{20}$$

which is in the order of $\mathcal{O}(\sqrt{\frac{d}{m}})$. Moreover, as long as the number s of corrupted measurements is at most in the order of $\Theta(n)$, i.e., the number of corrupted data per column is bounded, (15) indicates that the recovery error is still in the order of $\mathcal{O}(\sqrt{\frac{d}{m}})$,

$$\|(\hat{L}+\hat{E}) - (L^*+E^*)\|_F/\sqrt{mn} \le \mathcal{O}(\sqrt{\frac{d}{m}}), \tag{21}$$

$$\text{and } \|\hat{L} - L^*\|_F/\sqrt{mn} \le \mathcal{O}(\sqrt{\frac{d}{m}}). \tag{22}$$

Comparing (20) and (22), one can see that our method can handle corruptions such that the recovery error is in the same order as that of noncorrupted measurements.

(2) Asymptotic recovery of the actual data. Since $\sqrt{\frac{d}{m}}$ decreases to 0 when m increases to infinity, the left-hand side

of (22) decreases to 0. Note that $\|L^*\|_F$ is in the order of $\sqrt{mn}$ when the matrix dimension increase. Thus, when both m and n approach infinity, the relative error between $\hat{L}$ and L^* decreases to 0. Thus, $\hat{L}$ is sufficiently close to L^* when the matrix dimension is large enough.

(3) Recovery enhancement over low-rank approaches. Data recovery from quantized and partially corrupted measurements has been studied in [19], assuming that the rank of the ground-truth matrix L^* is much less than m and n. Ref. [19] shows that if s is at most $\Theta(n)$, the relative recovery error is $\mathcal{O}(\sqrt{\frac{r}{m}})$. Moreover, this error bound is order-wise optimal in the sense that for any recovery method, there always exists at least one rank-r matrix L such that the relative recovery error of L is at least $\Theta(\sqrt{\frac{r}{m}})$.

Focusing on L^* that contains points from p d-dimensional subspaces, Theorem 1 shows that we can reduce the relative recovery error to $\mathcal{O}(\sqrt{\frac{d}{m}})$. Note that the rank r can be as large as pd if L^* is subspace independent. If one directly apply the method in [19] to our problem, the relative recovery error can be $\mathcal{O}(\sqrt{\frac{pd}{m}})$ for subspace-independent datasets. That indicates that our method can reduce the recovery error by a factor of $\sqrt{1/p}$ compared with directly applying the existing quantized matrix recovery method. This bound does not contradict the informational theoretical bound in [19] as we only consider rank-r matrices that contain points from p d-dimensional subspaces instead of general rank-r matrices. Moreover, our method does not need the low-rank assumption which is needed in [19]. In fact, if p is very large, L^* can be full-rank. p does not affect the bound of the recovery error, while p affects the failure probability of $C_1 \exp(-2C_2\epsilon n/p)$. As long as p is $O(n^\alpha)$ for any $\alpha < 1$, the failure probability decays to zero as n increases to infinity.

The clustering performance is analyzed through the subspace-preserving property of the coefficient matrix in the literature, see e.g., [13], [22], [46]. If $\hat{C}$ satisfies the subspace-preserving property, the coefficients between any pair of points that are not in the same subspace are all zero. One can leverage this property to separate the points accurately. Note that even when the subspace-preserving property is not met, i.e., there exists a nonzero $\hat{C}_{ij}$ for columns i and j in different clusters, it is still possible to obtain the correct clustering results using methods like spectral clustering [45]. Details of spectral clustering [37] are introduced in Section IV. Intuitively, since C^* is subspace-preserving, when $\hat{L}$ is sufficiently close to L^*, $\hat{C}$ should have the subspace-preserving property.

We next provide a sufficient condition for $\hat{C}$ obtained from (6) to be subspace-preserving in Proposition 1. We say a set of points in a d-dimensional affine space is in *general position* if every subset of d or fewer data points is linearly independent [11].

Proposition 1. *If $\hat{L}$ contains points in general positions from p independent subspaces, denoted by $\hat{S}_i$ ($i \in [p]$), and the division of groups of $\hat{L}$ is the same as that of L^*, then the global minimizer $\hat{C}$ of (6) has subspace-preserving property of L^*.*

Proof: Given any i, from (8), we know that $\hat{l}_i = \hat{L}\hat{c}_i$, where $\hat{l}_i$ is the i-th column of $\hat{L}$, and $\hat{c}_i$ is the i-th column of $\hat{C}$. We also have $l_i^* = L^* c_i^*$, where l_i^* is the i-th column of L^*, and c_i^* is the i-th column of C^*. Without loss of generality, assume $\hat{l}_i \in \hat{S}_1$. We only need to prove that $\hat{c}_i$ is supported at the locations corresponding to data points belonging to $\hat{S}_1$.

Since C^* is subspace-preserving, and the division of groups of $\hat{L}$ is the same as that of L^*, c_i^* is supported at the locations corresponding to data points belonging to $\hat{S}_1$ in $\hat{L}$. Define $h = c_i^* - \hat{c}_i$, then h can be divided into two parts: $h = h_\Lambda + h_{\bar{\Lambda}}$, where the vector $h_\Lambda \in \mathbb{R}^n$ is supported at the locations corresponding to data points belonging to $\hat{S}_1$, and the vector $h_{\bar{\Lambda}} \in \mathbb{R}^n$ is supported at the locations corresponding to data points belonging to $\{\hat{S}_j\}_{j=2}^p$. If $h_{\bar{\Lambda}} = 0$, the claim holds trivially. We next consider $h_{\bar{\Lambda}} \neq 0$. We have $\hat{l}_i = \hat{L}(c_i^* - h) = \hat{L}(c_i^* - h_\Lambda) - \hat{L}h_{\bar{\Lambda}}$. Note that $\hat{L}(c_i^* - h_\Lambda) \in \hat{S}_1$, $\hat{L}h_{\bar{\Lambda}} \notin \hat{S}_1$, and $\hat{l}_i \in \hat{S}_1$. Then we must have $\hat{L}h_{\bar{\Lambda}} = 0$ and $\hat{l}_i = \hat{L}(c_i^* - h_\Lambda)$. From the assumption that data in each subspace are in general position, we know that $c_i^* - h_\Lambda$ has d nonzero entries. Thus, $\hat{c}_i = c_i^* - h_\Lambda - h_{\bar{\Lambda}}$ has number of nonzero entries larger than d, which contradicts with the constraint in (8). Thus, $h_{\bar{\Lambda}} = 0$, and the claim holds. ∎

Note that Proposition 1 provides a sufficient but not necessary condition for $\hat{C}$ to be subspace-preserving. We will also show numerically that it is possible to obtain correct clustering using spectral clustering even when $\hat{C}$ is not subspace-preserving in Section V.

We finally remark that the assumptions in Theorem 1 and Proposition 1 are only introduced to simplify the theoretical analyses, while our method applies to other UoS datasets even if these assumptions do not hold.

IV. Sparse Alternative Proximal Algorithm for Data Recovery and Clustering

Here we develop a fast algorithm to solve the nonconvex problem (6) with the convergence analysis. Since L is at most rank r, we decompose L as $L = UV^T$, where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$. We simplify our algorithm by replacing $L = LC$ using $V^T = V^T C$. Then we change the constraint $L = LC$ into a penalty function $\|V^T - V^T C\|_F^2$ in the objective. Note that $V^T = V^T C$ and $L = LC$ are equivalent only when U is rank r. In general, $V^T = V^T C$ implies that $L = LC$ but not vice versa. The revised problem of (6) is written as follows:

$$\begin{aligned}&(\hat{U}, \hat{V}, \hat{L}, \hat{E}, \hat{C})\\ &= \underset{\substack{U\in\mathbb{R}^{m\times r}, V\in\mathbb{R}^{n\times r}\\ L,E,C\in\mathbb{R}^{m\times n}}}{\arg\min} H(U,V,L,E,C) \quad \text{s.t. } (L,E,C)\in\mathcal{S}_f',\end{aligned} \tag{23}$$

where

$$\begin{aligned}H(U,V,L,E,C) =& F(L,E) + \frac{\lambda_1}{2}\|V^T - V^T C\|_F^2\\ &+ \frac{\lambda_2}{2}\|UV^T - L\|_F^2,\end{aligned} \tag{24}$$

$$\begin{aligned}\mathcal{S}_f' =& \{(L,E,C) : \|L\|_\infty \le \alpha_1, \|E\|_\infty \le \alpha_2,\\ &\|E\|_0 \le s, \|c_i\|_0 \le d, C_{ii} = 0, \forall i \in [n]\},\end{aligned} \tag{25}$$

The parameters λ_1 and λ_2 affect the constraints and the gap between solutions of (23) and (6) [16]. The solution of (23) is the same as that of (6) when λ_1 and λ_2 approach the infinity.

