Sketched Subspace Clustering

Panagiotis A. Traganitis, *Student Member, IEEE*, and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—The immense amount of daily generated and communicated data presents unique challenges in their processing. Clustering, the grouping of data without the presence of ground-truth labels, is an important tool for drawing inferences from data. Subspace clustering (SC) is a relatively recent method that is able to successfully classify nonlinearly separable data in a multitude of settings. In spite of their high clustering accuracy, SC methods incur prohibitively high computational complexity when processing large volumes of high-dimensional data. Inspired by random sketching approaches for dimensionality reduction, the present paper introduces a randomized scheme for SC, termed Sketch-SC, tailored for large volumes of high-dimensional data. Sketch-SC accelerates the computationally heavy parts of state-of-the-art SC approaches by compressing the data matrix across both dimensions using random projections, thus enabling fast and accurate large-scale SC. Performance analysis as well as extensive numerical tests on real data corroborate the potential of Sketch-SC and its competitive performance relative to state-of-the-art scalable SC approaches.**

***Index Terms*—Subspace clustering, big data, random projections, sketching.**

I. Introduction

The permeation of the Internet and social networks into our daily life, as well as the ever increasing number of connected devices and highly accurate instruments, has trademarked society and computing research with a “data deluge.” Naturally, it is desirable to extract information and inferences from the available data. However, the sheer amount of data and their potentially large dimensionality introduces numerous challenges in their processing and analysis, as traditional statistical inference and machine learning processes do not necessarily scale. As the cost of cloud computing is declining, traditional approaches have to be redesigned to take advantage of the flexibility provided by distributed computing across multiple nodes as well as decreasing the computational burden per node, since in many cases each computing node might be an inexpensive machine.

Clustering (a.k.a. unsupervised classification) is a method of grouping data, without having labels available. Also referred to as graph partitioning or community identification, it finds applications in data mining, signal processing, and machine learning. Arguably, the most popular clustering algorithm is K-means due to its simplicity [1]. However, K-means, as well as its kernel-based variants, provide meaningful clustering results only when data, after mapped to an appropriate feature space, form “tight” groups that can be separated by hyperplanes [1].

Subspace clustering (SC) on the other hand, is a popular method for clustering nonlinearly separable data which are generated by a union of (affine) subspaces in a high-dimensional Euclidean space [2]. SC has well-documented impact in applications, as diverse as image and video segmentation, and identification of switching linear systems in controls [2]. Recent advances advocate SC with high clustering performance at the price of high computational complexity [2].

The goal of this paper is to introduce a randomized scheme for reducing the computational burden of SC algorithms when the number of data, and possibly their dimensionality, is prohibitively large, while maintaining high levels of clustering accuracy. Building on random projection (RP) methods, that have been used for dimensionality reduction [3], [4], the present paper employs RP matrices to *sketch* and *compress* the available data to a computationally affordable level, while also reducing drastically the number of optimization variables. In doing so, the proposed method markedly broadens the applicability of high-performing SC algorithms to the big data regime.

Moreover, the present contribution analyzes the performance of Sketch-SC, by leveraging the well-established theory of random matrices and Johnson-Lindenstrauss transforms [3], [5]. To assess the proposed Sketch-SC scheme, extensive numerical tests on real data are presented, comparing the proposed approach to state-of-the-art SC and large-scale SC methods [6], [7]. Compared to our conference precursor in [8], comprehensive numerical tests are included here, along with a rigorous performance analysis.

The rest of the paper is organized as follows. Section II provides SC preliminaries along with notation and prior art. Section III introduces the proposed Sketch-SC scheme for large-scale datasets, while Section IV provides pertinent performance bounds. Section V presents numerical tests conducted to evaluate the performance of Sketch-SC in comparison with state-of-the-art SC and large-scale SC algorithms. Finally, concluding remarks and future research directions are given in Section VI. Proofs of theorems and propositions as well as supporting lemmata are included in Appendix A.

Notation: Unless otherwise noted, lowercase bold letters $\boldsymbol{x}$ denote vectors, uppercase bold letters $\mathbf{X}$ represent matrices, and calligraphic uppercase letters $\mathcal{X}$ stand for sets. The (i, j)th entry of matrix $\mathbf{X}$ is denoted by $[\mathbf{X}]_{ij}$; $\text{rank}(\mathbf{X})$ and $\text{range}(\boldsymbol{X})$ denote the rank and column span of a matrix $\mathbf{X}$, respectively; and $\mathbf{X} = \mathbf{U}_\rho \mathbf{\Sigma}_\rho \mathbf{V}_\rho^\top$ denotes the singular value decomposition

Manuscript received July 22, 2017; revised October 26, 2017; accepted November 17, 2017. Date of publication December 8, 2017; date of current version February 13, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sotirios Chatzis. This work was supported by the National Science Foundation under Grants 1500713 and 1514056. This paper was presented in part at the 50th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, November 2016. *(Corresponding author: Georgios B. Giannakis.)*

The authors are with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: traga003@umn.edu; georgios@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2781649

(SVD) of a rank ρ, $D \times N$ matrix $\mathbf{X}$, where $\mathbf{U}_\rho$ is $D \times \rho$, $\boldsymbol{\Sigma}_\rho$ is $\rho \times \rho$, and $\mathbf{V}_\rho$ is $N \times \rho$. For a positive integer $r < \rho$, the SVD of $\mathbf{X}$ can be rewritten as

$$\begin{aligned}\mathbf{X} = \mathbf{U}_\rho \boldsymbol{\Sigma}_\rho \mathbf{V}_\rho^\top &= [\mathbf{U}_r \bar{\mathbf{U}}_r] \begin{bmatrix} \boldsymbol{\Sigma}_r & \\ & \bar{\boldsymbol{\Sigma}}_r \end{bmatrix} \begin{bmatrix} \mathbf{V}_r^\top \\ \bar{\mathbf{V}}_r^\top \end{bmatrix} \\ &= \mathbf{X}_r + \bar{\mathbf{X}}_r \end{aligned} \tag{1}$$

where $\boldsymbol{\Sigma}_r$ is an $r \times r$ diagonal matrix with the largest r singular values of $\mathbf{X}$ in descending order, and $\mathbf{X}_r = \mathbf{U}_r \boldsymbol{\Sigma}_r \mathbf{V}_r^\top$ is the best rank-r approximation of $\mathbf{X}$ in the sense that $\mathbf{X}_r$ minimizes $\|\mathbf{X} - \mathbf{X}_r\|_F$. Accordingly, $\bar{\boldsymbol{\Sigma}}_r$ is a $(\rho - r) \times (\rho - r)$ diagonal matrix containing the remaining singular values of $\mathbf{X}$ and $\bar{\mathbf{X}}_r = \bar{\mathbf{U}}_r \bar{\boldsymbol{\Sigma}}_r \bar{\mathbf{V}}_r^\top$. The D-dimensional real Euclidean space is denoted by $\mathbb{R}^D$, the set of positive real numbers by $\mathbb{R}_+$, the set of positive integers by $\mathbb{Z}_+$, the expectation operator by $\mathbb{E}[\cdot]$, and the ℓ_2-norm by $\|\cdot\|$.

II. Preliminaries

A. SC Problem Statement

Consider N vectors $\{\boldsymbol{x}_i\}_{i=1}^N$ of size $D \times 1$ drawn from a union of K affine subspaces, each denoted by $\mathcal{S}_k$, adhering to the model

$$\boldsymbol{x}_i = \mathbf{C}^{(k)} \boldsymbol{y}_i^{(k)} + \boldsymbol{\mu}^{(k)} + \boldsymbol{v}_i\,, \quad \forall \boldsymbol{x}_i \in \mathcal{S}_k \tag{2}$$

where d_k (possibly with $d_k \ll D$) is the dimensionality of $\mathcal{S}_k$; $\mathbf{C}^{(k)}$ is a $D \times d_k$ matrix whose columns form a basis of $\mathcal{S}_k$; the d_k-dimensional vector $\boldsymbol{y}_i^{(k)}$ is the low-dimensional representation of $\boldsymbol{x}_i$ in $\mathcal{S}_k$ with respect to (w.r.t.) $\mathbf{C}^{(k)}$; the $D \times 1$ vector $\boldsymbol{\mu}^{(k)}$ is the "centroid" or intercept of $\mathcal{S}_k$; and, $\boldsymbol{v}_i$ denotes the $D \times 1$ noise vector capturing unmodeled effects. If $\mathcal{S}_k$ is linear, then $\boldsymbol{\mu}^{(k)} = \mathbf{0}$. Let also $\boldsymbol{\pi}_i$ denote the cluster assignment vector of $\boldsymbol{x}_i$, and $[\boldsymbol{\pi}_i]_k$ the kth entry of $\boldsymbol{\pi}_i$ that is constrained to satisfy $[\boldsymbol{\pi}_i]_k \geq 0$ and $\sum_{k=1}^K [\boldsymbol{\pi}_i]_k = 1$. If $\boldsymbol{\pi}_i \in \{0,1\}^K$, then $\boldsymbol{x}_i$ lies in only one subspace (hard clustering), while if $\boldsymbol{\pi}_i \in [0,1]^K$, then $\boldsymbol{x}_i$ can belong to multiple clusters (soft clustering). In the latter case, $[\boldsymbol{\pi}_i]_k$ can be thought of as the probability that $\boldsymbol{x}_i$ belongs to $\mathcal{S}_k$. Clearly in the case of hard clustering, (2) can be rewritten as

$$\boldsymbol{x}_i = \sum_{k=1}^K [\boldsymbol{\pi}_i]_k \left(\mathbf{C}^{(k)} \boldsymbol{y}_i^{(k)} + \boldsymbol{\mu}^{(k)} \right) + \boldsymbol{v}_i. \tag{3}$$

Given the $D \times N$ data matrix $\mathbf{X} := [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ and the number of subspaces K, the goal is to find the data-to-subspace assignment vectors $\{\boldsymbol{\pi}_i\}_{i=1}^N$, the subspace bases $\left\{\mathbf{C}^{(k)}\right\}_{k=1}^K$, their dimensions $\{d_k\}_{k=1}^K$, the low-dimensional representations $\{\boldsymbol{y}_i^{(k)}\}_{i=1}^N$, as well as the centroids $\{\boldsymbol{\mu}^{(k)}\}_{k=1}^K$ [2]. SC can be formulated as follows

$$\begin{aligned} \min_{\boldsymbol{\Pi}, \{\mathbf{C}^{(k)}\}, \{\boldsymbol{y}_i^{(k)}\}, \mathbf{M}} \quad & \sum_{k=1}^K \sum_{i=1}^N [\boldsymbol{\pi}_i]_k \|\boldsymbol{x}_i - \mathbf{C}^{(k)} \boldsymbol{y}_i^{(k)} - \boldsymbol{\mu}^{(k)}\|_2^2 \\ \text{subject to (s.to)} \quad & \boldsymbol{\Pi}^\top \mathbf{1} = \mathbf{1}; \quad [\boldsymbol{\pi}_i]_k \geq 0,\ \forall (i,k) \end{aligned} \tag{4}$$

where $\boldsymbol{\Pi} := [\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N]$, $\mathbf{M} := [\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \ldots, \boldsymbol{\mu}^{(k)}]$, and $\mathbf{1}$ denotes the all-ones vector of matching dimensions.

The problem in (4) is non-convex as all of $\boldsymbol{\Pi}, \{\mathbf{C}^{(k)}\}_{k=1}^K$, $\{d_k\}_{k=1}^K, \{\boldsymbol{y}_i^{(k)}\}$, and $\mathbf{M}$ are unknown. It is known that when $K = 1$ and $\mathbf{C}$ is orthonormal, (4) boils down to PCA [9]

$$\begin{aligned} \min_{\mathbf{C}, \{\boldsymbol{y}_i\}, \boldsymbol{\mu}} \quad & \sum_{i=1}^N \|\boldsymbol{x}_i - \mathbf{C} \boldsymbol{y}_i - \boldsymbol{\mu}\|_2^2 \\ \text{s.to} \quad & \mathbf{C}^\top \mathbf{C} = \mathbf{I} \end{aligned} \tag{5}$$

where $\mathbf{I}$ denotes the identity matrix of appropriate dimension. Notice that for $K = 1$, it holds that $[\boldsymbol{\pi}_i]_k = 1$. Moreover, if $\mathbf{C}^{(k)} := \mathbf{0}$, $\forall k$, looking for $\{\boldsymbol{\mu}^{(k)}\}_{k=1}^K$, $\{\boldsymbol{\pi}_i\}_{i=1}^N$ with $K > 1$, amounts to K-means clustering

$$\begin{aligned} \min_{\boldsymbol{\Pi}, \mathbf{M}} \quad & \sum_{k=1}^K \sum_{i=1}^N [\boldsymbol{\pi}_i]_k \|\boldsymbol{x}_i - \boldsymbol{\mu}^{(k)}\|_2^2 \\ \text{s.to} \quad & \boldsymbol{\Pi}^\top \mathbf{1} = \mathbf{1}\,. \end{aligned} \tag{6}$$