We propose a Sparse Alternative Proximal Algorithm (Sparse-APA) to solve (6), details summarized in Algorithm 1. The gradient of $H(U,V,L,E,C)$ with respect to U, V, L, E, and C are shown as follows:

$$\nabla_U H = \lambda_2(UV^T - L)V, \tag{26}$$

$$\nabla_V H = \lambda_1(I-C)(V-C^TV) + \lambda_2(UV^T-L)^TU, \tag{27}$$

$$\nabla_L H = \nabla F(L,E) - \lambda_2(UV^T - L), \tag{28}$$

$$\nabla_E H = \nabla F(L,E), \tag{29}$$

$$\nabla_C H = -\lambda_1 V(V^T - V^TC), \tag{30}$$

where

$$[\nabla F(L,E)]_{ij} = \frac{\dot{\Phi}(\omega_{Y_{ij}} - X_{ij}) - \dot{\Phi}(\omega_{Y_{ij}-1} - X_{ij})}{\Phi(\omega_{Y_{ij}} - X_{ij}) - \Phi(\omega_{Y_{ij}-1} - X_{ij})}, \forall (i,j).$$

The initialization L^0, U^0 and V^0 are selected as follows,

$$L^0_{ij} = \begin{cases} \frac{\omega_l - \omega_{l-1}}{2} & \text{if} \quad Y_{ij} = l, 0 < l < K \\ \frac{\alpha_1 - \omega_{K-1}}{2} & \text{if} \quad Y_{ij} = K \\ \frac{\alpha_1' - \omega_1}{2} & \text{if} \quad Y_{ij} = 0 \end{cases} \tag{31}$$

Let $L^0 = U_{L^0}\Sigma_{L^0}V_{L^0}^T$ denote the singular value decomposition of L^0, set

$$U^0 = U_{L^0}\Sigma_{L^0}^{1/2} \quad \text{and} \quad V^0 = \Sigma_{L^0}^{1/2}V_{L^0} \tag{32}$$

Each iteration of our algorithm contains steps of proximal gradient descent with proximal mapping [38]. We introduce the definition of the proximal map and its reduced form in Definition 2.

Definition 2. *[5] Let $\kappa(u)$ be a proper and lower semicontinuous function of u. The proximal map associated to κ is defined as:*

$$prox^{\kappa}(x) = \arg\min_u \{\kappa(u) + \frac{1}{2}\|u-x\|_F^2\}, \tag{33}$$

Specially, when $\kappa(u)$ is an indicator function, i.e., $\kappa(u) = 0$ if $u \in \chi$ for some set χ and $\kappa(u) = \infty$ if $u \notin \chi$, the proximal map reduces to a projection onto χ, defined by

$$prox^{\kappa}(x) = \arg\min_{u \in \chi} \|u-x\|_F^2, \tag{34}$$

Since there are multiple variables in our problem, we update these variables alternatively. When updating a variable in the $(t+1)$-th iteration, we use the updated variables in the $(t+1)$-th iteration and the non-updated variables in the t-th iteration. Applying the proximal gradient method [38] to our setup results in the following updates,

$$U^{t+1} \in prox(U^t - \tau_U \nabla_U H((U^t, V^t, L^t, E^t, C^t)), \tag{35}$$

$$V^{t+1} \in prox(V^t - \tau_V \nabla_V H((U^{t+1}, V^t, L^t, E^t, C^t)), \tag{36}$$

$$L^{t+1} \in prox^{B(L)}(L^t - \tau_L \nabla_L H((U^{t+1}, V^{t+1}, L^t, E^t, C^t)), \tag{37}$$

$$E^{t+1} \in prox^{J_1(E)+J_2(E)} (E^t - \tau_E \nabla_E H((U^{t+1}, V^{t+1}, L^t, E^{t+1}, C^t)), \tag{38}$$

$$C^{t+1} \in prox^{K_1(C)+K_2(C)} (C^t - \tau_C \nabla_C H((U^{t+1}, V^{t+1}, X^{t+1}, E^{t+1}, C^t)), \tag{39}$$

where $B(X)$, $J_1(E)$, $J_2(E)$, $K_1(C)$, and $K_2(C)$ are indicator functions and

$$B(L) = \begin{cases} \infty & \text{if } \|L\|_\infty > \alpha_1 \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

$$J_1(E) = \begin{cases} \infty & \text{if } \|E\|_\infty > \alpha_2 \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

$$J_2(E) = \begin{cases} \infty & \text{if } \|E\|_0 > s \\ 0 & \text{otherwise} \end{cases} \tag{42}$$

$$K_1(C) = \begin{cases} \infty & \text{if there exists a } C_{ii} \text{ s.t. } C_{ii} \neq 0, i \in [n] \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

$$K_2(C) = \begin{cases} \infty & \text{if there exists a } c_i \text{ s.t. } \|c_i\|_0 > d, i \in [n] \\ 0 & \text{otherwise} \end{cases} \tag{44}$$

Combining (37) and (40), one can see that the projection of (37) is obtained by setting $L_{ij}^{t+1} = \alpha_1$ if $L_{ij}^{t+1} > \alpha_1$ and $L_{ij}^{t+1} = -\alpha_1$ if $L_{ij}^{t+1} < -\alpha_1$. These correspond to Steps 4-5 of Algorithm 1. Similarly, combining (38), (41) and (42), we set $E_{ij}^{t+1} = \alpha_2$ if $E_{ij}^{t+1} > \alpha_2$ and set $E_{ij}^{t+1} = -\alpha_2$ if $E_{ij}^{t+1} < -\alpha_2$. We then only keep s entries with the largest absolute values and set other nonzero entries to zero. These correspond to Steps 7-10 in Algorithm 1. Combining (39), (43) and (44), one can see that the projection of (43) can be obtained by setting diagonal entries of C^{t+1} to zero. The projection of (44) is obtained by keeping d entries with the largest absolute value of c_j^{t+1} for any j. Projected operations of C correspond to Steps 12-16 in Algorithm 1.

The step sizes in the t-th iteration are selected as

$$\tau_U = \frac{1}{\lambda_2\|(V^t)^TV^t\|_F}, \tag{45}$$

$$\tau_V = \frac{1}{\|\lambda_1(I-C^t)(I-C^t)^T\|_F + \|\lambda_2(U^t)^TU^t\|_F}, \tag{46}$$

$$\tau_L = \frac{1}{\frac{\sqrt{mn}}{\sigma^2\beta^2} + \lambda_2\sqrt{m}}, \tag{47}$$

$$\tau_E = \frac{\sigma^2\beta^2}{\sqrt{mn}}, \tag{48}$$

and

$$\tau_C = \frac{1}{\lambda_1\|V^t(V^t)^T\|_F}. \tag{49}$$

These step sizes are smaller or equal to the reciprocals of the smallest Lipschitz constants of $\nabla_U H$, $\nabla_V H$, $\nabla_L H$, $\nabla_E H$, and $\nabla_C H$ in the t-th iteration, respectively. Details of the calculations are shown in Appendix A. This property is useful for the convergence analysis of Sparse-APA. As mentioned earlier, if λ_1 and λ_2 are large enough, (23) approximates (6). However, if λ_1 and λ_2 are too large, the step sizes in (45)-(47) and (49) are very small, and that affect the convergence rate. One practical solution is to dynamically change λ_1 and λ_2 from small to large with a fixed multiplier and use the result from the previous step as the initialization in the current

step [44]. In our numerical experiments, we fix λ_1 and λ_2 to simplify the algorithm. We also remark that each step size is only related to at most two variables. This simplifies the computation.

Since C^t is a sparse matrix with only d nonzero entries per column, we can utilize its sparse property to reduce the computational complexity. Then one can check that the computational complexity of Algorithm 1 in each iteration is in the order of $\mathcal{O}(mnr)$ and is dominated by step 2 and step 4.

Same as SSC [13] and LRR [32], we utilize the normalized spectral clustering [37] to obtain the final group labels base on the solution C returned by Algorithm 1. Details are shown in Algorithm 2. We briefly summarize the steps here. Let C denote the coefficient matrix returned by Algorithm 1. The normalized spectral clustering method first computes the Laplacian matrix $\mathcal{L}^{\text{sym}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where $W = \frac{|C|+|C|^T}{2}$, D is the diagonal matrix, and D_{ii} is the sum of the i-th row of W. We then form the matrix Ψ, where columns of Ψ are p eigenvectors corresponding to the bottom p eigenvalues of $\mathcal{L}^{\text{sym}}$. The last step is normalizing rows of Ψ and then applying the k-means algorithm [21] to n rows. k-mean algorithm alternatively finds centroids of the clusters and labels data points.

Algorithm 1 Sparse-APA

Input: Quantized matrix $Y \in \mathbb{R}^{m\times n}$, initialization matrices $L^0 \in \mathbb{R}^{m\times n}$, $U^0 \in \mathbb{R}^{m\times r}$, $V^0 \in \mathbb{R}^{n\times r}$ and zero matrix $C^0 \in \mathbb{R}^{n\times n}$, $E^0 \in \mathbb{R}^{m\times n}$, parameters r, d, β and T.

1 **for** $t = 0, 1, 2, ..., T$ **do**
2 $\quad U^{t+1} = U^t - \tau_U \nabla_U H_(U^t, V^t, L^t, E^t, C^t)$.
3 $\quad V^{t+1} = V^t - \tau_V \nabla_V H(U^{t+1}, V^t, L^t, E^t, C^t)$.
4 $\quad L^{t+1} = L^t - \tau_L \nabla_L H(U^{t+1}, V^{t+1}, L^t, E^t, C^t)$.
5 $\quad$ If $L_{ij}^{t+1} > \alpha_1$, set $L_{ij}^{t+1} = \alpha_1$. If $L_{ij}^{t+1} < -\alpha_1$, set $L_{ij}^{t+1} = -\alpha_1$.
6 $\quad E^{t+1} = E^t - \tau_E \nabla_E H(U^{t+1}, V^{t+1}, L^{t+1}, E^t, C^t)$.
7 $\quad$ If $E_{ij}^{t+1} > \alpha_2$, set $E_{ij}^{t+1} = \alpha_2$. If $E_{ij}^{t+1} < -\alpha_2$, set $E_{ij}^{t+1} = -\alpha_2$.
8 $\quad$ **if** $\sum_j \sum_i \mathbf{1}_{[E_{ij}^{t+1}\neq 0]} > s$, $\forall i \in [m], \forall j \in [n]$ **then**
9 $\quad\quad E^{t+1}$ only keeps s entries with the largest absolute values. Other nonzero entries are set to be zero.
10 $\quad$ **end if**
11 $\quad C^{t+1} = C^t - \tau_C \nabla_C H(U^{t+1}, V^{t+1}, L^{t+1}, E^{t+1}, C^t)$.
12 $\quad$ Set $C_{ii}^{t+1} = 0$, $\forall i \in [n]$
13 $\quad$ **for** every $j = 1, 2, ..., n$ **do**
14 $\quad\quad$ **if** $\sum_i \mathbf{1}_{[C_{ij}^{t+1}\neq 0]} > d$, $\forall j \in [n]$ **then**
15 $\quad\quad\quad c_j^{t+1}$ only keeps d entries with the largest absolute values. Other nonzero entries are set to be zero.
16 $\quad\quad$ **end if**
17 $\quad$ **end for**
18 **end for**
19 **Return:** L, E and C.

Next we provide the theorem of the convergence of Sparse-APA.

Theorem 2. *Sparse-APA globally converges to a critical point of (23) from any initial point.*

Algorithm 2 Spectral Clustering [37]

Input: Coefficient matrix $C \in \mathbb{R}^{n\times n}$

1 Normalize the columns of C as $\frac{c_i}{\|c_i\|_0} \to c_i$.
2 Obtain the affinity matrix W as $W = \frac{|C|+|C|^T}{2}$.
3 Define D to be the diagonal matrix and D_{ii} is the sum of the i-th row of W
4 Construct the Laplacian matrix $\mathcal{L}^{\text{sym}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
5 Compute p eigenvectors corresponding to the bottom p eigenvalues of $\mathcal{L}^{\text{sym}}$ and form the matrix Ψ by stacking the p eigenvectors in columns.
6 Normalize each row of Ψ and cluster n rows into p groups using k-means algorithm.
7 **Return:** Segmentation of the data.

Proof: Sparse-APA can be viewed as a special case of Proximal Alternating Linearized Minimization (PALM) algorithm from the results in [5]. From [5], Sparse-APA globally converges to a critical point of (23) from any initial point, if provided that $H(U, V, L, E, C)$ is Lipschitz differentiable, and

$$H(U, V, L, E, C) + K_1(C) + K_2(C) + B(L) + J_1(E) + J_2(E) \tag{50}$$

satisfies the Kurdyka-Lojasiewicz (KL) property.

The proof of the Lipschitz differentiable property of $H(U, V, L, E, C)$ can be found in B. $B(L)$, $J_1(E)$, $J_2(E)$, $K_1(C)$ and $K_2(C)$ are indicator functions of semi-algebraic sets. Therefore, they are KL functions according to [5]. Since $H(U, V, L, E, C)$ is differentiable everywhere, or equivalently, real analytic, $H(U, V, L, E, C)$ also has the KL property according to the examples in session 2.2 of [50]. Thus, (50) satisfies the KL property. ■

V. Experiments

We evaluate the performance on both synthetic data and actual data from the Extended Yale Face Dataset B [30],[20]. The recovery performance is measured by the relative recovery error $\|L^* - \tilde{L}\|_F^2 / \|L^*\|_F^2$, where L^* denotes the actual data, and $\tilde{L}$ denotes the recovered data. After obtaining the coefficient matrix using Sparse-APA, we implemented spectral clustering [37] in Algorithm 2 to cluster the recovered data matrix. The clustering performance is measured by the clustering error ratio, which is the fraction of data points that are incorrectly labeled among the total n data points. The corruption rate $\frac{s}{mn}$ is the fraction of nonzero entries in E^*. We set $\lambda_1 = \sqrt{mn}$, $\lambda_2 = \frac{\sqrt{mn}}{2}$.

For data recovery, we compare Sparse-APA with two existing methods for recovering a low-rank matrix from quantized measurements. One method is Approximate Projected Gradient Method (APGM) [4], which solves the nonconvex quantized low-rank matrix recovery problem using a gradient descent method. APGM does not consider corruptions in the measurements. The other method is Quantized Robust Principal Component Analysis (QRPCA) [28], which solves the convex relaxation of the quantized low-rank matrix recovery problem. QRPCA also handles partial corruptions by promoting a low-rank matrix using the nuclear norm and

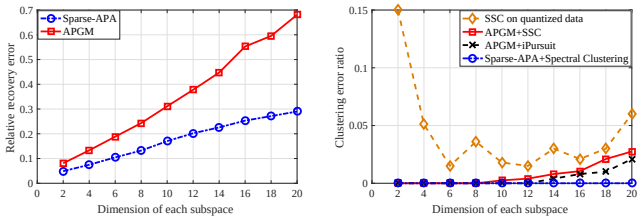
promoting a sparse error matrix using the ℓ_1 norm. We also compare with two existing data clustering methods SSC [13] and Innovation Pursuit (iPursuit) [42]. SSC first estimates the subspace-preserving coefficient matrix C by solving (5). SSC then applies spectral clustering to cluster the data. iPursuit clusters subspaces with considerable intersections. It first solves an ℓ_1 minimization problem to find the innovation of one subspace to the sum of other subspaces. After identifying one subspace, it repeats this process to find other subspaces consecutively. For data clustering, we compare "Sparse-APA + spectral clustering" with "APGM + SSC," which means first applying APGM to recover the actual data and then applying SSC on the recovered data to cluster the data points. Similarly, we also compare with "APGM + iPursuit," " QRPCA + SSC," and "QRPCA + iPursuit." For SSC in all experiments, we use Alternating Direction Method of Multipliers (ADMM) [6] to solve it. The simulations run in MATLAB on a computer with 3.4 GHz Intel Core i7. The iteration number T is set to be 100.

A. Performance on synthetic data

We first test Sparse-APA on synthetic data. We generate independent subspaces $\{S_i\}_{i=1}^p, p = 5$ following the approach in [17]. Specifically, let $M_i \in \mathbb{R}^{m\times d}, i \in [p]$ denote the base matrices of the p subspaces and $R_i \in \mathbb{R}^{m\times d}, i \in [p-1]$ denote random matrices that are generated independently from the standard normal distribution $\mathcal{N}(0, 1)$. Entries in M_1 are also generated from $\mathcal{N}(0, 1)$. $M_i (i > 1)$ is generated by applying orthogonal-triangular decomposition to $M_{i-1}+R_i$ and keeping the largest d columns of the unitary matrix.

We vary the dimension d of each subspace from 2 to 20. The data points from subspace S_i are generated as $M_i q$, where entries of $q \in \mathbb{R}^d$ are sampled independently from $\mathcal{N}(0, 1)$. We set $m = 100$ and generate $n_i = 200$ for data in each subspace (except for those comparisons that need to vary m or n_i). We then rescale L^* such that $L^*_{ij} \in [-1, 1]$. The entries of the noise matrix N are drawn i.i.d. from $\mathcal{N}(0, 0.24^2)$. K is set to be 5, with $\omega_1 = -0.8$, $\omega_2 = -0.3$, $\omega_3 = 0.3$, and $\omega_4 = 0.8$. We test Sparse-APA on this dataset when d varies from 2 to 20. Note that L^* is a full rank matrix when d is 20. In all the simulations on synthetic data using Sparse-APA and APGM, we set the recovery rank $r = pd$, which is the same with the rank of the actual data matrix L^*.