B. Prior Work

Various algorithms have been developed by the machine learning [2] and data-mining community [10] to solve (4). Generalizing the ubiquitous K-means [11] the K-subspaces algorithm [12] builds on alternating optimization to solve (4). For $\boldsymbol{\Pi}$ and $\{d_k\}_{k=1}^K$ fixed, bases of the subspaces can be recovered using the SVD on the data associated with each subspace. Indeed, given $\mathbf{X}^{(k)} := [\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{N_k}}]$, belonging to $\mathcal{S}_k$ ($\sum_{k=1}^K N_k = N$), a basis $\mathbf{C}^{(k)}$ can be obtained from the first d_k (from the left) singular vectors of $\mathbf{X}^{(k)} - [\boldsymbol{\mu}^{(k)}, \ldots, \boldsymbol{\mu}^{(k)}]$, where $\boldsymbol{\mu}^{(k)} = (1/N_k) \sum_{i \in \mathcal{S}_k} \boldsymbol{x}_i$. On the other hand, when $\{\mathbf{C}^{(k)}, \boldsymbol{\mu}^{(k)}\}_{k=1}^K$ are given, the assignment matrix $\boldsymbol{\Pi}$ can be recovered in the case of hard clustering by finding the closest subspace to each datapoint; that is, $\forall i \in \{1, 2, \ldots, N\}$, $\forall k \in \{1, \ldots, K\}$, we obtain

$$[\boldsymbol{\pi}_i]_k = \begin{cases} 1, & \text{if } k = \displaystyle\arg\min_{k' \in \{1, \ldots, K\}} \left\| \tilde{\boldsymbol{x}}_i^{(k')} - \mathbf{C}^{(k')} \mathbf{C}^{(k')^\top} \tilde{\boldsymbol{x}}_i^{(k')} \right\|_2^2 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\tilde{\boldsymbol{x}}_i^{(k)} := \boldsymbol{x}_i - \boldsymbol{\mu}^{(k)}$ and $\|\tilde{\boldsymbol{x}}_i^{(k)} - \mathbf{C}^{(k)} \mathbf{C}^{(k)^\top} \tilde{\boldsymbol{x}}_i^{(k)}\|_2$ is the distance of $\boldsymbol{x}_i$ from $\mathcal{S}_k$. Thus, the K-subspaces algorithm operates as follows: (i) Fix $\boldsymbol{\Pi}$ and solve for the remaining unknowns; and (ii) fix $\{\mathbf{C}^{(k)}, \boldsymbol{\mu}^{(k)}\}_{k=1}^K$, and solve for $\boldsymbol{\Pi}$. Since SVD is involved, SC entails high computational complexity, whenever d_k and/or N_k are massive.

A probabilistic (soft) counterpart of K-subspaces is the mixture of probabilistic PCA [13], which assumes that data are drawn from a mixture of degenerate (zero-variance) Gaussians. Building on the same assumption, the agglomerative lossy compression (ALC) minimizes the required number of bits to "encode" each cluster, up to a certain distortion level [14]. Algebraic schemes, such as generalized (G)PCA approach SC from a linear algebra point of view, but generally their performance is guaranteed only for independent and noise-less subspaces [15]. Additional interesting methods recover subspaces by finding local linear subspace approximations [16]; by thresholding the correlations between data [17]; or by identifying the subspaces one by one [18]. Recently, multilinear methods for SC of tensor data have also been advocated [19]; see also [20]–[22] for online clustering approaches to handle streaming data.

Arguably the most successful class of solvers for (4) relies on *spectral clustering* [23] to find the data-to-subspace assignments. Algorithms in this class generate first an $N \times N$ symmetric weighted adjacency matrix $\mathbf{W}$ to capture the non-directional similarity between data vectors, and then perform spectral clustering on $\mathbf{W}$. Matrix $\mathbf{W}$ implies a graph $\mathcal{G}$ whose vertices correspond to data and the weight of the edge connecting vertex i and vertex j is given by $[\mathbf{W}]_{ij}$. Spectral clustering algorithms form the graph Laplacian matrix

$$\mathbf{L} := \text{diag}(\mathbf{W}\mathbf{1}) - \mathbf{W} \tag{8}$$

where diag$(\mathbf{W}\mathbf{1})$ is a diagonal matrix holding $\mathbf{W}\mathbf{1}$ on its diagonal. The algebraic multiplicity of the 0 eigenvalue of $\mathbf{L}$ yields the number of connected components in $\mathcal{G}$, while the corresponding eigenvectors are indicator vectors of these connected components [23]. Afterwards, having formed $\mathbf{L}$, the K eigenvectors $\{\mathbf{v}_k\}_{k=1}^K$ corresponding to the trailing eigenvectors of $\mathbf{L}$ are found, and K-means is performed on the rows of the $N \times K$ matrix $\mathbf{V} := [\mathbf{v}_1, \ldots, \mathbf{v}_K]$ to obtain clustering assignments [23].

Sparse subspace clustering (SSC) [24] exploits the fact that under the union of subspaces model (4), only a small percentage of data suffices to provide a low-dimensional affine representation of $\boldsymbol{x}_i$; that is, $\boldsymbol{x}_i = \sum_{j=1, j\neq i}^N w_{ij}\boldsymbol{x}_j$, $\forall i \in \{1, 2, \ldots, N\}$. Specifically, SSC solves the following sparsity-promoting optimization problem

$$\begin{aligned} &\min_{\mathbf{Z}} \quad \|\mathbf{Z}\|_1 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 \\ &\text{s.to} \quad \mathbf{Z}^\top\mathbf{1} = \mathbf{1}; \quad \text{diag}(\mathbf{Z}) = \mathbf{0} \end{aligned} \tag{9}$$

where $\mathbf{Z} := [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N]$; column $\boldsymbol{z}_i$ is sparse and contains the coefficients for the representation of $\boldsymbol{x}_i$; $\lambda > 0$ is the regularization coefficient; and $\|\mathbf{Z}\|_1 := \sum_{i,j=1}^N |[\mathbf{Z}]_{i,j}|$. The constraint diag$(\mathbf{Z}) = \mathbf{0}$ ensures that the solution of the optimization problem is not a trivial one ($\mathbf{Z} = \mathbf{I}$), while $\mathbf{Z}^\top\mathbf{1} = \mathbf{1}$ is employed to guarantee that the $\mathbf{Z}$ found is invariant to shifting the data by a constant vector [2]. Matrix $\mathbf{Z}$ is used to create the weighted adjacency matrix $[\mathbf{W}]_{ij} := |[\mathbf{Z}]_{ij}| + |[\mathbf{Z}]_{ji}|$. Finally, spectral clustering, is performed on $\mathbf{W}$ and cluster assignments are identified. Using those assignments, $\mathbf{M}$ is found by taking sample means per cluster, and $\{\mathbf{C}^{(k)}\}_{k=1}^K$, $\{\boldsymbol{y}_i^{(k)}\}_{i=1}^N$ are obtained by applying SVD on $\mathbf{X}^{(k)} - [\boldsymbol{\mu}^{(k)}, \ldots, \boldsymbol{\mu}^{(k)}]$.

The low-rank representation (LRR) approach to SC is similar to SSC, but replaces the ℓ_1-norm in (9) with the nuclear one: $\|\mathbf{Z}\|_* := \sum_{i=1}^\rho \sigma_i(\mathbf{Z})$, where ρ stands for the rank and $\sigma_i(\mathbf{Z})$ for the ith singular value of $\mathbf{Z}$. Specifically, LRR solves the following optimization problem [25]

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_{2,1} \tag{10}$$

where $\|\mathbf{X}\|_{2,1} := \sum_{j=1}^N \|\boldsymbol{x}_j\|_2$, and $\boldsymbol{x}_j$ denotes the j-th column of $\mathbf{X}$.

Another popular algorithm is termed least-squares regression (LSR) [26]. It solves an optimization problem similar to (10), but replaces the ℓ_1/nuclear norm with the Frobenius one. Specifically, LSR solves

$$\min_{\mathbf{Z}} \frac{1}{2}\|\mathbf{Z}\|_F^2 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{X}\mathbf{Z}\|_F^2 \tag{11}$$

which admits the following closed-form solution $\mathbf{Z}^* = \lambda\left(\lambda\mathbf{X}^\top\mathbf{X} + \mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{X}$. Combining SSC with LSR, the elastic net SC (EnSC) approaches employ a convex combination of ℓ_1- and Frobenius-norm regularizers [27], [28]. The high clustering accuracy achieved by these self-dictionary methods comes at the price of high complexity. Solving (9), (10) or (11) scales cubically with the number of data N, on top of performing spectral clustering across K clusters, which renders these methods computationally prohibitive for large-scale SC. When data are high-dimensional ($D \gg$), methods based on (statistical) leverage scores, random projections [4], [29]–[31], preconditioning and sampling [32], or our recent sketching and validation (SkeVa) [33] approach can be employed to reduce complexity to an affordable level. Random projection based methods left multiply the data matrix $\mathbf{X}$, with a $d \times D$ data-agnostic random matrix, thereby reducing the dimensionality of the data vectors from D to d. This type of dimensionality reduction has been shown to reduce computational costs while not incurring significant clustering performance degradation when $d = \mathcal{O}(\sum_{k=1}^K d_k)$ [29]. When the number of data vectors is large ($N \gg$), the scalable SSC/LRR/LSR approach [34] involves drawing randomly $n < N$ data, performing SSC/LRR/LSR on them, and expressing the rest of the data according to the clusters identified by that random draw of samples. While this approach clearly reduces complexity, performance can potentially suffer as the random sample may not be representative of the entire dataset, especially when $n \ll N$ and clusters are unequally populated. Other approaches focus on greedy methods, such as orthogonal matching pursuit (OMP), for solving (9) [6], [35]. More recently, an active set method, termed Oracle guided Elastic Net (ORGEN) [7], can be used to reduce the complexity of SSC and EnSC tasks, by solving only for the entries of $\mathbf{Z}$ that correspond to data vectors that are highly correlated.

The present paper introduces a novel approach based on random projections that creates a compact yet expressive dictionary that can be employed by SSC/LRR/LSR to reduce the number of optimization variables to $\mathcal{O}(nN)$ for $n < N$, thus yielding low computational complexity. In addition, the proposed approach can be combined with random projection methods to reduce data dimensionality, which further scales down computational costs.

III. Sketched Subspace Clustering

Consider the following unifying optimization problem

$$\min_{\mathbf{A} \in \mathcal{C}} h(\mathbf{A}) + \lambda L(\mathbf{X} - \mathbf{B}\mathbf{A}) \tag{12}$$

where $\mathbf{B}$ is an appropriate $D \times n$ known basis matrix (dictionary), $h(\mathbf{A})$ is a regularization function of the $n \times N$ matrix $\mathbf{A}$, $L(\cdot)$ is an appropriate loss function, and $\mathcal{C}$ is a constraint set for $\mathbf{A}$. Eq. (12) will henceforth be referred to as *Sketch-SC objective*. As mentioned in Sec. II-B, the ability of $\mathbf{A}$, obtained from (12) to distinguish data for clustering depends on the choice of $h(\cdot)$, and on $\mathbf{B}$. For SSC, LSR and LRR, $\mathbf{B} = \mathbf{X}$, $n = N$ and $h(\cdot)$ is $\|\cdot\|_1$, $\frac{1}{2}\|\cdot\|_F^2$, $\|\cdot\|_*$, and $L(\cdot)$ is $\frac{1}{2}\|\cdot\|_F^2$, $\frac{1}{2}\|\cdot\|_F^2$ and $\frac{1}{2}\|\cdot\|_F^2$ or $\frac{1}{2}\|\cdot\|_{2,1}$ respectively. The constraint set for SSC is $\mathcal{C} = \{\mathbf{A} \in \mathbb{R}^{N\times N} : \mathbf{A}^\top\mathbf{1} = \mathbf{1}; \text{diag}(\mathbf{A}) = \mathbf{0}\}$, while for LSR and LRR, we have $\mathcal{C} = \mathbb{R}^{N\times N}$.

```
Algorithm 1: Linear sketched data model for Sketch-SC.
Input: $D \times N$ data matrix $\mathbf{X}$; Number of columns of $\mathbf{R}$ $n$;
    regularization parameter $\lambda$;
Output: Model matrix $\mathbf{A}$;
  1: Generate $N \times n$ JLT matrix $\mathbf{R}$.
  2: Form $D \times n$ dictionary $\mathbf{B} = \mathbf{X}\mathbf{R}$.
  3: Solve (12) for the cost in (14), (15), (16) to obtain $\mathbf{A}$.
```

A. High Volume of Data

As the aim of the present manuscript is to introduce scalable methods for subspace clustering, the dictionaries considered from now on will have $n \ll N$, bringing the number of variables to $\mathcal{O}(nN)$. In particular, the dictionaries employed will have the form, $\mathbf{B} := \mathbf{X}\mathbf{R}$, where $\mathbf{R}$ is a $N \times n$ sketching matrix. The role of $\mathbf{R}$ is to "compress" $\mathbf{X}$, while retaining as much information from it as possible. To this end, the celebrated Johnson-Lindenstrauss lemma [5] will be invoked.

Lemma 1: [5] Given $\varepsilon > 0$, for any subset $\mathcal{V} \subset \mathbb{R}^N$ containing d vectors of size $N \times 1$, there exists a map $q : \mathbb{R}^N \to \mathbb{R}^n$ such that for $n \geq n_0 = \mathcal{O}(\varepsilon^{-2} \log d)$, it holds for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{V}$

$$(1-\varepsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \leq \|q(\boldsymbol{x}) - q(\boldsymbol{y})\|_2^2 \leq (1+\varepsilon)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2. \tag{13}$$

In particular, random matrices known as Johnson-Lindenstrauss transforms will be employed since they exhibit useful properties.