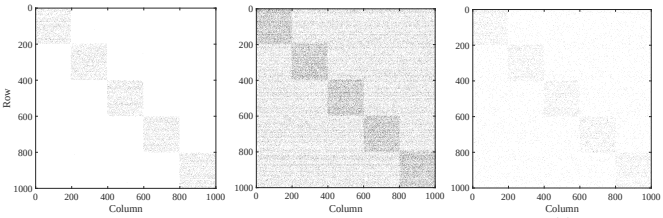
We first consider the case of no corruptions, i.e., $E^* = 0$. Since the number of nonzeros in E^* might not be known in practice, we usually set s slightly larger than the estimated number of nonzeros. Here we set $s = 0.05mn$ in Sparse-APA to estimate the performance that s is larger than the actual number of nonzeros. For recovery performance, we compare Sparse-APA with APGM. For subspace clustering, we compare Sparse-APA+Spectral Clustering with APGM + SSC, APGM + iPursuit. We also compare Sparse-APA with directly applying SSC on quantized data. We run each method 100 times and average the results. As shown in Fig. 2, Sparse-APA outperforms other methods in both data recovery and data clustering compared with the other methods. Note that by Theorem 1, the recovery error of equation (13) in terms of Frobenius norm is upper bounded by $\mathcal{O}(\sqrt{\frac{d}{m}})$. Since the relative recovery error in the numerical results is defined as the square of the Frobenius norm, the relative recovery error obtained by Sparse-APA is approximately proportional to d in Fig. 2 (a), which coincides with Theorem 1. In Fig. 2 (b), the average clustering error ratios obtained by Sparse-APA are all zeros when the dimension varies.



(a) (b)

Fig. 2. (a) Relative recovery error when d changes. (b) Clustering error ratio when d changes ($p = 5$).

We can also check the clustering performance from the perspective of the subspace-preserving property. Fig. 3 (a) shows the locations of nonzero values of the estimated coefficient matrix $\hat{C}$ obtained by Sparse-APA when $d = 10$, and $p = 5$. Fig. 3 (b) shows the locations of nonzero values in $\hat{C}$ obtained by APGM + SSC. We can find that the coefficient matrix obtained by Sparse-APA only have nonzero entries in diagonal blocks, while the matrix obtained by APGM + SSC has many nonzero entries outside the diagonal blocks. We then choose the largest d nonzero entries in each column of the coefficient matrix obtained by APGM + SSC, and the result is shown in Fig. 3 (c). There are still quite a few nonzero entries outside the diagonal blocks in Fig. 3 (c). This shows that Sparse-APA can achieve the desired subspace-preserving property better.

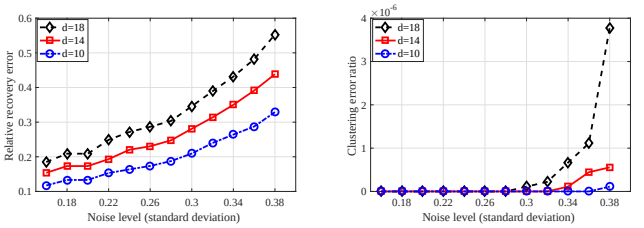


(a) (b) (c)

Fig. 3. (a) The coefficient matrix obtained by Sparse-APA. (b) The coefficient matrix obtained by APGM + SSC. (c) The coefficient matrix obtained by APGM + SSC with thresholding. ($d = 10$, $p = 5$)

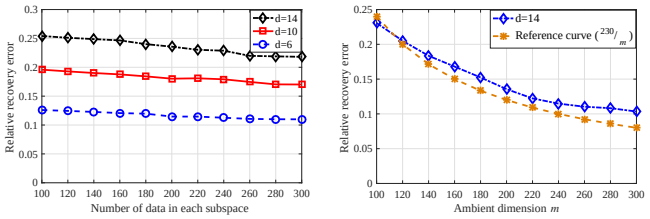
We also evaluate Sparse-APA by varying only one parameter each time. Fig. 4 shows the relative recovery error and the clustering error ratio under different noise levels. Each entry of N is drawn from $\mathcal{N}(0, \sigma^2)$ with σ changing from 0.16 to 0.38. Fig. 5(a) shows the recovery error when the subspace dimension d remains the same, and the number of data points in each subspace n_i increases. We set n_i the same for all i and increase n_i from 100 to 300. We can see that the error decreases when the number of data points increases. We then keep $n_i = 300$ and increase m from 100 to 300. Fig. 5 (b) shows the results when $d = 14$. We also compare the results obtained by Sparse-APA with a decreasing curve which is inversely proportional to m. We can find that the relative recovery error obtained by Sparse-APA is approximately inversely proportional to m,

which coincides with Theorem 1. Note our clustering results are all 100% correct in this setup. We only show the recovery performance in Fig. 5.



(a) (b)

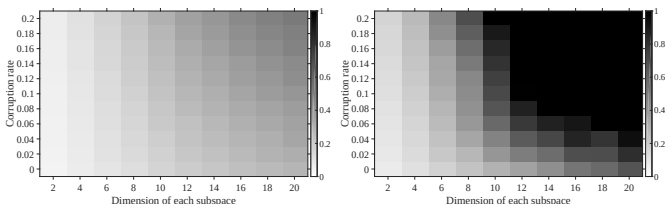
Fig. 4. (a) Relative recovery error when the noise level changes. (b) Clustering error ratio when the noise level changes. (Matrix dimension 100×1000, $p = 5$)



(a) (b)

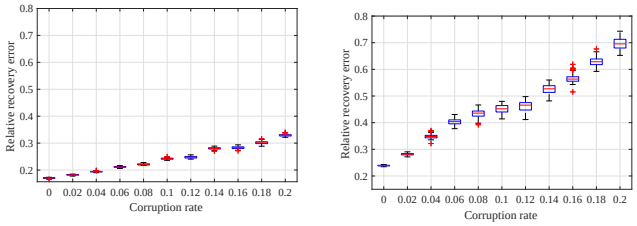
Fig. 5. (a) Recovery error when the number of data points in each subspace changes. Noise level $\mathcal{N}(0, 0.24^2)$, $p = 5$, $m = 100$ (b) Recovery error when the ambient dimension m changes. Noise level $\mathcal{N}(0, 0.24^2)$, $p = 5$, $n = 1500$

We next consider the case of partial corruptions. Nonzero entries of the E^* are independently and uniformly selected from $[0.5, 5]$ and $[-5, -0.5]$. Fig. 6(a) shows the relative recovery error by Sparse-APA when the corruption rate and the dimension of each subspace change. The corruption rate $\frac{s}{mn}$ changes from 0 to 0.2. The dimension d of each subspace changes from 2 to 20. The recovery performance by QRPCA is shown in Fig. 6(b). Sparse-APA has a better performance even in the full rank case under 20% corruption rate, while QRPCA has a significant recovery error when d is greater than 12 or $\frac{s}{mn}$ is larger than 0.1. Fig. 7 (a) and (b) show the box-plot-diagram of relative recovery error when $d = 8$ and the corruption rate changes. The top and bottom of each "box" is the 25th and 75th percentile of the samples, respectively. Fig. 7 shows that Sparse-APA has both smaller recovery errors and smaller variances. In Fig. 8, we compare Sparse-APA + Spectral Clustering with QRPCA + iPursuit and QRPCA + SSC when $d = 8, r = 40$ and $d = 12, r = 60$, respectively. The clustering error on average is at most in the order of 10^{-5} using Sparse-APA. Compared with other methods, Sparse-APA has the best performance.



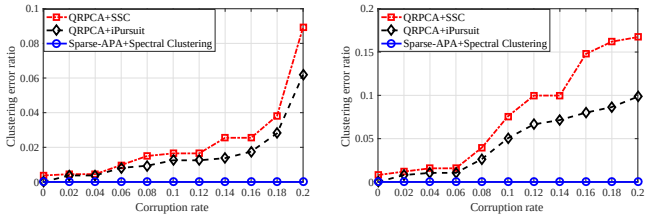
(a) Sparse-APA (b) QRPCA

Fig. 6. Relative recovery errors when the dimension of each subspace and the corruption rate change.



(a) Sparse-APA (b) QRPCA

Fig. 7. Box-plot-diagram of relative recovery error when the corruption rate changes. ($d = 8$)



(a) $d = 8, r = 40$ (b) $d = 12, r = 60$

Fig. 8. Clustering error ratio comparisons when corruption rate changes.

We further test the Sparse-APA + Spectral Clustering in larger datasets and show the performances of recovery, clustering, and running time. We fix $p = 10$ and $\frac{s}{mn} = 0.1$. We choose three groups of data for testing: (1) $m = 500$, $n = 5000$, $n_i = 500$ for all i, $d = 10$ (2) $m = 1000$, $n = 10000$, $n_i = 1000$ for all i, $d = 10$ and (3) $m = 2000$, $n = 200000$, $n_i = 2000$ for all i, $d = 20$. The results are shown in Table I.

TABLE I
PERFORMANCE OF SPARSE-APA+SPECTRAL CLUSTERING IN LARGE DATASETS ($p = 10$, 10% CORRUPTION)

	Relative recovery error	Clustering error ratio	Runing time
$m = 500, n = 5000,$ $n_i = 500\ \forall i \in [10], d = 5$	0.086	0	610s
$m = 1000, n = 10000,$ $n_i = 1000\ \forall i \in [10], d = 10$	0.077	0	2700s
$m = 2000, n = 20000,$ $n_i = 2000\ \forall i \in [10], d = 20$	0.075	0	14230s

B. Performance on real data

We then test Sparse-APA on the Extended Yale Face Dataset B [20], [30]. The dataset consists of 192×168 pixel cropped face images belonging to 38 different individuals. Each group has 64 face images with various illumination and poses. We downsample all images to 48×42, rescale all pixels to values in $[0, 1]$, and vectorize them. We first pick 10 subjects with 64 images per subject. We obtain the original data matrix $L^* \in R^{2016 \times 640}$ with $L^*_{ij} \in [0, 1]$. In all the simulations below (except for those that vary K), K is set to be 5, and $\omega_1 = 0.05$, $\omega_2 = 0.4$, $\omega_3 = 0.7$, and $\omega_4 = 0.95$. The entries of noise matrix N are drawn i.i.d. from $\mathcal{N}(0, 0.24^2)$. Each result is averaged over 100 runs.

Face images of one subject under various illumination lie close to a 9-dimensional subspace [1]. Hence we can set d in $6 \sim 15$ in Sparse-APA. Moreover, since subspaces belonging

to different people are not independent, we vary the recovery rank r in Sparse-APA to test the performance. Fig. 9 and Fig. 10 show the relative recovery error and clustering error shown in Figs. 9 and 10, results do not differ much in a certain range of d and r.

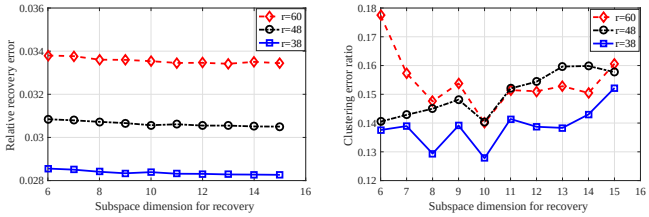


Fig. 9. Relative recovery error and clustering error ratio when d and r in Sparse-APA change.

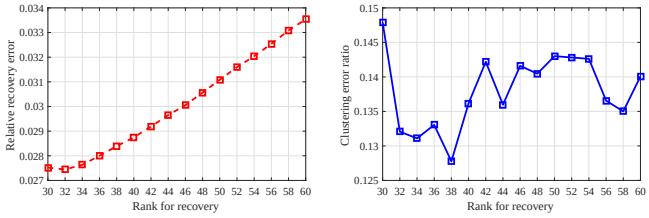
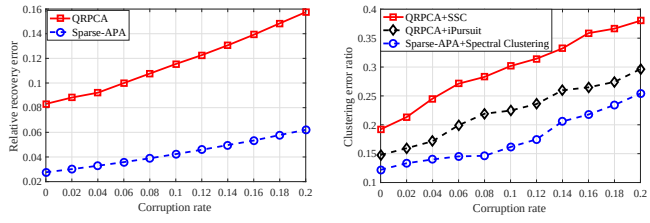


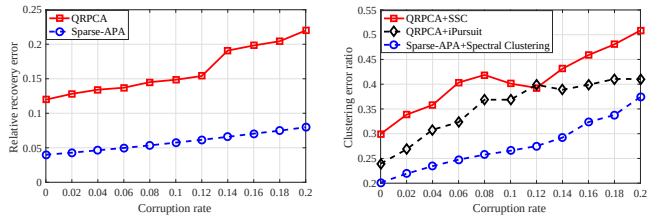
Fig. 10. Relative recovery error and clustering error ratio when r in Sparse-APA varies. ($d = 10$)

We next fix $d = 10$ and $r = 38$ for the following tests. We first test the performance of Sparse-APA on cases with additional corrupted data. Nonzero entries of the sparse matrix E^* are uniformly selected from $[0.5, 5]$ and $[-5, -0.5]$. The corruption rate changes from 0 to 0.2. We compare the recovery performance of Sparse-APA and QRPCA in Fig. 11 (a). In Fig. 11 (b), we compare the clustering performance with QRPCA + iPursuit and QRPCA + SSC. Fig. 12 shows the same comparison when K is set to 3. Three quantized levels are set as $\omega_0 = -\infty$, $\omega_1 = 0.2$, $\omega_2 = 0.8$ and $\omega_3 = \infty$. The results show that Sparse-APA has the best performance among all these methods.



(a) Relative recovery error (b) Clustering error ratio

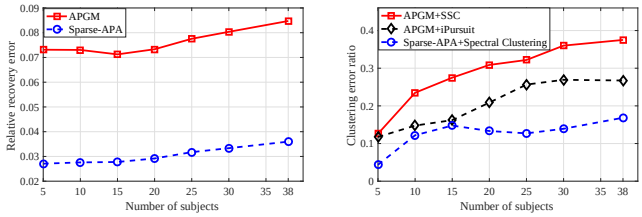
Fig. 11. Comparisons when corruption rate changes. ($K = 5$)



(a) Relative recovery error (b) Clustering error ratio

Fig. 12. Comparisons when corruption rate changes. ($K = 3$)

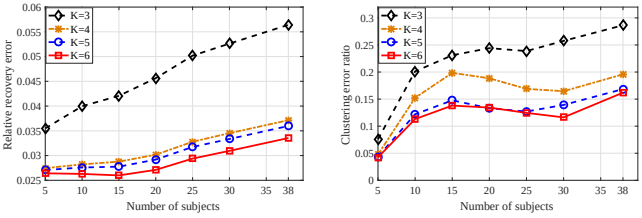
We next vary the subject number from 5 to 38. In Fig. 13 (a), we compare Sparse-APA with APGM for the recovery performance. In Fig. 13 (b), we compare with APGM + iPursuit and APGM + SSC for the clustering performance. Sparse-APA achieves the best performance among all these methods.



(a) Relative recovery error (b) Clustering error ratio

Fig. 13. Comparisons when the number of subjects changes.

Fig. 14 shows the performance of Sparse-APA with different number of subjects and different level K. When $K = 3$, we set $\omega_0 = -\infty$, $\omega_1 = 0.2$, $\omega_2 = 0.8$ and $\omega_3 = \infty$. When $K = 4$, we set $\omega_0 = -\infty$, $\omega_1 = 0.1$, $\omega_2 = 0.4$, $\omega_3 = 0.8$, and $\omega_4 = \infty$. Boundaries selections are the same as before when $K = 5$. When $K = 6$, we set $\omega_0 = -\infty$, $\omega_1 = 0.05$, $\omega_2 = 0.35$, $\omega_3 = 0.55$, $\omega_4 = 0.75$, $\omega_5 = 0.95$ and $\omega_6 = \infty$. As shown in Fig. 14, the recovery and the clustering errors decrease when K increases.



(a) (b)

Fig. 14. (a) Relative recovery error when number of subjects changes with different level K. (b) Clustering error ratio comparisons when number of subjects changes with different level K.

In Fig. 15, we show visual comparisons of different steps of processing and recovery under different conditions. Images of Fig. 15 belong to two different subjects. Fig. 15 (a) shows the image with noise and 10% corruption. Fig. 15 (b) and (c) show the images after quantization with $K = 5$ and $K = 3$, respectively. Fig. 15 (d) is the quantized image with 10% missing blocks when $K = 5$. The quantized images are visualized by first transforming all quantized values to values in $[0, 1]$, and then transforming new normalized values to unsigned 8-bit integer. Fig. 15 (e) shows the original image. Fig. 15 (f) and (g) show the recovered images corresponding to (b) and (c). Fig. 15(i)-(p) follow similarly for a different subject.

Table II shows the running time of different methods. $126s+10s$ for QRPCA + SSC means that QRPCA takes $126s$ and SSC takes $10s$. Sparse-APA is 5-10 times faster than the other two methods.