Definition 1: [3, Def. 2.3], [4] An $N \times n$ random matrix $\mathbf{R}$ forms a Johnson-Lindenstrauss transform (JLT(ε, δ, d)) with parameters ε, δ, d if there exists a function f, such that for any $\varepsilon > 0, \delta < 1, d \in \mathbb{Z}_+$ and d-element subset $\mathcal{V} \subset \mathbb{R}^N$, with $n = \Omega(\frac{\log d}{\varepsilon^2} f(\delta))$, it holds that

$$\Pr\left\{(1-\varepsilon)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{x}^\top \mathbf{R}\|_2^2 \leq (1+\varepsilon)\|\boldsymbol{x}\|_2^2\right\} \geq 1 - \delta$$

for any $1 \times N$ vector $\boldsymbol{x}^\top \in \mathcal{V}$.

One example of a random JLT matrix is a matrix with independent and identically distributed (i.i.d.) entries drawn from a normal $\mathcal{N}(0,1)$ distribution scaled by a factor $1/\sqrt{n}$ [3]. Rescaled random sign matrices, that is matrices with i.i.d. ± 1 entries multiplied by $1/\sqrt{n}$ are also JLTs [4], [36], and matrix products involving these matrices can be computed fast [37]. Another class of JLTs that allows for efficient matrix multiplication includes the so-called Fast (F)JLTs. This class of FJLTs samples randomly and rescales rows of a fixed orthonormal matrix, such as the discrete Fourier transform (DFT) matrix, or, the Hadamard matrix [38], [39]; see also [3], [40], [41] where sparse JLT matrices have been advocated.

The following proposition proved in the appendix justifies the use of JLTs for constructing our dictionary $\mathbf{B}$ in (12).

Proposition 1: Let $\mathbf{X}$ be a $D \times N$ matrix such that $\text{rank}(\mathbf{X}) = \rho$, and define the $D \times n$ matrix $\mathbf{B} := \mathbf{X}\mathbf{R}$, where $\mathbf{R}$ is a JLT(ε, δ, D) of size $N \times n$. If $n = \mathcal{O}(\rho \frac{\log(\rho/\varepsilon)}{\varepsilon^2} f(\delta))$ then w.p. at least $1 - \delta$, it holds that

$$\text{range}(\mathbf{X}) = \text{range}(\mathbf{B}).$$

This proposition asserts that with a proper choice of the sketching matrix $\mathbf{R}$, the dictionary $\mathbf{B}$ is as expressive as $\mathbf{X}$ for solving (12), as it preserves the column space of $\mathbf{X}$ with high probability. The next proposition provides a similar bound on the reduced dimension n, when $n < \text{rank}(\mathbf{X}) := \rho$.

Proposition 2: Let $\mathbf{X}$ be a $D \times N$ matrix such that $\text{rank}(\mathbf{X}) = \rho$, and define the $D \times n$ matrix $\mathbf{B} := \mathbf{X}\mathbf{R}$, where $\mathbf{R}$ is a JLT(ε, δ, D) of size $N \times n$. If $n = \mathcal{O}(r \frac{\log(r/\varepsilon)}{\varepsilon^2} f(\delta))$, then w.p. at least $1 - 2\delta$ it holds that

$$\|\mathbf{B}(\mathbf{V}_r^\top \mathbf{R})^\dagger - \mathbf{U}_r \boldsymbol{\Sigma}_r\|_F \leq \left(\varepsilon \frac{\sqrt{1+\varepsilon}}{\sqrt{1-\varepsilon}} + 1 + \varepsilon\right) \|\bar{\mathbf{X}}_r\|_F.$$

Prop. 2 suggests that $\mathbf{B}$ approximately inherits the range of $\mathbf{X}_r$.

Upon constructing a $\mathbf{B}$ adhering to Prop. 1 or Prop. 2, (12) can be solved for different choices of h. When $h(\mathbf{A}) = \frac{1}{2}\|\mathbf{A}\|_F^2$, the optimization task (termed henceforth *Sketch-LSR*)

$$\min_{\mathbf{A}} \frac{1}{2}\|\mathbf{A}\|_F^2 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{B}\mathbf{A}\|_F^2 \tag{14}$$

is solved by $\mathbf{A}^* = \lambda\left(\lambda \mathbf{B}^\top \mathbf{B} + \mathbf{I}\right)^{-1} \mathbf{B}^\top \mathbf{X}$, incurring complexity $\mathcal{O}(n^3 + n^2 D + nDN)$. Accordingly, our *Sketch-SSC* corresponds to $h(\mathbf{A}) = \|\mathbf{A}\|_1 = \sum_{ij} |[\mathbf{A}]_{ij}|$ and relies on the objective

$$\min_{\mathbf{A}} \|\mathbf{A}\|_1 + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{B}\mathbf{A}\|_F^2 \tag{15}$$

that can be solved efficiently to obtain $\mathbf{A}$ using the alternating direction method of multipliers (ADMM) [42], as per [24], or any other efficient LASSO solver. The ADMM solver for (15) incurs complexity $\mathcal{O}(n^3 + n^2 D + nDN + n^2 NI)$, where I is the required number of iterations until convergence, and the constraint $\text{diag}(\mathbf{A}) = \mathbf{0}$ is no longer required as $\mathbf{I}$ is not a trivial solution of (15). Proceeding along similar lines, our *Sketch-LRR* objective, for $h(\mathbf{A}) = \|\mathbf{A}\|_*$ aims at

$$\min_{\mathbf{A}} \|\mathbf{A}\|_* + \frac{\lambda}{2}\|\mathbf{X} - \mathbf{B}\mathbf{A}\|_F^2 \tag{16}$$

that can be solved using the augmented Lagrange multiplier (ALM) method of [25], which incurs complexity $\mathcal{O}(n^3 + n^2 D + nDN + (nDN + nN^2 + n^2 N)I)$, where I is the number of iterations until convergence. In addition, (16) can be solved using the $\ell_{2,1}$ norm instead of the Frobenius norm for the fitting term $\mathbf{X} - \mathbf{B}\mathbf{A}$. The entire process to obtain the data model $\mathbf{A}$ is outlined in Algorithm 1. Detailed algorithms for solving (15) and (16) are described in Appendix B.

Remark 1: An optimal data-driven choice of $\mathbf{R}$ would be interesting only if finding it incurs manageable complexity - a topic which goes beyond the scope of this submission and constitutes a worthy future research direction.

Remark 2: Upon computing $\mathbf{B}$, (14) and (15) can be readily parallelized across columns of $\mathbf{X}$. In the nuclear norm case of (16) one can employ the following identity [22], [43]

$$\|\mathbf{A}\|_* = \min_{\mathbf{Z} = \mathbf{P}\mathbf{Q}^\top} \frac{1}{2}(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) \tag{17}$$

where $\mathbf{A}$ is some $n \times N$ matrix of rank ρ and $\mathbf{P}$ and $\mathbf{Q}$ are $n \times \rho$ and $N \times \rho$ matrices respectively. This is especially useful when multiple computing nodes are available, or the data is scattered across multiple devices. Without (17), distributed solvers of (16) are challenged because as columns of $\mathbf{A}$ are added the SVD needed to find the nuclear norm has to be recomputed, which is not the case with (17).

```
Algorithm 2: Linear sketched data model for Sketch-SC and $D \gg$.
Input: $D \times N$ data matrix $\mathbf{X}$; Lower dimension $d$; Number
    of columns of $\mathbf{R}$ $n$; regularization parameter $\lambda$;
Output: Model matrix $\mathbf{A}$;
  Generate $d \times D$ JLT matrix $\check{\mathbf{R}}$.
  Generate $N \times n$ JLT matrix $\mathbf{R}$.
  Form $d \times N$ matrix $\check{\mathbf{X}} = \check{\mathbf{R}}\mathbf{X}$.
  Create $d \times n$ dictionary $\check{\mathbf{B}} = \check{\mathbf{X}}\mathbf{R}$.
  Solve (18) to obtain sketched data model $\mathbf{A}$.
```

Remark 3: Existing general guidelines for choosing the regularization parameter λ for SSC and LRR [24], [25] rely on cross-validation and apply also to the proposed Algorithm 1 and 2 here.

B. High-Dimensional Data

The complexity of all the aforementioned algorithms depends on the data dimensionality D. As such, datasets containing high-dimensional vectors will certainly increase the computational complexity. As mentioned in Sec. II-B, dimensionality reduction techniques can be employed to reduce the computational burden of SC approaches. Using PCA for instance, a $d < D$-dimensional subspace that describes most of the data variance can be found. This, however, can be prohibitively expensive for large-scale datasets where $N \gg$. For such cases, our idea is to combine the method described in the previous section with randomized dimensionality reduction techniques [29]. Let $\check{\mathbf{R}}$ be a $d \times D$ JLT matrix, where $d \ll D$ is the target dimensionality, and consider the $d \times N$ matrix $\check{\mathbf{X}} := \check{\mathbf{R}}\mathbf{X}$, which is a reduced dimensionality version of the original data $\mathbf{X}$. The Sketch-SC objective then becomes

$$\min_{\mathbf{A}} h(\mathbf{A}) + \lambda L(\check{\mathbf{X}} - \check{\mathbf{B}}\mathbf{A}) \tag{18}$$

where $\check{\mathbf{B}} := \check{\mathbf{X}}\mathbf{R}$ is a $d \times n$ dictionary of reduced dimension with $\mathbf{R}$ being an $N \times n$ JLT matrix as in (12). Upon forming $\check{\mathbf{X}}$ and $\check{\mathbf{B}}$, (18) can be solved for different choices of h as in Sec. III-A. The steps of our algorithm for high-dimensional data are summarized in Algorithm 2.

Remark 4: While carrying out the products $\mathbf{XR}$, $\check{\mathbf{R}}\mathbf{X}$ or $\check{\mathbf{X}}\mathbf{R}$ can be computationally expensive in cases, they can be accelerated using modern numerical linear algebra tools, such as the Mailman algorithm [37] or by employing the Welsh-Hadamard transform [32], [44].

C. Obtaining Cluster Assignments Using $\mathbf{A}$

After obtaining the $N \times N$ matrix $\mathbf{Z}$ in (9), (10) or (11), a typical post-processing step for SSC, LSR, and LRR, is to perform spectral clustering, using $\mathbf{W} := |\mathbf{Z}| + |\mathbf{Z}^\top|$ as the adjacency matrix. This step however, is not possible for the matrix $\mathbf{A}$ obtained from (14), (15) or (16), because it has size $n \times N$, with $n < N$.

While $\mathbf{A}$ cannot be directly used for spectral clustering, a k-nearest neighbor graph [1] can be constructed from the columns of $\mathbf{A}$. Let $\boldsymbol{a}_i$ denote the i-th column of $\mathbf{A}$, and $\mathcal{K}_i$ the set of the k columns of $\mathbf{A}$ that are closest to $\boldsymbol{a}_i$, in the Euclidean distance sense. The $N \times N$ adjacency matrix $\mathbf{W}$ can then be constructed with entries

$$[\mathbf{W}]_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{a}_j \in \mathcal{K}_i \text{ or } \boldsymbol{a}_i \in \mathcal{K}_j \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

In addition, non-binary edge weights can be assigned as

$$[\mathbf{W}]_{ij} = \begin{cases} w_{ij}, & \text{if } \boldsymbol{a}_j \in \mathcal{K}_i \text{ or } \boldsymbol{a}_i \in \mathcal{K}_j \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

where w_{ij} is some scalar that depends on $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$. For instance, if heat kernel weights are used, then $w_{ij} = \exp(-\|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2^2/\sigma^2)$, for some $\sigma > 0$. The resultant mutual k-nearest neighbor matrix $\mathbf{W}$ can then be employed for spectral clustering. Note that the $N \times N$ matrix $\mathbf{W}$ emerging from (19) or (20) will be sparse with $\mathcal{O}(N)$ nonzero entries, which can accelerate the eigendecomposition schemes employed for spectral clustering [45], [46]. The overall scheme is tabulated in Algorithm 3.

```
Algorithm 3: Obtaining clustering assignments from $\mathbf{A}$.
Input: $n \times N$ matrix $\mathbf{A}$; Number of nearest neighbors $k$;
    Number of clusters $K$
Output: Clustering assignments
  Find $k$-nearest neighbors for each column of $\mathbf{A}$.
  Create matrix $\mathbf{W}$ using (19) or (20).
  Apply spectral clustering on $\mathbf{W}$.
```

Remark 5: When N and n are large, computation of the k nearest neighbors can be computationally taxing. Many efficient algorithms are available to accelerate the construction of the k nearest neighbor graph [47], [48]. In addition, approximate nearest neighbor (ANN) methods [49]–[51] can be employed to speed up the post-processing step even further. Finally, this post-processing step can be employed for regular SSC, LSR, and LRR.

IV. Performance Analysis

In this section, performance of the proposed method will be quantified analytically. Albeit not the tightest, the bounds to be derived will provide nice intuition on why the proposed methods work. The following theorem bounds the representation error of Sketch-LSR in the noise less case.

Theorem 1: Consider noise-free and normalized data vectors obeying (3) with $\boldsymbol{v}_i \equiv \mathbf{0}$, to form columns of a $D \times N$ data matrix $\mathbf{X}$, with unit ℓ_2 norm per column, and rank$(\mathbf{X}) = \rho$. Let also $\mathbf{R}$ denote a JLT(ε, δ, D) of size $N \times n$. Let $\boldsymbol{g}^*(\boldsymbol{x}) := \mathbf{X}\boldsymbol{z}^* = \boldsymbol{x}$ denote the representation of $\boldsymbol{x}$ provided by LSR, and $\hat{\boldsymbol{g}}(\boldsymbol{x}) := \mathbf{XR}\hat{\boldsymbol{a}}$ the representation given by Sketch-LSR. If $n = \mathcal{O}(r\frac{\log(r/\varepsilon)}{\varepsilon^2}f(\delta))$, then the following bound holds w.p. at least $1 - 2\delta$

$$\|\boldsymbol{g}^*(\boldsymbol{x}) - \hat{\boldsymbol{g}}(\boldsymbol{x})\|_2 \leq \lambda\ (1 + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\ \sqrt{\rho - r}\ \sigma_{r+1}^2) + \frac{1}{\sqrt{1+\varepsilon}}$$

with λ as in (12), and σ_{r+1} denotes the $(r+1)$st singular value of $\mathbf{X}$.

Theorem 1 implies that the larger n is, the smaller the upper bound becomes as a smaller singular value of $\mathbf{X}$ is selected. This also suggests that datasets exhibiting lower rank can be compressed more (with smaller n), while retaining representation accuracy. The following corollaries extend the result of Theorem 1 to the Sketch-SSC and Sketch-LRR cases.

Corollary 1: Consider the setting of Theorem 1, and let $\hat{\boldsymbol{g}}(\boldsymbol{x}) := \mathbf{X}\mathbf{R}\hat{\boldsymbol{a}}$ be the representation of a datum given by Sketch-SSC. The following bound holds w.p. at least $1 - 2\delta$

$$\|\boldsymbol{g}^*(\boldsymbol{x}) - \hat{\boldsymbol{g}}(\boldsymbol{x})\|_2 \leq \lambda \left(1 + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\, \sqrt{\rho - r}\, \sigma_{r+1}^2\right) + \sqrt{\frac{n}{1-\varepsilon}}$$

with λ as in (12), and σ_{r+1} denotes the $(r+1)$st singular value of $\mathbf{X}$.

This corollary is a direct consequence of the fact that for any $n \times 1$ vector $\boldsymbol{x}$, it holds that $\|\boldsymbol{x}\|_1 \leq \sqrt{n}\|\boldsymbol{x}\|_2$. Accordingly, the following corollary for Sketch-LRR holds because for any rank n matrix $\mathbf{X}$ we have $\|\mathbf{X}\|_* \leq \sqrt{n}\|\mathbf{X}\|_F$.

Corollary 2: Consider the setting of Theorem 1, and let $\boldsymbol{g}^*(\mathbf{X}) := \mathbf{X}\mathbf{Z}$ and $\hat{\boldsymbol{g}}(\mathbf{X}) := \mathbf{X}\mathbf{R}\hat{\mathbf{A}}$ be the representations of all the data given by LRR and Sketch-LRR respectively. The following bound holds w.p. at least $1 - 2\delta$

$$\|\boldsymbol{g}^*(\mathbf{X}) - \hat{\boldsymbol{g}}(\mathbf{X})\|_F \leq \lambda \left(\sqrt{N} + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\, \sqrt{\rho - r}\, \sigma_{r+1}^2\right) + \sqrt{\frac{n}{1-\varepsilon}}$$

with λ as in (12), and σ_{r+1} denotes the $(r+1)$st singular value of $\mathbf{X}$.

For the Sketch-SSC and Sketch-LRR, tighter bounds could possibly be derived by taking into account the special structures of the ℓ_1 and nuclear norms, instead of invoking norm inequalities.

For a dataset $\mathbf{X}$ drawn from a union of subspaces model, batch methods such as SSC, LSR and LRR, should produce a matrix of representations $\mathbf{Z}$ that is block-diagonal, under certain conditions on the separability of subspaces [25], [26]. This, in turn, implies that for data $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{S}_k, \boldsymbol{x}_\ell \in \mathcal{S}_{k'}$ for $k \neq k'$, it holds that

$$\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 \leq \|\boldsymbol{z}_i - \boldsymbol{z}_\ell\|_2 \tag{21}$$

that is the representations of two points in the same subspace, are closer than the representations of two points that lie in different subspaces. The following proposition suggests that this property is approximately inherited by the Sketch-SC algorithms of Sec. III, with high probability.

Proposition 3: Consider $\boldsymbol{x}_i = \mathbf{X}\boldsymbol{z}_i$ and $\boldsymbol{x}_j = \mathbf{X}\boldsymbol{z}_j$, and their representation provided by SSC, LRR or LSR $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, respectively. Let $\rho = \text{rank}(\mathbf{X})$ and $\boldsymbol{a}_i$, $\boldsymbol{a}_j$ be the representation obtained by the corresponding Sketch algorithm of Section III; that is, $\boldsymbol{x}_i = \mathbf{X}\mathbf{R}\boldsymbol{a}_i$, where the $N \times n$ matrix $\mathbf{R}$ is a JLT(ε, δ, D). If $n = \mathcal{O}(\rho \frac{\log(\rho/\varepsilon)}{\varepsilon^2} f(\delta))$, then w.p. at least $1 - \delta$ it holds that

$$\frac{1}{\sqrt{1+\varepsilon}}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 \leq \|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2 \leq \frac{1}{\sqrt{1-\varepsilon}}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2.$$

Proposition 3 also justifies the use of the k-nearest neighbor graph as a post-processing step in Sec. III-C.

As will be seen in the ensuing section, the proposed approach has comparable performance to other high-accuracy SC approaches while requiring markedly less time.

V. Numerical Tests

The proposed method is validated in this section using real datasets. Sketch-SC methods (termed throught this section as *Sketch-SSC, Sketch-LSR* and *Sketch-LRR*) are compared to SSC, LSR, LRR, the orthogonal matching pursuit method (OMP) for large-scale SC [6], as well as ORGEN [7]. When datasets are large ($N \gg$), the proposed methods are only compared to OMP and ORGEN. The figures of merit evaluated are following.

- Accuracy, i.e., percentage of correctly clustered data:
$$\text{Accuracy} := \frac{\text{number of data correctly clustered}}{N}.$$
- Time (in seconds) required for clustering all data. For Algorithms 1 and 2 this includes the time required to generate the JLT matrices $\mathbf{R}$, the time required for computing the products $\mathbf{B} = \mathbf{X}\mathbf{R}$, and in the case of Algorithm 2 $\check{\mathbf{X}} = \check{\mathbf{R}}\mathbf{X}$, $\check{\mathbf{B}} = \check{\mathbf{X}}\mathbf{R}$, as well as the time required for Algorithm 3.

All experiments were performed on a machine with an Intel Core-i5 4570 CPU with 16 GB of RAM. The software used to conduct all experiments is MATLAB [52]. K-means and ANN were implemented using the VLfeat package [53]. All results represent the averages of 10 independent Monte Carlo runs. The regularization scalar λ [cf. (9)] of SSC and Sketch-SSC is computed as per [24, Prop. 1], and it is controlled by a parameter α. ORGEN has two parameters that need to be specified, namely λ and α. LRR and Sketch-LRR employ the $\ell_{2,1}$ norm for the residual $\mathbf{X} - \mathbf{X}\mathbf{Z}$. For LRR, LSR, Sketch-LRR, Sketch-LSR, OMP and ORGEN the parameters are tuned to optimize empirically the performance of each method considered.

The real datasets tested are Hopkins 155 [54], the Extended Yale Face dataset [55], the COIL-100 database [56], and the MNIST handwritten digits dataset [57].

A. Assessing the Effect of Different JLTs

Before comparing the proposed scheme with state-of-the-art competing alternatives, the effect of different JLT matrices on the SC task was tested on two datasets: the Extended Yale Face dataset and the COIL-100 database. The different $N \times n$ JLT matrices assessed are: matrices with i.i.d. ± 1 entries rescaled by $1/\sqrt{n}$ (denoted as *Rademacher*); matrices with i.i.d. $\mathcal{N}(0, 1)$ entries rescaled by $1/\sqrt{n}$ (denoted as *Normal*); Sparse embedding matrices as described in [3], [41] (denoted as *Sparse*); Fast JLTs using the Hadamard matrix as described in [38] (denoted as *Hadamard FJLT*). Fig. 1 depicts the performance of Algorithm 1 for different choices of JLT for the two aforementioned datasets. All JLT matrices achieve comparable performance for the Yale Face database. However, this is not true for the COIL-100 dataset, where the Rademacher JLT seem to provide the most consistent performance.

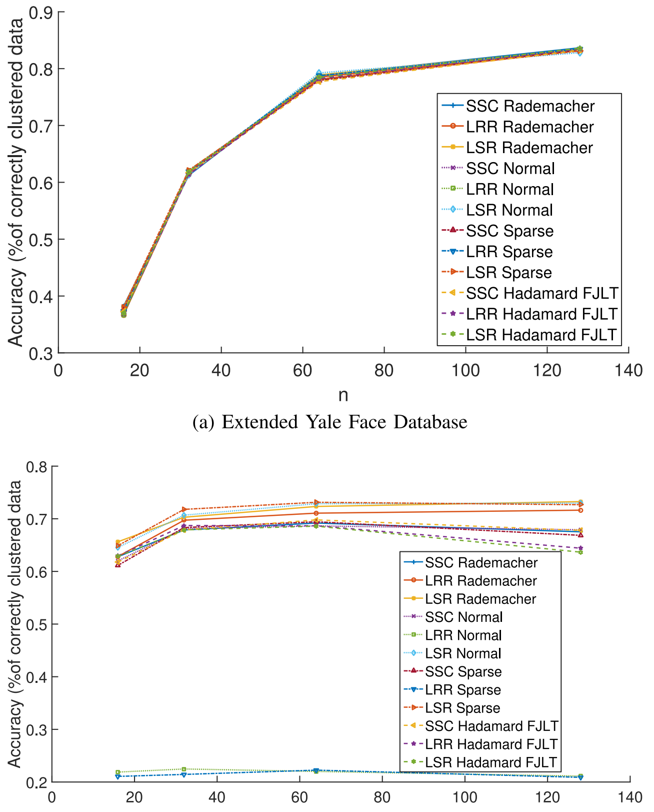
For all tests in the rest of this section Algorithm 1 and 2 use random matrices $\mathbf{R}$, and $\check{\mathbf{R}}$ that are generated having i.i.d. ± 1 entries rescaled by $1/\sqrt{n}$.

B. High Volume of Data

In this section the performance of Sketch-SC (Algorithm 1) is assessed on all datasets. Hopkins 155 is a popular benchmark dataset for subspace clustering and motion segmentation. It contains 155 video sequences, with N points tracked

TABLE I
RESULTS FOR $K = 2$ AND $K = 3$ MOTIONS FOR THE HOPKINS155 DATASET

$K = 2$						
Algorithm	SSC	LRR	LSR	Sketch-SSC	Sketch-LRR	Sketch-LSR
Accuracy	0.9839	0.9723	0.982	**0.946**	**0.9435**	**0.9319**
Time (s)	0.6902	0.9478	0.093	**0.0795**	**0.0808**	**0.0787**
$K = 3$						
Algorithm	SSC	LRR	LSR	Sketch-SSC	Sketch-LRR	Sketch-LSR
Accuracy	0.9747	0.9253	0.9654	**0.8942**	**0.9415**	**0.9242**
Time (s)	1.566	1.295	0.1797	**0.1755**	**0.1459**	**0.1829**



(a) Extended Yale Face Database

(b) COIL-100

Fig. 1. Simulated tests on real datasets Extended Yale Face Database and COIL-100, evaluating the clustering performance with different JLT matrix $\mathbf{R}$.

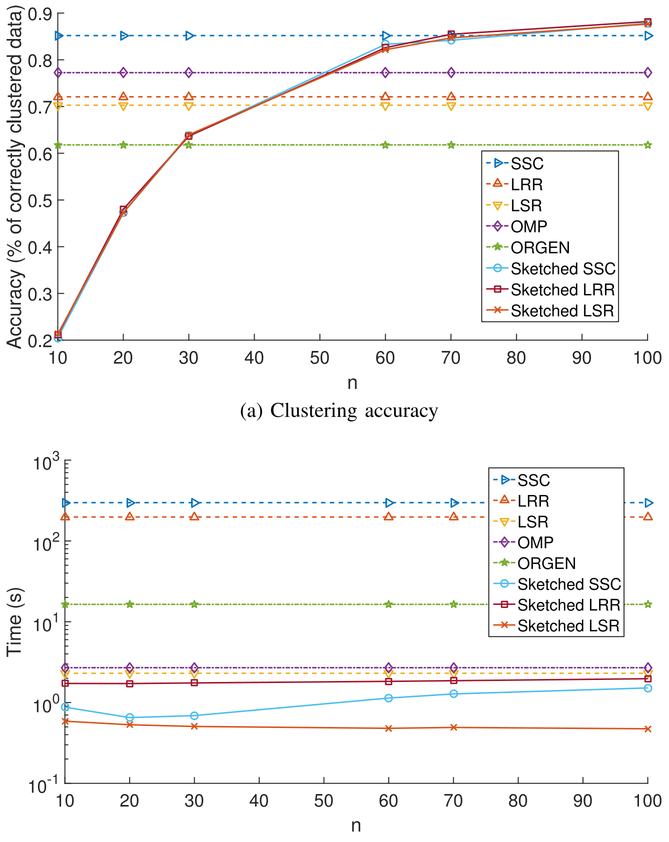
in each frame of a video sequence. Clusters ($K = 2$ or $K = 3$) represent different objects moving in the video sequence. The results for the Hopkins 155 dataset are listed in Table I for $K = 2$ and $K = 3$ clusters, with $n = 0.15N$ for the proposed methods. Here $\alpha = 800$ was used for SSC and $\alpha = 100$ for Sketch-SSC, $\lambda = 1$ for LRR and $\lambda = 10$ for Sketch-LRR, $\lambda = 4.6 \cdot 10^{-3}$ for LSR and Sketch-LSR. The number of nearest neighbors for Algorithm 3 is set to $k = 5$. As the size of the dataset is small, large computational gains are not expected by using Algorithm 1. Nevertheless, the Sketch-SC methods achieve comparable accuracy to their batch counterparts, while in most cases (except one) requiring less time.