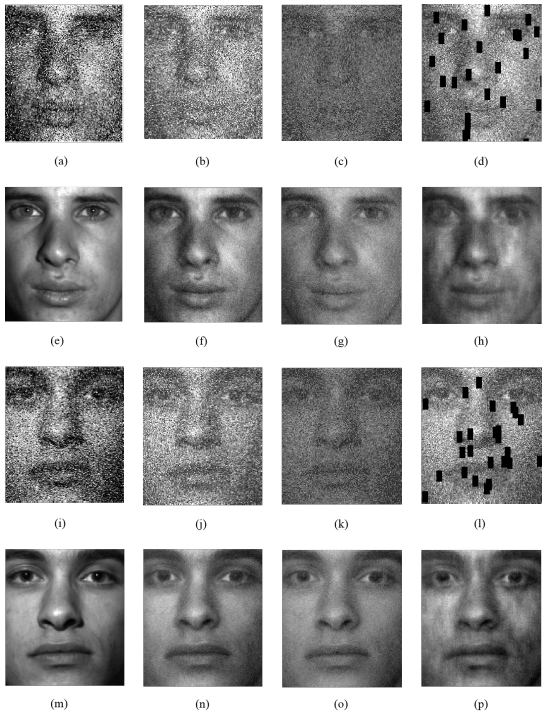


Fig. 15. (a)(i) Image with noise and 10% corruption (b)(j) Quantized image ($K = 5$) (c)(k) Quantized image ($K = 3$) (d)(l) Quantized image with missing blocks ($K = 5$) (e)(m) Original image (f)(n) Recovered image with 10% corruption ($K = 5$) (g)(o) Recovered image with 10% corruption ($K = 3$) (h)(p) Recovered image with 10% missing blocks ($K = 5$).

TABLE II
RUNNING TIME OF DIFFERENT METHODS (10% CORRUPTION, $m = 2016$)

	10 subjects	20 subjects	30 subjects
Sparse-APA+ Spectral Clustering	29s	78s	127s
QRPCA+SSC	126s+10s	530s+30s	1127s+75s
QRPCA+iPursuit	126s+23s	530s+55s	1127s+118s

VI. CONCLUSION AND DISCUSSIONS

This paper is the first work that studies the combined data recovery and subspace clustering problem when the ground-truth data points belong to one of p low-dimensional subspaces, and the obtained measurements are quantized and partially corrupted. The relative data recovery error of the proposed method approaches zero asymptomatically. The error bound is reduced by a factor of $\sqrt{1/p}$ compared with that of directly applying the existing quantized low-rank matrix recovery methods. A Sparse Alternative Proximal Gradient Algorithm is developed to solve the nonconvex combined data recovery and clustering problem. The method is validated on synthetic and Extended Yale Face B datasets. The method developed here can be applied to image denoising, sensor networks, and power system monitoring. Future works include extending the method to handle data losses and distributed implementation of the developed method.

ACKNOWLEDGEMENT

This research is supported in part by ARO W911NF-17-1-0407, NSF 1508875, and IBM corporation.

APPENDIX

A. Lemmas used in the proof of Theorem 1

Lemma 1. *[4] Take any two numbers m and n such that $1 \leq m \leq n$. Suppose that $A = [a_{ij}]_{1\leq i\leq m, 1\leq j\leq n}$ is a matrix whose entries are independent random variables that satisfy, for some $\sigma^2 \in [0, 1]$,*

$$\mathbb{E}[a_{ij}] = 0, \quad \mathbb{E}[a_{ij}^2] \leq \sigma^2, \quad \text{and} \quad |a_{ij}| \leq 1 \quad a.s. \tag{51}$$

Suppose that $\sigma^2 \geq m^{-1+\epsilon}$ for some $\epsilon > 0$. Then

$$P(\|A\|_2 \geq 2.01\sigma\sqrt{m}) \leq C_1(\epsilon)e^{-C_2\sigma^2 m}, \tag{52}$$

where $C_1(\epsilon)$ is a constant that depends only on ϵ and C_2 is a positive universal constant.

Lemma 2. *Let $\hat{\theta} = \text{vec}(\hat{X})$, $\theta^* = \text{vec}(X^*)$, and $\hat{X} = \hat{L} + \hat{E}$, $X^* = L^* + E^* \in \mathcal{S}_f$. Follow the same assumptions as those of Theorem 1. Then with probability at least $1 - C_1e^{-C_2\xi n/p}$,*

$$\begin{aligned} |\left\langle \nabla_\theta \mathcal{F}(\theta^*), \hat{\theta} - \theta^* \right\rangle| \leq & 4.02L_\alpha\sqrt{\xi dn}(\|\hat{X} - X^*\|_F + 2\alpha_2\sqrt{s}) \\ & + 2\alpha_2 s L_\alpha, \end{aligned} \tag{53}$$

holds for the positive constants C_1 and C_2.

Proof: The proof is built upon the proofs of Theorem 1 in [19] and Theorem 3.1 in [4]. Here we extend it to L^* with columns in p subspaces.

Consider

$$Z_{ij} := [L_\alpha^{-1}\nabla_X F(X^*)]_{ij} = -L_\alpha^{-1}\sum_{l=1}^{K}\frac{\dot{f}_l(X_{ij}^*)}{f_l(X_{ij}^*)}\mathbf{1}_{[Y_{ij}=l]}.$$

Using (12) and the fact that $\sum_{l=1}^{K} f_l(X_{ij}) = 1$, we have $\mathbb{E}[Z_{ij}] = 0$, $|Z_{ij}| \leq 1$, and $\mathbb{E}[Z_{ij}^2] \leq 1$.

There exists a permutation matrix Γ^* such that L^* can be written as $L^* = [L_1^*, L_2^*, ..., L_p^*]\Gamma^*$. By Assumption 1, $\hat{L}$ can be written as $\hat{L} = [\hat{L}_1, \hat{L}_2, ..., \hat{L}_p]\hat{\Gamma}, \hat{L}_i \in \hat{S}_i$, where $\hat{\Gamma}$ is a permutation matrix and $\{\hat{S}_i\}_{i=1}^p$ are subspaces with dimension of each subspace smaller or equal to d.

Given a matrix $G = [G_1, G_2...G_i...G_p]\Gamma \in \mathbb{R}^{m\times n}$, for some permutation matrix Γ, we define an operator $P_{\Omega^{G_i}}$ that keeps entries of G in the locations of G_i unchanged and sets other entries as zero. Mathematically, $P_{\Omega^{G_i}}(X) = [0, 0...G_i...0, 0]\Gamma \in \mathbb{R}^{m\times n}$. Then one can check that $\|\nabla_X F(X_i^*)\|_2 = \|\nabla_X F(P_{\Omega^{X_i^*}}(X^*))\|_2$. By Lemma 1, we can obtain

$$\|L_\alpha^{-1}\nabla_X F(P_{\Omega^{X_i^*}}(X^*))\|_2 = \|L_\alpha^{-1}\nabla_X F(X_i^*)\|_2 \leq 2.01\sqrt{\xi n/p} \tag{54}$$

holds with probability at least $1 - C_1e^{-C_2\xi n/p}$ for some positive constants C_1 and C_2.

Note that

$$\begin{aligned} & \|P_{\Omega^{\hat{L}_i}}(\hat{L}) - P_{\Omega^{L_i^*}}(L^*)\|_* \\ & \leq \sqrt{2d}\|P_{\Omega^{\hat{L}_i}}(\hat{L}) - P_{\Omega^{L_i^*}}(L^*)\|_F = \sqrt{2d}\|\hat{L}_i - L_i^*\|_F. \end{aligned} \tag{55}$$

We then have

$$
\begin{aligned}
&|\left\langle \nabla_\theta \mathcal{F}(\theta^*), \hat{\theta}-\theta^* \right\rangle| = |\left\langle \nabla_X F(X^*), \hat{X}-X^* \right\rangle| \\
&\le |\left\langle \nabla_X F(X^*), \hat{L}-L^* \right\rangle| + |\left\langle \nabla_X F(X^*), \hat{E}-E^* \right\rangle| \\
&\overset{(a)}{=} |\sum_{i=1}^{p} \left\langle \nabla_X F(X^*), P_{\Omega^{\hat{L}_i}}(\hat{L}) - P_{\Omega^{L_i^*}}(L^*) \right\rangle| \\
&+ |\left\langle \nabla_X F(X^*), \hat{E}-E^* \right\rangle| \\
&\overset{(b)}{=} |\sum_{i=1}^{p} \left\langle P_{\Omega^{\hat{L}_i \cup L_i^*}}(\nabla_X F(X^*)), P_{\Omega^{\hat{L}_i}}(\hat{L}) - P_{\Omega^{L_i^*}}(L^*) \right\rangle| \\
&+ |\left\langle \nabla_X F(X^*), \hat{E}-E^* \right\rangle| \\
&\overset{(c)}{\le} \sum_{i=1}^{p} \|P_{\Omega^{\hat{L}_i \cup L_i^*}}(\nabla_X F(X^*))\|_2 \|P_{\Omega^{\hat{L}_i}}(\hat{L}) - P_{\Omega^{L_i^*}}(L^*)\|_* \\
&+ 2\alpha_2 s L_\alpha \\
&\overset{(d)}{\le} 2.01 L_\alpha \sqrt{2\xi n/p} \sum_{i=1}^{p} \sqrt{2d} \|\hat{L}_i - L_i^*\|_F + 2\alpha_2 s L_\alpha \\
&\le 4.02 L_\alpha \sqrt{\xi dn} \|\hat{L}-L^*\|_F + 2\alpha_2 s L_\alpha \\
&\le 4.02 L_\alpha \sqrt{\xi dn} (\|\hat{X}-X^*\|_F + \|\hat{E}-E^*\|_F) + 2\alpha_2 s L_\alpha \\
&\le 4.02 L_\alpha \sqrt{\xi dn} (\|\hat{X}-X^*\|_F + 2\alpha_2\sqrt{s}) + 2\alpha_2 s L_\alpha
\end{aligned}
$$

holds with probability at least $1 - C_1 e^{-2C_2 \xi n/p}$. (a) holds from the linearity of the inner product. (b) holds because only nonzero entries contribute to the inner product. Here, $P_{\Omega^{\hat{L}_i \cup L_i^*}}$ means keeping entries in the locations of $\hat{L}_i$ and L_i^* unchanged and setting other entries as zero. (c) holds from $|\langle A, B\rangle| \le \|A\|_2 \|B\|_*$. (d) holds from (54) and (55). ∎