The Extended Yale Face database contains $N = 2,414$ face images of $K = 38$ people, each of dimension $D = 2,016$. Fig. 2 shows the results for this dataset for varying n, where $\alpha = 30$ for SSC and $\alpha = 50$ for Sketch-SSC, $\lambda = 0.15$ for LRR and Sketch-LRR, $\lambda = 10^6$ for LSR and Sketch-LSR, the number of non-zeros per column of $\mathbf{Z}$ for OMP is set to 5, while $\lambda = 0.7$ and $\alpha = 200$ for ORGEN. The number of nearest neighbors for Algorithm 3 is set to $k = 5$. The proposed algorithms exhibit comparable accuracy to their batch counterparts, in particular SSC, and also achieve higher accuracy than the state-of-the-art large-scale algorithms OMP and ORGEN, as n increases. Interestingly, with $n \approx 0.03 \cdot N$ the proposed methods achieve the accuracy of batch SSC. In addition, the proposed approach requires markedly less time than the batch methods, and less time than OMP and ORGEN as well.

The Columbia object-image dataset (COIL-100) contains $N = 7,200$ images of size 32×32 corresponding to $K = 100$ objects. Each cluster corresponds to one object, and contains images of it from 72 different angles. Fig. 3 shows the comparisons on this dataset for varying n, where $\alpha = 25$ for SSC and $\alpha = 500$ for Sketch-SSC, $\lambda = 0.9$ for LRR and $\lambda = 10^{-4}$ for Sketch-LRR, $\lambda = 10^2$ for LSR and Sketch-LSR, the number of non-zeros per column of $\mathbf{Z}$ for OMP is set to 2, while $\lambda = 0.95$ and $\alpha = 3$ for ORGEN. The number of nearest neighbors for Algorithm 3 is set to $k = 5$. The proposed approaches exhibit performance comparable to the state-of-the-art as n increases, while requiring significantly less time. Note that, OMP requires almost the same time as the proposed approaches, however its clustering performance is significantly lower.

Fig. 4 plots the singular values of the Extended Yale Face Database and the COIL-100 dataset. For both, the largest singular values are approximately the first 70 ones. Note that for the Extended Yale face database our proposed approaches attain their best performance for approximately $n = 70$ yielding a compression ratio of $\frac{2414}{70} \approx 34.5$, while for the COIL-100 database our proposed approaches reach their peak performance again for $n = 70$, but this time the compression ratio is $\frac{7200}{70} \approx 102.85$. This suggests that, indeed, datasets that exhibit low rank can be compressed with a lower n.

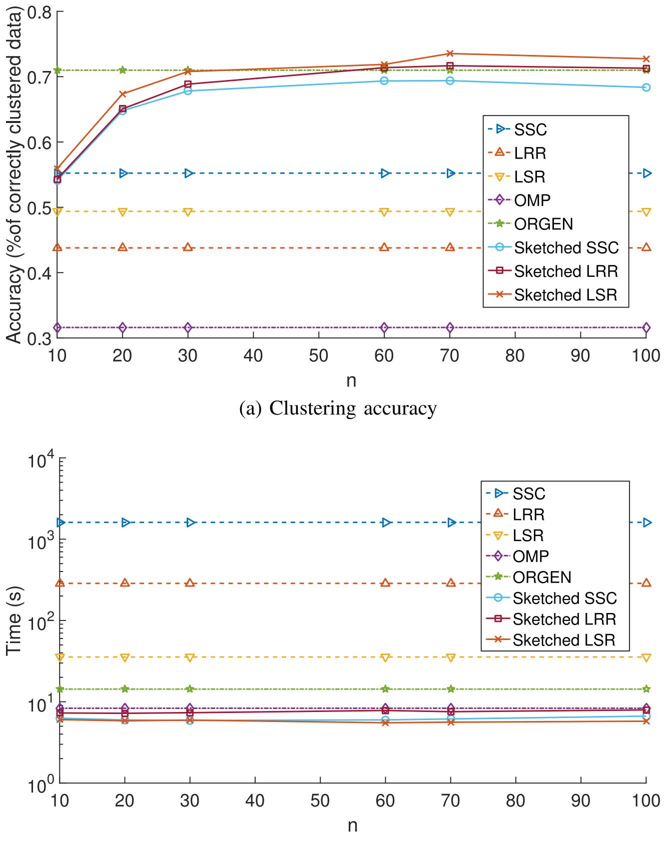
Due to their large size, tests on the following three datasets compare Algorithm 1 only to OMP and ORGEN. The results for the following three datasets are listed in Table II. The MNIST dataset contains 70,000 images of handwritten digits, each of dimension 28×28, with $K = 10$ clusters, one per digit. Here the dataset is preprocessed with a scattering convolutional network [58] and PCA to bring each image dimension down to $D = 500$, as per [6], [7]. Here $n = 200$, $\alpha = 12,000$ for Sketch-SSC, $\lambda = 1$ for Sketch-LRR, $\lambda = 10^{-1}$ for Sketch-LSR, the number of non-zeros per column of $\mathbf{Z}$ for OMP is set to 10, while $\lambda = 0.95$ and $\alpha = 120$ for ORGEN. The number of nearest neighbors for Algorithm 3 is set to $k = 3$, and the set of nearest neighbors for each datum is found using the ANN implementation of the VLfeat package. In this scenario ORGEN showcases the best clustering performance, however Sketch-



(a) Clustering accuracy

(b) Clustering time

Fig. 2. Simulated tests on real dataset Extended Yale Face Database B, with $N = 2,414$ data dimension $D = 2,016$ and $K = 38$ clusters for varying n.



(a) Clustering accuracy

(b) Clustering time

Fig. 3. Simulated tests on real dataset COIL-100, with $N = 7,200$ data dimension $D = 1,025$ and $K = 100$ clusters for varying n.

LRR and Sketch-LSR exhibit comparable accuracy, while requiring markedly less time.

The CoverType dataset consists of $N = 581,012$ data belonging to $K = 7$ clusters. Each cluster corresponds to a different forest cover type. Data are vectors of dimension $D = 54$ that contain cartographic variables, such as soil type, elevation, hillshade etc. Here $n = 150$, $\alpha = 1$ for Sketch-SSC, $\lambda = 10^{-8}$ for Sketch-LRR, $\lambda = 10^{4}$ for Sketch-LSR, the number of non-zeros per column of $\mathbf{Z}$ for OMP is set to 15, while $\lambda = 0.95$ and $\alpha = 500$ for ORGEN. The number of nearest neighbors for Algorithm 3 is set to $k = 10$, and the set of nearest neighbors for each datum is found using the ANN implementation of the VLfeat package.

The PokerHand database contains $N = 10^{6}$ data, belonging to $K = 10$ classes. Each datum is a 5-card hand drawn from a deck of 52 cards, with each card being described by its suit (spades, hearts, diamonds, and clubs) and rank (Ace, 2, 3, ..., Queen, King). Each class represents a valid Poker hand. Here $n = 30$, $\alpha = 10$ for Sketch-SSC, $\lambda = 1$ for Sketch-LRR, $\lambda = 10^{2}$ for Sketch-LSR, the number of non-zeros per column of $\mathbf{Z}$ for OMP is set to 10. The number of nearest neighbors for Algorithm 3 is set to $k = 20$, and the set of nearest neighbors for each datum is found using the ANN implementation of the VLfeat package. Results are not reported for ORGEN as the algorithm did not converge within 24 hours. For both the CoverType and PokerHand datasets, most algorithms exhibit comparable accuracy, while Algorithm 1 requires again less time than OMP or ORGEN.

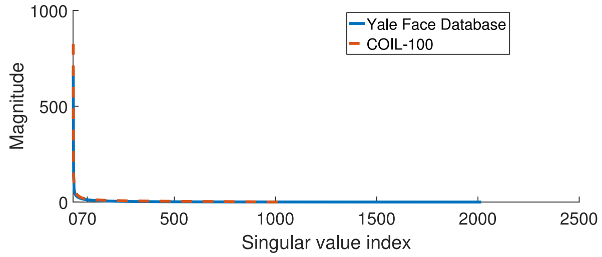


Fig. 4. Singular value plots for the Extended Yale Face database and the COIL-100 dataset.

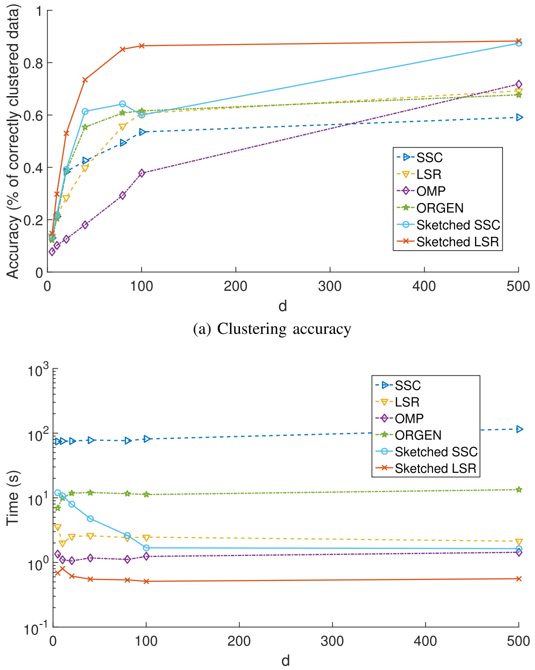
C. High-Dimensional Data

In this section, the performance of Sketch-SC approaches combined with randomized dimensionality reduction (Algorithm 2) is assessed, for the Extended Yale Face database.

Fig. 5 depicts the simulation results on the Extended Yale Face database, when performing dimensionality reduction, for varying d. Here Algorithm 2, with fixed $n = 70$ is compared to its batch counterparts, OMP and ORGEN. LRR and Sketch-LRR are not included in this simulation as the algorithm failed for small values of d. All parameters are the same as the corresponding experiment in Sec. V-B. In this experiment, Sketch-LSR and Sketch-SSC outperform their competing alternatives in terms of clustering accuracy, while maintaining a low computational overhead. OMP also exhibits low computational time, at the expense of clustering accuracy.

TABLE II
RESULTS FOR THE PREPROCESSED MNIST DATASET ($N = 70,000$), THE COVERTYPE DATASET ($N = 581,012$) AND THE POKERHAND DATASET ($N = 1,000,000$)

Dataset		OMP	ORGEN	Sketch-SSC	Sketch-LRR	Sketch-LSR
MNIST	Accuracy	0.47049	0.93788	**0.85825**	**0.90644**	**0.90784**
	Time (s)	502.91	801.3954	**155.1017**	**156.7709**	**99.4724**
CoverType	Accuracy	0.4870	0.4873	**0.42387**	**0.3277**	**0.4860**
	Time (s)	$1.8947 * 10^4$	$2.9893 * 10^4$	**6064.8403**	**4468.5274**	**392.916**
PokerHand	Accuracy	0.5009	-	**0.5008**	**0.1608**	**0.44225**
	Time (s)	$4.6654 * 10^4$		$\mathbf{7.8 * 10^3}$	$\mathbf{3.6 * 10^4}$	$\mathbf{2.71 * 10^3}$



(a) Clustering accuracy

(b) Clustering time

Fig. 5. Simulated tests on real dataset Extended Yale Face Database B, with $N = 2,414$ data dimension $D = 2,016$ and $K = 38$ clusters for varying d and fixed $n = 70$.

VI. CONCLUSIONS AND FUTURE WORK

The present paper introduced a novel data-reduction scheme for subspace clustering, namely Sketch-SC, that enables grouping of data drawn from a union of subspaces based on a random sketching approach for fast, yet-accurate subspace clustering. Performance of the proposed scheme was evaluated both analytically and through simulated tests on multiple real datasets. Future research directions will focus on the development of online and distributed Sketch-SC, able to handle not only big, but also fast-streaming data. In addition, the sketched SC approach could be generalized to subspace clustering for tensor data.

APPENDIX A
TECHNICAL PROOFS

A. *Supporting Lemmata*

The following lemmata will be used to assist in the proofs of the propositions and theorems.

Lemma 2: [59, Corollary 11] Consider an $N \times k$ orthonormal matrix $\mathbf{V}$ with $N \geq k$, and a JLT(ε, δ, k) matrix $\mathbf{R}$ of size $N \times n$. If $n = \mathcal{O}(k\frac{\log(k/\varepsilon)}{\varepsilon^2} f(\delta))$, then the following holds w.p. at least $1 - \delta$

$$1 - \varepsilon \leq \sigma_i^2(\mathbf{V}^\top \mathbf{R}) \leq 1 + \varepsilon \quad \text{for } i = 1, \ldots, k \tag{22}$$

where $\sigma_i(\mathbf{V}^\top \mathbf{R})$ denotes the i-th singular value of $\mathbf{V}^\top \mathbf{R}$.

Lemma 3: [4, Lemma 8] Let $\varepsilon > 0$, and consider the $n \times k$ orthonormal matrix $\mathbf{V}$ with $n > k$, as well as the $n \times r$ matrix $\mathbf{R}$, with $r > k$ satisfying $1 - \varepsilon \leq \sigma_i^2(\mathbf{V}^\top \mathbf{R}) \leq 1 + \varepsilon$ for $i = 1, \ldots, k$. It then holds deterministically that

$$\|(\mathbf{V}^\top \mathbf{R})^\dagger - (\mathbf{V}^\top \mathbf{R})^\top\|_2 \leq \frac{\varepsilon}{\sqrt{1 - \varepsilon}}. \tag{23}$$

B. *Main Proofs*

Proposition 1: Let $\mathbf{X}$ be a $D \times N$ matrix such that $\text{rank}(\mathbf{X}) = \rho$, and define the $D \times n$ matrix $\mathbf{B} := \mathbf{X}\mathbf{R}$, where $\mathbf{R}$ is a JLT(ε, δ, D) of size $N \times n$. If $n = \mathcal{O}(\rho\frac{\log(\rho/\varepsilon)}{\varepsilon^2} f(\delta))$ then w.p. at least $1 - \delta$, it holds that

$$\text{range}(\mathbf{X}) = \text{range}(\mathbf{B}).$$

Proof: Let $\mathbf{X} = \mathbf{U}_\rho \mathbf{\Sigma}_\rho \mathbf{V}_\rho^\top$ be the SVD of $\mathbf{X}$. Since $\mathbf{V}_\rho$ is invertible and $\mathbf{\Sigma}_\rho$ is diagonal, it holds that

$$\text{range}(\mathbf{X}) = \text{range}(\mathbf{U}_\rho) \tag{24}$$

i.e., the columns of $\mathbf{X}$ can be written as linear combinations of the columns of $\mathbf{U}_\rho$ and vice versa. Now consider $\mathbf{B} = \mathbf{X}\mathbf{R} = \mathbf{U}_\rho \mathbf{\Sigma}_\rho \mathbf{V}_\rho^\top \mathbf{R} = \mathbf{U}_\rho \mathbf{\Sigma}_\rho \tilde{\mathbf{V}}_\rho^\top$, where $\tilde{\mathbf{V}}_\rho := \mathbf{R}^\top \mathbf{V}_\rho$, which implies $\text{range}(\mathbf{B}) \subseteq \text{range}(\mathbf{U}_\rho)$. By Lemma 2 $\tilde{\mathbf{V}}_\rho^\top := \mathbf{V}_\rho^\top \mathbf{R}$ is full row rank w.p. at least $1 - \delta$ and thus

$$\mathbf{B}(\tilde{\mathbf{V}}^\top)^\dagger = \mathbf{U}_\rho \mathbf{\Sigma}_\rho \tag{25}$$

which implies that $\text{range}(\mathbf{U}_\rho) = \text{range}(\mathbf{B}) = \text{range}(\mathbf{X})$, where the last equality is due to (24). ■

Proposition 2: Let $\mathbf{X}$ be a $D \times N$ matrix such that $\text{rank}(\mathbf{X}) = \rho$, and define the $D \times n$ matrix $\mathbf{B} := \mathbf{X}\mathbf{R}$, where $\mathbf{R}$ is a JLT(ε, δ, D) of size $N \times n$. If $n = \mathcal{O}(r\frac{\log(r/\varepsilon)}{\varepsilon^2} f(\delta))$, then w.p. at least $1 - 2\delta$ it holds that

$$\|\mathbf{B}(\mathbf{V}_r^\top \mathbf{R})^\dagger - \mathbf{U}_r \mathbf{\Sigma}_r\|_F \leq \left(\varepsilon\frac{\sqrt{1+\varepsilon}}{\sqrt{1-\varepsilon}} + 1 + \varepsilon\right) \|\bar{\mathbf{X}}_r\|_F.$$

Proof: From the first part of the proof of Prop. 1 we have that $\text{range}(\mathbf{B}) \subseteq \text{range}(\mathbf{U}_\rho)$. Now consider

$$\mathbf{B} = \mathbf{X}\mathbf{R} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top \mathbf{R} + \bar{\mathbf{U}}_r \bar{\mathbf{\Sigma}}_r \bar{\mathbf{V}}_r^\top \mathbf{R} \tag{26}$$

By Lemma 2 $\mathbf{V}_r^\top \mathbf{R}$ is full row rank w.p. at least $1-\delta$; thus, right multiplying (26) with $(\mathbf{V}_r^\top \mathbf{R})^\dagger$ yields $\mathbf{B}(\mathbf{V}_r^\top \mathbf{R})^\dagger = \mathbf{U}_r \boldsymbol{\Sigma}_r + \bar{\mathbf{U}}_r \bar{\boldsymbol{\Sigma}}_r \bar{\mathbf{V}}_r^\top \mathbf{R}(\mathbf{V}_r^\top \mathbf{R})^\dagger$, or

$$\mathbf{B}(\mathbf{V}_r^\top \mathbf{R})^\dagger - \mathbf{U}_r \boldsymbol{\Sigma}_r = \bar{\mathbf{U}}_r \bar{\boldsymbol{\Sigma}}_r \bar{\mathbf{V}}_r^\top \mathbf{R}(\mathbf{V}_r^\top \mathbf{R})^\dagger$$

which upon substituting $\bar{\mathbf{X}}_r$ boils down to

$$\begin{aligned} &\mathbf{B}(\mathbf{V}_r^\top \mathbf{R})^\dagger - \mathbf{U}_r \boldsymbol{\Sigma}_r \\ &= \bar{\mathbf{X}}_r \mathbf{R}(\mathbf{V}_r^\top \mathbf{R})^\dagger - \bar{\mathbf{X}}_r \mathbf{R}(\mathbf{V}_r^\top \mathbf{R})^\top + \bar{\mathbf{X}}_r \mathbf{R}(\mathbf{V}_r^\top \mathbf{R})^{\top\cdot} \end{aligned} \tag{27}$$

Using the triangle inequality, and the spectral submultiplicativity of the Frobenius norm, yields

$$\begin{aligned} &\|\mathbf{B}(\mathbf{V}_r^\top \mathbf{R})^\dagger - \mathbf{U}_r \boldsymbol{\Sigma}_r\|_F \\ &= \|\bar{\mathbf{X}}_r \mathbf{R}\left((\mathbf{V}_r^\top \mathbf{R})^\dagger - (\mathbf{V}_r^\top \mathbf{R})^\top\right)\|_F + \|\bar{\mathbf{X}}_r \mathbf{R}(\mathbf{V}_r^\top \mathbf{R})^\top\|_F \\ &\leq \|\bar{\mathbf{X}}_r \mathbf{R}\|_F \;\|(\mathbf{V}_r^\top \mathbf{R})^\dagger - (\mathbf{V}_r^\top \mathbf{R})^\top\|_2 \\ &\quad + \|\bar{\mathbf{X}}_r \mathbf{R}\|_F \;\|(\mathbf{V}_r^\top \mathbf{R})^\top\|_2. \end{aligned} \tag{28}$$

We have from Definition 1 $\|\bar{\mathbf{X}}_r \mathbf{R}\|_F \leq \sqrt{1+\varepsilon}\|\bar{\mathbf{X}}_r\|_F$ w.p. at least $1-\delta$, while Lemma 2 ensures $\|(\mathbf{V}_r^\top \mathbf{R})^\top\|_2 \leq \sqrt{1+\varepsilon}$ w.p. at least $1-\delta$. Since Lemma 3 also implies that $\|(\mathbf{V}_r^\top \mathbf{R})^\dagger - (\mathbf{V}_r^\top \mathbf{R})^\top\|_2 \leq \frac{\varepsilon}{\sqrt{1-\varepsilon}}$ we arrive at [cf. 28]

$$\|\mathbf{B}(\mathbf{V}_r^\top \mathbf{R})^\dagger - \mathbf{U}_r \boldsymbol{\Sigma}_r\|_F \leq \left(\varepsilon\frac{\sqrt{1+\varepsilon}}{\sqrt{1-\varepsilon}} + 1 + \varepsilon\right)\|\bar{\mathbf{X}}_r\|_F. \tag{29}$$

■

Theorem 1: Consider noise-free and normalized data vectors obeying (3) with $\boldsymbol{v}_i \equiv 0$, to form columns of a $D \times N$ data matrix $\mathbf{X}$, with unit ℓ_2 norm per column, and rank($\mathbf{X}$) $= \rho$. Let also $\mathbf{R}$ denote JLT(ε, δ, D) of size $N \times n$. Let $\boldsymbol{g}^*(\boldsymbol{x}) := \mathbf{X}\boldsymbol{a}^* = \boldsymbol{x}$ denote the ground-truth representation of $\boldsymbol{x}$, and $\hat{\boldsymbol{g}}(\boldsymbol{x}) := \mathbf{X}\mathbf{R}\hat{\boldsymbol{a}}$ the representation given by Sketch-LSR. If $n = \mathcal{O}(r\frac{\log(r/\varepsilon)}{\varepsilon^2}f(\delta))$, then the following bound holds w.p. at least $1-2\delta$

$$\begin{aligned} \|\boldsymbol{g}^*(\boldsymbol{x}) - \hat{\boldsymbol{g}}(\boldsymbol{x})\|_2 \leq \lambda\, &\left(1 + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\,\sqrt{\rho - r}\,\sigma_{r+1}^2\right) \\ &+ \frac{1}{\sqrt{1-\varepsilon}} \end{aligned}$$

with λ as in (12), and σ_{r+1} denotes the $(r+1)$st singular value of $\mathbf{X}$.

Proof: The proof will follow the steps in [60]. Consider the Sketch-LSR objective for $\boldsymbol{x}$, namely

$$\frac{\lambda}{2}\|\boldsymbol{x} - \mathbf{X}\mathbf{R}\boldsymbol{a}\|_2^2 + \|\boldsymbol{a}\|_2^2 \tag{30}$$

and the SVD $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. As $\mathbf{U}$ is unitary, minimizing (30) is equivalent to minimizing

$$\frac{\lambda}{2}\|\mathbf{U}^\top \boldsymbol{x} - \boldsymbol{\Sigma}\tilde{\mathbf{V}}^\top \boldsymbol{a}\|_2^2 + \|\boldsymbol{a}\|_2^2 \tag{31}$$

where $\tilde{\mathbf{V}}^\top := \mathbf{V}^\top \mathbf{R}$. Now, decompose the dataset as

$$\mathbf{X} = \mathbf{X}_r + \bar{\mathbf{X}}_r \tag{32}$$

where $\mathbf{X}_r := \mathbf{U}_r \boldsymbol{\Sigma}_r \mathbf{V}_r^\top$ and $\bar{\mathbf{X}}_r := \bar{\mathbf{U}}_r \bar{\boldsymbol{\Sigma}}_r \bar{\mathbf{V}}_r^\top$. Using (32), we can rewrite (31) as

$$\frac{\lambda}{2}\underbrace{\|\boldsymbol{\chi}_r - \boldsymbol{\Sigma}_r \tilde{\mathbf{V}}_r^\top \boldsymbol{a}\|_2^2}_{:=T_1^2} + \frac{\lambda}{2}\underbrace{\|\bar{\boldsymbol{\chi}}_r - \bar{\boldsymbol{\Sigma}}_r \bar{\tilde{\mathbf{V}}}_r^\top \boldsymbol{a}\|_2^2}_{:=T_2^2} + \underbrace{\|\boldsymbol{a}\|_2^2}_{:=T_3^2} \tag{33}$$

where $\boldsymbol{\chi}_r := \mathbf{U}_r^\top \boldsymbol{x}$, and $\bar{\boldsymbol{\chi}}_r := \bar{\mathbf{U}}_r^\top \boldsymbol{x}$. Selecting $\boldsymbol{a}$ as

$$\boldsymbol{a} = \tilde{\mathbf{V}}_r(\tilde{\mathbf{V}}_r^\top \tilde{\mathbf{V}}_r)^{-1}\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\chi}_r$$

T_1^2 vanishes, and T_2 reduces to

$$T_2 = \|\bar{\boldsymbol{\chi}}_r - \bar{\boldsymbol{\Sigma}}_r \bar{\tilde{\mathbf{V}}}_r^\top \tilde{\mathbf{V}}_r(\tilde{\mathbf{V}}_r^\top \tilde{\mathbf{V}}_r)^{-1}\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\chi}_r\|_2. \tag{34}$$

The triangle inequality and the submultiplicativity of the ℓ_2 norm, allows us to bound T_2 as

$$\begin{aligned} T_2 &\leq \|\bar{\boldsymbol{\chi}}_r\|_2 \\ &\quad + \|\bar{\boldsymbol{\Sigma}}_r \bar{\tilde{\mathbf{V}}}_r^\top\|_2 \;\|\tilde{\mathbf{V}}_r(\tilde{\mathbf{V}}_r^\top \tilde{\mathbf{V}}_r)^{-1}\|_2\|\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\chi}_r\|_2. \end{aligned} \tag{35}$$