Lemma 3 (Lemma A.3 in [2]). *Let $\hat{\theta} = \mathrm{vec}(\hat{X})$, $\theta^* = \mathrm{vec}(X^*)$, and $\hat{X} = \hat{L} + \hat{E}$, $X^* = L^* + E^* \in \mathcal{S}_f$. Then for any $\tilde{\theta} = \theta^* + \eta(\hat{\theta} - \theta^*)$ and any $\eta \in [0,1]$, we have*

$$
\left\langle \hat{\theta}-\theta^*, (\nabla^2_{\theta\theta} \mathcal{F}_Y(\tilde{\theta}))(\hat{\theta}-\theta^*) \right\rangle \ge \gamma_\alpha \|\hat{X}-X^*\|_F^2. \quad (56)
$$

B. Sparse-APA: Proof of the Lipschitz differential property and calculation of Lipschitz constants

We provide the Lipschitz differential property of $H(U,V,L,E,C)$ and compute the corresponding Lipschitz constants of its partial gradients. A function is Lipschitz differentiable if and only if all its partial gradients are Lipschitz continuous. The definition is shown in Definition 3.

Definition 3. *[5] For any fixed matrices $z_1, z_2, .., z_n$, matrix variable y, and a function $y \to \Upsilon(y, z_1, z_2, ..., z_n)$, the partial gradient $\nabla_y \Upsilon(y, z_1, z_2, ..., z_n)$ is said to be Lipschitz continuous with Lipschitz constant $L_p(z_1, z_2, ..., z_n)$, if the following holds*

$$
\begin{aligned}
&\|\nabla_y \Upsilon(y_1, z_1, z_2, ..., z_n) - \nabla_y \Upsilon(y_2, z_1, z_2, ..., z_n)\|_F \\
&\le L_p(z_1, z_2, ..., z_n) \|y_1 - y_2\|_F, \quad \forall y_1, y_2.
\end{aligned}
$$

Let L_{p1}^{t+1}, L_{p2}^{t+1}, L_{p3}^{t+1}, L_{p4}^{t+1} and L_{p5}^{t+1} denote the smallest Lipschitz constants of $\nabla_U H$, $\nabla_V H$, $\nabla_L H$, $\nabla_E H$ and $\nabla_C H$ in the $(t+1)$-th iteration. We have

$$
\begin{aligned}
&\|\nabla_U H(U_1, V^t, L^t, E^t, C^t) - \nabla_U H(U_2, V^t, L^t, E^t, C^t)\|_F \\
&= \|\lambda_2 (U_1 - U_2)(V^t)^T V^t\|_F \\
&\le \|\lambda_2 (V^t)^T V^t\|_2 \|U_1 - U_2\|_F \\
&\overset{(a)}{\le} \|\lambda_2 (V^t)^T V^t\|_F \|U_1 - U_2\|_F \\
&\overset{(b)}{=} \frac{1}{\tau_U(V^t)} \|U_1 - U_2\|_F,
\end{aligned} \quad (57)
$$

where (a) follows from the inequality $\|\cdot\|_2 \le \|\cdot\|_F$. Since $\|\lambda_2 (V^t)^T V^t\|_F \ge L_{p1}^{t+1}$, (b) follows from (45). (57) implies that

$$
L_{p1}^{t+1} \le \|\lambda_2 (V^t)^T V^t\|_F, \text{ and } \tau_U(V^t) \le 1/L_{p1}^{t+1}. \quad (58)
$$

$$
\begin{aligned}
&\|\nabla_V H(U^{t+1}, V_1, L^t, C^t) - \nabla_V H(U^{t+1}, V_2, L^t, C^t)\|_F \\
&= \|\lambda_1 (I - C^t)(I - C^t)^T (V_1 - V_2) + \\
&\quad \lambda_2 (V_1 - V_2)(U^{t+1})^T U^{t+1}\|_F \\
&\overset{(c)}{\le} (\|\lambda_1 (I - C^t)(I - C^t)^T\|_F + \|\lambda_2 (U^{t+1})^T U^{t+1}\|_F) \\
&\quad \cdot \|V_1 - V_2\|_F \\
&\overset{(d)}{=} \frac{1}{\tau_V(U^{t+1}, C^t)} \|V_1 - V_2\|_F,
\end{aligned} \quad (59)
$$

where (c) follows from the triangle inequality, and (d) follows from (46). (59) implies that $\tau_V(U^{t+1}, C^t) \le 1/L_{p2}^{t+1}$.

$$
\begin{aligned}
&\|\nabla_L H(U^{t+1}, V^{t+1}, L_1, E^t, C^t) - \\
&\quad \nabla_L H(U^{t+1}, V^{t+1}, L_2, E^t, C^t)\|_F \\
&= \|\nabla_X F(L_1, E^t) - \nabla_X F(L_2, E^t) + \lambda_2 (L_1 - L_2)\|_F \\
&\overset{(e)}{=} \|\nabla^2_{XX} F(\bar{L})(L_1 - L_2) + \lambda_2 (L_1 - L_2)\|_F \\
&\le \|\nabla^2_{XX} F(\bar{L}) + \lambda_2 I\|_2 \|L_1 - L_2\|_F \\
&\overset{(f)}{\le} \|\nabla^2_{XX} F(\bar{L}) + \lambda_2 I\|_F \|L_1 - L_2\|_F \\
&\le (\|\nabla^2_{XX} F(\bar{L})\|_F + \|\lambda_2 I\|_F) \|L_1 - L_2\|_F \\
&= (\|\nabla^2_{XX} F(\bar{L})\|_F + \lambda_2 \sqrt{m}) \|L_1 - L_2\|_F \\
&\le (\sqrt{mn} \|\nabla^2_{XX} F(\bar{L})\|_\infty + \lambda_2 \sqrt{m}) \|L_1 - L_2\|_F \\
&\overset{(g)}{\le} (\frac{\sqrt{mn}}{\sigma^2 \beta^2} + \lambda_2 \sqrt{m}) \|L_1 - L_2\|_F \\
&\overset{(h)}{=} \frac{1}{\tau_L(E^t)} \|L_1 - L_2\|_F,
\end{aligned} \quad (60)
$$

where $\nabla^2_{XX} F$ is the second order derivative of the function F and $\nabla^2_{XX} F(\bar{L})$ in (e) comes from the differential mean value theorem. (f) follows from the inequality $\|\cdot\|_2 \le \|\cdot\|_F$. Note that (4) is lower bounded by β, and the probability density function of the normal distribution and its derivative are upper bounded by $\frac{1}{\sqrt{2\pi}\sigma}$ and $\frac{e^{-1/2}}{\sqrt{2\pi}\sigma^2}$, respectively. Then one can easily check that $\|\nabla^2_{XX} F(\bar{L})\|_\infty$ is bounded by a positive constant $\frac{1}{\sigma^2\beta^2}$. (g) is thus obtained by upper bounding $\|\nabla^2_{XX} F(\bar{L})\|_\infty$ by $\frac{1}{\sigma^2\beta^2}$. (h) follows from (47). Thus, $\tau_L(E^t) \le \frac{1}{L_{p3}^{t+1}}$.