Now note that $\|\bar{\boldsymbol{\Sigma}}_r \bar{\tilde{\mathbf{V}}}_r^\top\|_2 \leq \|\bar{\boldsymbol{\Sigma}}_r \bar{\tilde{\mathbf{V}}}_r^\top\|_F = \|\bar{\mathbf{U}}_r \bar{\boldsymbol{\Sigma}}_r \bar{\tilde{\mathbf{V}}}_r^\top\|_F = \|\bar{\mathbf{X}}_r \mathbf{R}\|_F$ and recall from Definition 1 that $\|\bar{\mathbf{X}}_r \mathbf{R}\|_F \leq \sqrt{1+\varepsilon}$ $\|\bar{\mathbf{X}}_r\|_F \leq \sqrt{1+\varepsilon}\sqrt{\rho - r}\|\bar{\mathbf{X}}_r\|_2 \leq \sqrt{1+\varepsilon}\sqrt{\rho - r}\sigma_{r+1}^2$ w.p. at least $1-\delta$. By Lemma 2 $\tilde{\mathbf{V}}_r^\top = \mathbf{V}_r^\top \mathbf{R}$ is full row rank w.p. at least $1-\delta$; thus, $\tilde{\mathbf{V}}_r(\tilde{\mathbf{V}}_r^\top \tilde{\mathbf{V}}_r)^{-1} = \tilde{\mathbf{V}}_r^\dagger$, and $\|\tilde{\mathbf{V}}_r^\dagger\|_2 \leq \frac{1}{\sqrt{1-\varepsilon}}$. Furthermore, $\|\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\chi}_r\|_2 = \|\mathbf{V}_r \boldsymbol{\Sigma}_r^{-1}\mathbf{U}_r^\top \boldsymbol{x}\|_2 \leq 1$, and $\|\bar{\boldsymbol{\chi}}_r\|_2 = \|\bar{\mathbf{U}}_r^\top \boldsymbol{x}\|_2 = 1$. Similarly, T_3 in (33) can be bounded w.p. at least $1-\delta$ due to Lemma 2 as

$$\begin{aligned} 2T_3 &= \|\tilde{\mathbf{V}}_r(\tilde{\mathbf{V}}_r^\top \tilde{\mathbf{V}}_r)^{-1}\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\chi}_r\|_2 = \|\tilde{\mathbf{V}}_r^\dagger \boldsymbol{\Sigma}_r^{-1}\boldsymbol{\chi}_r\|_2 \\ &\leq \|\tilde{\mathbf{V}}_r^\dagger\|_2 \;\|\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\chi}_r\|_2 \leq \frac{1}{\sqrt{1-\varepsilon}} \end{aligned} \tag{36}$$

Finally, since the chosen $\boldsymbol{a}$ in (33) satisfies (35) and (36), so will do any minimizer $\hat{\boldsymbol{a}}$ of (30). ■

Corollary 1: Consider the setting of Theorem 1, and let $\hat{\boldsymbol{g}}(\boldsymbol{x}) := \mathbf{X}\mathbf{R}\hat{\boldsymbol{a}}$ be the representation of a datum given by Sketch-SSC. The following bound holds w.p. at least $1-2\delta$

$$\begin{aligned} \|\boldsymbol{g}^*(\boldsymbol{x}) - \hat{\boldsymbol{g}}(\boldsymbol{x})\|_2 \leq \lambda\, &\left(1 + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\,\sqrt{\rho - r}\,\sigma_{r+1}^2\right) \\ &+ \sqrt{\frac{n}{1-\varepsilon}} \end{aligned}$$

with λ as in (12), and σ_{r+1} denotes the $(r+1)$st singular value of $\mathbf{X}$.

Proof: Consider the Sketch-SSC objective for $\boldsymbol{x}$, namely

$$\frac{\lambda}{2}\underbrace{\|\boldsymbol{x} - \mathbf{X}\mathbf{R}\boldsymbol{a}\|_2^2}_{:=T_1^2} + \underbrace{\|\boldsymbol{a}\|_1}_{:=T_2}. \tag{37}$$

From Theorem 1 we have $T_1 \leq \lambda\,(1 + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\,\sqrt{\rho - r}\,\sigma_{r+1}^2)$, and $\|\boldsymbol{a}\|_2 \leq \frac{1}{\sqrt{1-\varepsilon}}$. Since for any $n \times 1$ vector $\boldsymbol{z}$ it holds that $\|\boldsymbol{z}\|_1 \leq \sqrt{n}\|\boldsymbol{z}\|_2$, we have $T_2 \leq \sqrt{n}\|\boldsymbol{a}\|_2 \leq \sqrt{\frac{n}{1-\varepsilon}}$ yielding the claim of the corollary. ■

Corollary 2: Consider the setting of Theorem 1, and let $\boldsymbol{g}^*(\mathbf{X}) := \mathbf{X}\mathbf{Z}$ and $\hat{\boldsymbol{g}}(\mathbf{X}) := \mathbf{X}\mathbf{R}\hat{\mathbf{A}}$ be the representations of all the data given by LRR and Sketch-LRR respectively. The following bound holds w.p. at least $1 - 2\delta$

$$\|\boldsymbol{g}^*(\mathbf{X}) - \hat{\boldsymbol{g}}(\mathbf{X})\|_F \le \lambda \left(\sqrt{N} + \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}\,\sqrt{\rho - r}\,\sigma_{r+1}^2\right) + \sqrt{\frac{n}{1-\varepsilon}}$$

with λ as in (12), and σ_{r+1} denoting the $(r+1)$st singular value of $\mathbf{X}$.

Proof: Consider the Sketch-LRR objective for $\mathbf{X}$, namely

$$\frac{\lambda}{2}\underbrace{\|\mathbf{X} - \mathbf{XRA}\|_F^2}_{:=T_1^2} + \underbrace{\|\mathbf{A}\|_*}_{:=T_2}. \tag{38}$$

As with Corollary 1, T_1 can be bounded using the results of Theorem 1, and $\|\mathbf{A}\|_F \le \frac{1}{\sqrt{1-\varepsilon}}$. Since for any rank n matrix $\mathbf{Z}$ it holds that $\|\mathbf{Z}\|_* \le \sqrt{n}\|\mathbf{Z}\|_F$ we have $T_2 \le \sqrt{n}\|\mathbf{A}\|_F \le \sqrt{\frac{n}{1-\varepsilon}}$, yielding the claim of the corollary. ■

Proposition 3: Consider $\boldsymbol{x}_i = \mathbf{X}\boldsymbol{z}_i$ and $\boldsymbol{x}_j = \mathbf{X}\boldsymbol{z}_j$, and their representation provided by SSC, LRR or LSR $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, respectively. Let $\rho = \operatorname{rank}(\mathbf{X})$ and $\boldsymbol{a}_i$, $\boldsymbol{a}_j$ be the representation obtained by the corresponding Sketch algorithm of Section III; that is, $\boldsymbol{x}_i = \mathbf{XR}\boldsymbol{a}_i$, where the $N \times n$ matrix $\mathbf{R}$ is a JLT(ε, δ, D). If $n = \mathcal{O}(\rho\frac{\log(\rho/\varepsilon)}{\varepsilon^2}f(\delta))$, then w.p. at least $1-\delta$ it holds that

$$\frac{1}{\sqrt{1+\varepsilon}}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 \le \|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2 \le \frac{1}{\sqrt{1-\varepsilon}}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2.$$

Proof: By definition, we have $\boldsymbol{x}_i = \mathbf{X}\boldsymbol{z}_i = \mathbf{XR}\boldsymbol{a}_i$, and thus

$$\mathbf{X}(\boldsymbol{z}_i - \boldsymbol{z}_j) = \mathbf{XR}(\boldsymbol{a}_i - \boldsymbol{a}_j) = \boldsymbol{x}_i - \boldsymbol{x}_j. \tag{39}$$

Let $\mathbf{X} = \mathbf{U}_\rho\mathbf{\Sigma}_\rho\mathbf{V}_\rho^\top$, and rewrite (39) as

$$\mathbf{U}_\rho\mathbf{\Sigma}_\rho\mathbf{V}_\rho^\top(\boldsymbol{z}_i - \boldsymbol{z}_j) = \mathbf{U}_\rho\mathbf{\Sigma}_\rho\mathbf{V}_\rho^\top\mathbf{R}(\boldsymbol{a}_i - \boldsymbol{a}_j). \tag{40}$$

Left-multiplying by $\mathbf{\Sigma}_\rho^{-1}\mathbf{U}_\rho^\top$ reduces (40) to

$$\mathbf{V}_\rho^\top(\boldsymbol{z}_i - \boldsymbol{z}_j) = \mathbf{V}_\rho^\top\mathbf{R}(\boldsymbol{a}_i - \boldsymbol{a}_j). \tag{41}$$

Taking the norm of both sides, and noting that $\mathbf{V}$ is an orthonormal matrix implies that

$$\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 = \|\mathbf{R}(\boldsymbol{a}_i - \boldsymbol{a}_j)\|_2 \tag{42}$$

which upon recalling Definition 1 yields

$$\begin{aligned}&\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2 \le \sqrt{1+\varepsilon}\|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2,\\ &\sqrt{1-\varepsilon}\|\boldsymbol{a}_i - \boldsymbol{a}_j\|_2 \le \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2\end{aligned} \tag{43}$$

w.p. at least $1-\delta$. ■

Appendix B
Algorithm Details

A. ADMM Algorithm for (15)

Consider the Sketch-SSC for a single datum $\boldsymbol{x}$

$$\min_{\boldsymbol{a}} \; \frac{\lambda}{2}\|\boldsymbol{x} - \mathbf{B}\boldsymbol{a}\|_2^2 + \|\boldsymbol{a}\|_1 \tag{44}$$

The optimization problem of (44) will be solved using the alternating direction method of multipliers [42]. Define a new $n \times 1$ vector of auxiliary variables $\boldsymbol{c}$, and consider the following optimization problem that is equivalent to (44)

$$\begin{aligned}\min_{\boldsymbol{a},\boldsymbol{c}} \quad & \frac{\lambda}{2}\|\boldsymbol{x} - \mathbf{B}\boldsymbol{a}\|_2^2 + \|\boldsymbol{c}\|_1\\ \text{s.to.} \quad & \boldsymbol{a} = \boldsymbol{c}.\end{aligned} \tag{45}$$

The augmented Lagrangian of (45) is

$$\mathcal{L} = \frac{\lambda}{2}\|\boldsymbol{x} - \mathbf{B}\boldsymbol{a}\|_2^2 + \|\boldsymbol{c}\|_1 + \frac{\nu}{2}\|\boldsymbol{a} - \boldsymbol{c} + \boldsymbol{\delta}\|_2^2 \tag{46}$$

where $\boldsymbol{\delta}$ is a $n \times 1$ vector of dual variables and $\nu > 0$ is a penalty parameter. At each ADMM iteration the variables $\boldsymbol{a}, \boldsymbol{c}$ are updated by setting the gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{a}$ and $\boldsymbol{c}$ respectively to $\mathbf{0}$. Furthermore, the dual variables $\boldsymbol{\delta}$ are updated using a gradient ascent step at each iteration. The update of $\boldsymbol{a}$ at the i-th iteration is given by

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{a}} = -\lambda\mathbf{B}^\top(\boldsymbol{x} - \mathbf{B}\boldsymbol{a}) + \nu(\boldsymbol{a} - \boldsymbol{c} + \boldsymbol{\delta}) = \mathbf{0} \Rightarrow$$

$$\boldsymbol{a}[i+1] = (\lambda\mathbf{B}^\top\mathbf{B} + \nu\mathbf{I})^{-1}(\lambda\mathbf{B}^\top\boldsymbol{x} + \nu(\boldsymbol{c}[i] - \boldsymbol{\delta}[i])) \tag{47}$$

where brackets indicate ADMM iteration indices. Accordingly, the update for $\boldsymbol{c}$ is given by

$$\boldsymbol{c}[i+1] = \mathcal{T}_{1/\nu}(\boldsymbol{a}[i+1] + \boldsymbol{\delta}[i]) \tag{48}$$

where $\mathcal{T}_\sigma(\cdot)$ denotes the element-wise soft-thresholding operator

$$\mathcal{T}_\sigma(z) := \begin{cases} z - \sigma & \text{if } z > \sigma\\ 0 & \text{if } |z| \le \sigma\\ z + \sigma & \text{if } z < -\sigma\end{cases}. \tag{49}$$

Finally, $\boldsymbol{\delta}$ is updated as

$$\boldsymbol{\delta}[i+1] = \boldsymbol{\delta}[i] + \boldsymbol{a}[i+1] - \boldsymbol{c}[i+1]. \tag{50}$$

The entire process is listed in Algorithm 4.

Algorithm 4: ADMM solver of Sketch-SSC [cf. (15)].

```
Input: $D \times N$ data matrix $\mathbf{X}$; $D \times n$ basis $\mathbf{B}$;
    regularization parameter $\lambda$;
Output: Model matrix $\mathbf{A}$;
for Each datum $\boldsymbol{x}_j$ to $\boldsymbol{x}_N$ do
    Initialize $\boldsymbol{a}_j[0], \boldsymbol{c}[0], \boldsymbol{\delta}[0]$
    repeat
        Compute $\boldsymbol{a}_j[i+1]$ using (47)
        Compute $\boldsymbol{c}[i+1]$ using (48)
        Compute $\boldsymbol{\delta}[i+1]$ using (50)
        Update iteration counter $i \leftarrow i+1$
    until convergence
end for
$\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]$.
```

B. ALM Algorithm for (16)

Consider the Sketch-LRR

$$\min_{\mathbf{A}} \; \frac{\lambda}{2}\|\mathbf{X} - \mathbf{BA}\|_F^2 + \|\mathbf{A}\|_* \tag{51}$$

The optimization problem of (51) will be solved using the augmented Lagrangian method (ALM) [25], [61]. Define a new

Algorithm 5: ALM solver of Sketch-LRR [cf. (16)].

```
Input: $D \times N$ data matrix $\mathbf{X}$; $D \times n$ basis $\mathbf{B}$;
    regularization parameter $\lambda$;
Output: Model matrix $\mathbf{A}$;
Initialize $\mathbf{A}, \mathbf{C}, \boldsymbol{\Delta}$
repeat
    Compute $\mathbf{A}[i+1]$ using (54)
    Compute $\mathbf{C}[i+1]$ using (55)
    Compute $\boldsymbol{\delta}[i+1]$ using (56)
    Update $\nu$ using (57)
    Update iteration counter $i \leftarrow i+1$
until convergence
```

$n \times N$ matrix of auxiliary variables $\mathbf{C}$, and consider the following optimization task that is equivalent to (51)

$$\begin{aligned}\min_{\mathbf{A},\mathbf{C}} \quad & \frac{\lambda}{2}\|\mathbf{X} - \mathbf{B}\mathbf{A}\|_F^2 + \|\mathbf{C}\|_* \\ \text{s.to.} \quad & \mathbf{A} = \mathbf{C} \end{aligned} \tag{52}$$

The augmented Lagrangian of (52) is

$$\mathcal{L} = \frac{\lambda}{2}\|\mathbf{X} - \mathbf{B}\mathbf{A}\|_F^2 + \|\mathbf{C}\|_* + \frac{\nu}{2}\|\mathbf{A} - \mathbf{C} + \boldsymbol{\Delta}\|_F^2 \tag{53}$$

where $\boldsymbol{\Delta}$ is a $n \times N$ matrix of dual variables and $\nu > 0$ is a penalty parameter. At each ALM iteration the variables $\mathbf{A}, \mathbf{C}$ are updated by setting the gradient of $\mathcal{L}$ w.r.t. $\mathbf{A}$ and $\mathbf{C}$ respectively to $\mathbf{0}$. Furthermore, the dual variables $\boldsymbol{\Delta}$ are updated using a gradient ascent step per iteration. The update of $\mathbf{A}$ at the i-th iteration is given by

$$\begin{aligned} & \frac{\partial \mathcal{L}}{\partial \mathbf{A}} = \mathbf{0} \Rightarrow \\ & \mathbf{A}[i+1] = (\lambda \mathbf{B}^\top \mathbf{B} + \nu \mathbf{I})^{-1}(\lambda \mathbf{B}^\top \mathbf{X} - \nu(\mathbf{C}[i] - \boldsymbol{\Delta}[i])) \end{aligned} \tag{54}$$

where brackets indicate ALM iteration indices. Accordingly, the update for $\mathbf{C}$ is given by

$$\mathbf{C}[i+1] = \arg\min_{\mathbf{C}} \frac{1}{\nu}\|\mathbf{C}\|_* + \frac{1}{2}\|\mathbf{C} - (\mathbf{A}[i+1] + \boldsymbol{\Delta}[i])\|_F^2. \tag{55}$$

Note that the update (55) can be performed using the Singular Value Thresholding algorithm [62]. Finally $\boldsymbol{\Delta}$ is updated as

$$\boldsymbol{\Delta}[i+1] = \boldsymbol{\Delta}[i] + \mathbf{A}[i+1] - \mathbf{C}[i+1] \tag{56}$$

and the penalty parameter is also updated as

$$\nu = \min(p\nu, \nu_{\max}) \tag{57}$$

where $p > 1$ is a prescribed constant, and $\nu_{\max}$ is a predefined maximum limit for ν.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2001.

[2] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[3] D. P. Woodruff, "Sketching as a tool for numerical linear algebra," *Found. Trends Theor. Comput. Sci.*, vol. 10, no. 1/2, pp. 1–157, 2014.

[4] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized dimensionality reduction for k-means clustering," *IEEE Trans. Inf. Theory*, vol. 61, no. 2, pp. 1045–1062, Feb. 2015.

[5] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemp. Math.*, vol. 26, no. 189–206, 1984.

[6] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, vol. 1, pp. 3918–3927.

[7] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 3928–3937.

[8] P. A. Traganitis and G. B. Giannakis, "A randomized approach to large-scale subspace clustering," in *Proc. 50th Asilomar Conf. Signals, Syst. Comput.*, 2016, pp. 1019–1023.

[9] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.

[10] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 90–105, 2004.

[11] S. Lloyd, "Least-squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[12] P. K. Agarwal and N. H. Mustafa, "K-means projective clustering," in *Proc. 23rd ACM SIGMOD-SIGACT-SIGART Symp.*, Paris, France, Jun. 2004, pp. 155–165.

[13] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, 1999.

[14] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, Sep. 2007.

[15] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.

[16] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *Int. J. Comput. Vis.*, vol. 100, no. 3, pp. 217–240, 2012.

[17] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, pp. 6320–6342, Nov. 2015.

[18] M. Rahmani and G. Atia, "Innovation pursuit: A new approach to subspace clustering," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6276–6291, Sep. 2017.

[19] P. A. Traganitis and G. B. Giannakis, "PARAFAC-based multilinear subspace clustering for tensor data," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Washington, DC, USA, 2016, pp. 1280–1284.

[20] T. Zhang, A. Szlam, and G. Lerman, "Median k-flats for hybrid linear modeling with many outliers," in *Proc. 12th Int. Conf. Comput. Vis. Workshops*, Kyoto, Japan, Sep. 2009, pp. 234–241.

[21] P. A. Traganitis and G. B. Giannakis, "Efficient subspace clustering of large-scale data streams with misses," in *Proc. Annu. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, 2016, pp. 590–595.

[22] J. Shen, P. Li, and H. Xu, "Online low-rank subspace clustering by basis dictionary pursuit," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA, 2016, pp. 622–631.

[23] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[24] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[25] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 663–670.

[26] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan, "Robust and efficient subspace segmentation via least-squares regression," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 347–360.

[27] Y. Panagakis and C. Kotropoulos, "Elastic net subspace clustering applied to pop/rock music structure analysis," *Pattern Recognit. Lett.*, vol. 38, pp. 46–53, 2014.

[28] Y. Fang, R. Wang, B. Dai, and X. Wu, "Graph-based learning via auto-grouped sparse regularization and kernelized extension," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 142–154, 2015.

[29] R. Heckel, M. Tschannen, and H. Bölcskei, "Dimensionality-reduced subspace clustering," *Inf. Inference J. IMA*, vol. 6, no. 3, pp. 246–283, Sep. 2017.

[30] D. Pimentel-Alarcón, L. Balzano, and R. Nowak, "Necessary and sufficient conditions for sketched subspace clustering," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput.*, Champaign, IL, USA, 2016, pp. 1335–1343.

[31] Y. Wang, Y.-X. Wang, and A. Singh, "A theoretical analysis of noisy sparse subspace clustering on dimensionality-reduced data," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 1422–1431.

[32] F. Pourkamali-Anaraki and S. Becker, “Preconditioned data sparsification for big data with applications to PCA and K-means,” *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 2954–2974, May 2017.

[33] P. A. Traganitis, K. Slavakis, and G. B. Giannakis, “Sketch and validate for big data clustering,” *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 678–690, Jun. 2015.

[34] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, “A unified framework for representation-based subspace clustering of out-of-sample and large-scale data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2499–2512, Dec. 2016.

[35] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, “Greedy feature selection for subspace clustering.” *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2487–2517, 2013.

[36] D. Achlioptas, “Database-friendly random projections: Johnson-Lindenstrauss with binary coins,” *J. Comput. Syst. Sci.*, vol. 66, no. 4, pp. 671–687, 2003.

[37] E. Liberty and S. W. Zucker, “The mailman algorithm: A note on matrix-vector multiplication,” *Inf. Process. Lett.*, vol. 109, no. 3, pp. 179–182, 2009.

[38] N. Ailon and B. Chazelle, “The fast Johnson–Lindenstrauss transform and approximate nearest neighbors,” *SIAM J. Comput.*, vol. 39, no. 1, pp. 302–322, 2009.

[39] N. Ailon and E. Liberty, “Fast dimension reduction using Rademacher series on dual BCH codes,” *Discrete Comput. Geom.*, vol. 42, no. 4, 2009, Art. no. 615.

[40] F. Pourkamali-Anaraki and S. Hughes, “Memory and computation efficient pca via very sparse random projections,” in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1341–1349.

[41] K. L. Clarkson and D. P. Woodruff, “Low rank approximation and regression in input sparsity time,” in *Proc. 45th Annu. ACM Symp. Theory Comput.*, 2013, pp. 81–90.

[42] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, “Decentralized learning for wireless communications and networking,” in *Splitting Methods in Communication and Imaging, Science and Engineering*, R. Glowinski, S. Osher, and W. Yin, Eds. New York, NY, USA: Springer, 2016.

[43] P. A. Traganitis and G. B. Giannakis, “Efficient subspace clustering of large-scale data streams with misses,” in *Proc. Annu. Conf. Inf. Sci. Systems*, Princeton, NJ, USA, Mar. 2016, pp. 590–595.

[44] Q. Le, T. Sarlos, and A. Smola, “Fastfood - approximating kernel expansions in loglinear time,” in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 244–252. [Online]. Available: http://jmlr.org/proceedings/papers/v28/le13.html

[45] R. B. Lehoucq, D. C. Sorensen, and C. Yang, “ARPACK users’ guide: solution of large-scale eigenvalue problems with implicitly restarted arnoldi methods,” *Soc. Ind. Appl. Math.*, vol. 6. pp. xv–142, 1998.

[46] V. Kalantzis, R. Li, and Y. Saad, “Spectral Schur complement techniques for symmetric eigenvalue problems,” *Electron. Trans. Numer. Anal.*, vol. 45, pp. 305–329, 2016.

[47] D. C. Anastasiu and G. Karypis, “L2knng: Fast exact k-nearest neighbor graph construction with l2-norm pruning,” in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Melbourne, VIC, Australia, 2015, pp. 791–800.

[48] Y. Park, S. Park, S.-g. Lee, and W. Jung, “Greedy filtering: A scalable algorithm for k-nearest neighbor graph construction,” in *International Conference on Database Systems for Advanced Applications*. Bali, Indonesia: Springer, 2014, pp. 327–341.

[49] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: A Procrustean approach to learning binary codes for large-scale image retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[50] P. Indyk and R. Motwani, “Approximate nearest neighbors: Towards removing the curse of dimensionality,” in *Proc. 30th Annu. ACM Symp. Theory Comput.*, Dallas, TX, USA, 1998, pp. 604–613.

[51] M. Slaney and M. Casey, “Locality-sensitive hashing for finding nearest neighbors [lecture notes],” *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 128–131, Mar. 2008.

[52] *MATLAB Version 8.6.0 (R2015b)*. Natick, MA, USA: MathWorks Inc., 2015.

[53] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms,” 2008. [Online]. Available: http://www.vlfeat.org/.

[54] R. Tron and R. Vidal, “A benchmark for the comparison of 3-D motion segmentation algorithms,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, 2007, pp. 1–8.

[55] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[56] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia object image library (coil-100),” Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-006-96, 1996.

[57] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[58] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.

[59] T. Sarlos, “Improved approximation algorithms for large matrices via random projections,” in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci.*, Berkeley, CA, USA, 2006, pp. 143–152.

[60] Y. Yang, M. Pilanci, and M. J. Wainwright, “Randomized sketches for kernels: Fast and optimal non-parametric regression,” *Ann. Stat.*, vol. 45, no. 3, pp. 991–1023, Jun. 2017.

[61] Z. Lin, M. Chen, and Y. Ma, “The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices,” Technical Report UILU-ENG-09-2215, UIUC, Oct. 2009.

[62] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

Panagiotis A. Traganitis (S’14) received the Diploma in electrical and computer Engineering from the National Technical University of Athens, Athens, Greece, in 2013, and the M.Sc. degree in electrical engineering from the University of Minnesota (UMN), Twin Cities, Minneapolis, MN, USA, in 2015. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, UMN, Twin Cities. His research interests include statistical signal processing, distributed learning, big data analytics, and network science.

Georgios B. Giannakis (F’97) received the Diploma in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1981, and the M.Sc. degree in electrical engineering, the M.Sc. degree in mathematics, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1983, 1986, and 1986, respectively. He was with the University of Virginia from 1987 to 1998, and since 1999, he has been a Professor with the University of Minnesota, Minneapolis, MN, USA, where he holds an Endowed Chair in Wireless Telecommunications, a University of Minnesota McKnight Presidential Chair in Electrical and Computer Engineering, and serves as the Director of the Digital Technology Center.

His general interests include the areas of communications, networking, and statistical signal processing—subjects on which he has published more than 400 journal papers, 700 conference papers, 25 book chapters, two edited books, and two research monographs (h-index 128). His current research interests include learning from Big Data, wireless cognitive radios, and network science with applications to social, brain, and power networks with renewables. He is the coinventor of 30 patents issued, and the co-recipient of nine best journal paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He was also the recipient of the Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2015). He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.