$$
\begin{aligned}
&\|\nabla_E H(U^{t+1},V^{t+1},L^{t+1},E_1,C^t)-\\
&\quad\nabla_E H(U^{t+1},V^{t+1},L^{t+1},E_2,C^t)\|_F\\
&=\|\nabla_X F(L^{t+1},E_1)-\nabla_X F(L^{t+1},E_2)\|_F\\
&\overset{\text{(i)}}{=}\|\nabla^2_{XX}F(\bar{E})(E_1-E_2)\|_F\\
&\le\|\nabla^2_{XX}F(\bar{E})\|_F\|E_1-E_2\|_F\\
&\le\sqrt{mn}\|\nabla^2_{XX}F(\bar{E})\|_\infty\|E_1-E_2\|_F\\
&\overset{\text{(j)}}{\le}\frac{\sqrt{mn}}{\sigma^2\beta^2}\|E_1-E_2\|_F\\
&\overset{\text{(k)}}{=}\frac{1}{\tau_E(L^{t+1})}\|E_1-E_2\|_F,
\end{aligned}
\tag{61}
$$

where (i) follows from the differential mean value theorem. (j) is obtained by upper bounding $\|\nabla^2_{XX}F(\bar{E})\|_\infty$ by $\frac{1}{\sigma^2\beta^2}$. (k) follows from (48). (61) implies that $\tau_E(L^{t+1})=\frac{\sigma^2\beta^2}{\sqrt{mn}}\le\frac{1}{L^{t+1}_{p4}}$.

$$
\begin{aligned}
&\|\nabla_C H(U^{t+1},V^{t+1},L^{t+1},E^{t+1},C_1)-\\
&\quad\nabla_C H(U^{t+1},V^{t+1},L^{t+1},E^{t+1},C_2)\|_F\\
&=\|\lambda_1 V^{t+1}(V^{t+1})^T(C_1-C_2)\|_F\\
&\le\|\lambda_1 V^{t+1}(V^{t+1})^T\|_F\|C_1-C_2\|_F\\
&\overset{\text{(l)}}{=}\frac{1}{\tau_C(V^{t+1})}\|C_1-C_2\|_F,
\end{aligned}
\tag{62}
$$

where (l) follows from (49). Thus $\tau_C(V^{t+1})\le\frac{1}{L^{t+1}_{p5}}$.

Based on Definition 3, (57)-(62) guarantee the Lipschitz differentiable of $H(U,V,L,E,C)$, and give the Lipschitz constants as well as the step sizes of the Sparse-APA.

REFERENCES

[1] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[2] S. A. Bhaskar, "Probabilistic low-rank matrix recovery from quantized measurements: Application to image denoising," in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, 2015, pp. 541–545.

[3] ——, "Localization from connectivity: A 1-bit maximum likelihood approach," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2939–2953, 2016.

[4] ——, "Probabilistic low-rank matrix completion from quantized measurements," *The Journal of Machine Learning Research*, vol. 17, no. 60, pp. 1–34, 2016.

[5] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[6] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[7] S. P. Boyd and L. Vandenberghe, *Convex optimization*. New York, NY, USA: Cambridge University Press, 2004.

[8] T. Cai and W.-X. Zhou, "A max-norm constrained minimization approach to 1-bit matrix completion," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3619–3647, 2013.

[9] Y. Cao and Y. Xie, "Categorical matrix completion," in *Proceedings of the IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2015, pp. 369–372.

[10] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3973–3985, 2011.

[11] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Transactions on Electronic Computers*, no. 3, pp. 326–334, 1965.

[12] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters, "1-bit matrix completion," *Information and Inference*, vol. 3, no. 3, pp. 189–223, 2014.

[13] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[14] B. Eriksson, L. Balzano, and R. Nowak, "High-rank matrix completion," in *Proceedings of the Conference on Artificial Intelligence and Statistics*, 2012, pp. 373–381.

[15] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3818–3825.

[16] A. V. Fiacco and G. P. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1990.

[17] P. Gao, M. Wang, J. H. Chow, M. Berger, and L. M. Seversky, "Missing data recovery for high-dimensional signals with nonlinear low-dimensional structures," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5421–5436, 2017.

[18] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1006–1013, 2016.

[19] P. Gao, R. Wang, M. Wang, and J. H. Chow, "Low-rank matrix recovery from noisy, quantized and erroneous measurements," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2918–2932, 2018.

[20] A. S. Georghiades, B. Peter N, and K. David J, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Berlin, Germany: Springer, 2001.

[22] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.

[23] P. Ji, I. D. Reid, R. Garg, H. Li, and M. Salzmann, "Low-rank kernel subspace clustering," *arXiv preprint arXiv:1707.04974*, 2017

[24] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 24–33.

[25] O. Klopp, J. Lafond, É. Moulines, and J. Salmon, "Adaptive multinomial matrix completion," *Electronic Journal of Statistics*, vol. 9, no. 2, pp. 2950–2975, 2015.

[26] L. I. Kuncheva, J. J. Rodríguez, C. O. Plumpton, D. E. Linden, and S. J. Johnston, "Random subspace ensembles for fMRI classification," *IEEE Transactions on Medical Imaging*, vol. 29, no. 2, pp. 531–542, 2010.

[27] J. Lafond, O. Klopp, E. Moulines, and J. Salmon, "Probabilistic low-rank matrix completion on finite alphabets," in *Advances in Neural Information Processing Systems*, 2014, pp. 1727–1735.

[28] A. S. Lan, C. Studer, and R. G. Baraniuk, "Matrix recovery from quantized and corrupted measurements," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4973–4977.

[29] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk, "Sparse factor analysis for learning and content analytics." *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1959–2008, 2014.

[30] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[31] G. Lerman and T. Zhang, "Robust recovery of multiple subspaces by geometric lp minimization," *The Annals of Statistics*, vol. 39, no. 5, pp. 2686–2715, 2011.

[32] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[33] G. Liu, Q. Liu, and P. Li, "Blessing of dimensionality: Recovering mixture data via dictionary pursuit," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 47–60, 2017.

[34] G. Liu, H. Xu, J. Tang, Q. Liu, and S. Yan, "A deterministic analysis for LRR," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 417–430, 2016.

[35] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

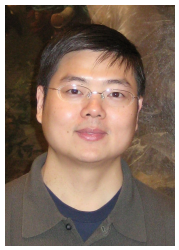
[36] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 347–360.
[37] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2002, pp. 849–856.
[38] N. Parikh and B. Stephen, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
[39] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proceedings of the IEEE International Conference on Image Processing*, 2014, pp. 2849–2853.
[40] V. M. Patel, H. Van Nguyen, and R. Vidal, "Latent space sparse and low-rank subspace clustering," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 691–701, 2015.
[41] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2016, pp. 1925–1931.
[42] M. Rahmani and G. K. Atia, "Innovation pursuit: A new approach to subspace clustering," *IEEE Transactions on Signal Processing*, vol. 65, no. 23, pp. 6276–6291, 2017.
[43] A. Reinhardt, F. Englert, and D. Christin, "Enhancing user privacy by preprocessing distributed smart meter data," in *Proceedings of the Conference on Sustainable Internet and ICT for Sustainability*, 2013, pp. 1–7.
[44] J. A. Snyman, N. Stander, and W. J. Roux, "A dynamic penalty function method for the solution of structural optimization problems," *Applied Mathematical Modelling*, vol. 18, no. 8, pp. 453–460, 1994.
[45] M. Soltanolkotabi and E. J. Candès, "A geometric analysis of subspace clustering with outliers," *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.
[46] M. Soltanolkotabi, E. Elhamifar, and E. J. Candés, "Robust subspace clustering," *The Annals of Statistics*, vol. 42, no. 2, pp. 669–699, 2014.
[47] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.
[48] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.
[49] S. Xiong, A. D. Sarwate, and N. B. Mandayam, "Randomized requantization with local differential privacy," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 2189–2193.
[50] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.
[51] D. Zhou, I. Councill, H. Zha, and C. L. Giles, "Discovering temporal communities from social network documents," in *Proceedings of the IEEE 7th International Conference on Data Mining*, 2007, pp. 745–750.
[52] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, 2013.

Ren Wang (S'16) received the B.E. degree and M.S. degree in Electrical Engineering from Tsinghua University, Beijing, China, in 2013 and 2016.

He is pursuing the Ph.D. degree in Electrical Engineering at Rensselaer Polytechnic Institute, Troy, NY. His research interests include unsupervised learning, optimization and their applications in power systems.

Meng Wang (M'12) received B.S. and M.S. degrees from Tsinghua University, China, in 2005 and 2007, respectively. She received the Ph.D. degree from Cornell University, Ithaca, NY, USA, in 2012.

She is an Assistant Professor in the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute, Troy, NY, USA. Her research interests include high-dimensional data analytics, machine learning, power systems monitoring, and synchrophasor technologies.



Jinjun Xiong (M'06) received the Ph.D. degree in Electrical Engineering from University of California, Los Angeles, in 2006.

He is a program director for cognitive computing systems research at the IBM T.J. Watson Research Center and a codirector of the IBM-Illinois Center for Cognitive Computing Systems Research. His research interests include cognitive computing, big data analytics, deep learning, smarter energy, and application of cognitive computing for industrial solutions. He has received two Best Paper Awards, one Best Paper in Track Award, and seven nominations for Best Paper Awards. He is an IBM Master Inventor and has received various IBM technical achievement awards and the IEEE Region One Outstanding Technical Contribution Award.