Subspace Clustering via Optimal Direction Search

Mostafa Rahmani, *Student Member, IEEE* and George K. Atia, *Member, IEEE*

Abstract—**This paper presents a new spectral-clustering-based approach to the subspace clustering problem. Underpinning the proposed method is a convex program for optimal direction search, which for each data point d, finds an optimal direction in the span of the data that has minimum projection on the other data points and non-vanishing projection on d. The obtained directions are subsequently leveraged to identify a neighborhood set for each data point. An Alternating Direction Method of Multipliers (ADMM) framework is provided to efficiently solve for the optimal directions. The proposed method is shown to often outperform the existing subspace clustering methods, particularly for unwieldy scenarios involving high levels of noise and close subspaces, and yields the state-of-the-art results for the problem of face clustering using subspace segmentation.**

Index Terms—**Spectral Clustering, Convex Optimization, Unsupervised Learning, Face Clustering, Innovation pursuit**

I. Introduction

In many applications of signal processing and machine learning, the data can be well-approximated with low-dimensional subspaces [1]. Subspace recovery methods have been instrumental in reducing dimensionality and recognizing intrinsic patterns in data. Principal Component Analysis (PCA) is a standard tool which approximates the data with a single low-dimensional subspace that has minimum distance from the data points [2]–[4]. However, in many applications the data admits clustering structures, wherefore a union of subspaces can better model the data [5].

In the subspace clustering problem, the data points lie in a union of an unknown number of unknown linear subspaces whose dimensions are also generally unknown. The role of a subspace segmentation algorithm then is to learn these low-dimensional subspaces and to cluster the data points to their respective subspaces. This data model has been widely applied to many modern signal processing and machine learning applications, including computer vision [5], [6], gene expression analysis [7], [8], and image processing [9].

Many different approaches to subspace clustering were devised in related work, including statistical-based approaches [10]–[13], spectral clustering [14], the algebraic-geometric approach [15], the innovation pursuit approach [16], and iterative methods [17], [18]. We refer the reader to [5], [14], [16] for an overview of the topic. Much of the recent work has focused on spectral-clustering [19] based methods [14], [20]–[27], which all share a common structure. Specifically, a neighborhood set for each data point is first identified to construct a similarity matrix. Subsequently, spectral clustering

[19] is applied to the similarity matrix. Spectral-clustering-based methods differ mostly in the first step.

There exists several recent spectral-clustering-based methods with superior empirical performance. SSC is a popular spectral-clustering-based algorithm, which finds a sparse representation for each data point with respect to the rest of data to construct the similarity matrix [14]. It was shown in [24] that SSC can yield exact clustering even for subspaces with intersections under certain conditions. A different algorithm called Low-Rank Representation (LRR) [23] uses nuclear norm minimization to build the similarity matrix. In [22], the inner product between the data points is used as measure of similarity to find a neighborhood set for each data point.

A. Summary of contributions

This paper presents a new spectral-clustering-based subspace segmentation method dubbed Direction search based Subspace Clustering (DSC). Underlying our approach is a direction search program that associates an optimal direction with each data point. For each data point, the algorithm finds an optimal direction in the column space of the data matrix that has minimum projection on the rest of the data and non-vanishing projection on that data point. An optimization framework is presented to find all the directions by solving one convex program. Subsequently, the similarity matrix is formed using the obtained directions. The presented numerical experiments demonstrate that DSC often outperforms existing spectral-clustering-based methods, and remarkably improves over the state-of-the-art result for the problem of face clustering using subspace segmentation. In addition, an iterative method to efficiently solve the proposed direction search optimization is provided.

B. Notation and data model

Bold-face upper-case letters are used to denote matrices and bold-face lower-case letters are used to denote vectors. For a vector $\mathbf{a}$, $\|\mathbf{a}\|_p$ denotes its ℓ_p-norm. Given two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ with an equal number of rows, the matrix $\mathbf{A} = [\mathbf{A}_1 \ \mathbf{A}_2]$ is the matrix formed from the concatenation of $\mathbf{A}_1$ and $\mathbf{A}_2$. Given a vector $\mathbf{a}$, $|\mathbf{a}|$ is the vector of absolute values of the elements of $\mathbf{a}$. Given a matrix $\mathbf{A}$, $\mathbf{a}_i$ denotes its ith column, $\|\mathbf{A}\|_{1,1} = \sum_i \|\mathbf{a}_i\|_1$, and $\|\mathbf{A}\|_{1,2} = \sum_i \|\mathbf{a}_i\|_2$, $\text{col}(\mathbf{A})$ its column space, and $\text{tr}(\mathbf{A})$ its trace. In addition, $\text{diag}(\mathbf{A})$ returns a vector of the diagonal elements of $\mathbf{A}$. The symbol $\oplus$ denotes the direct sum operator.

In this paper, the data is assumed to follow the subspace clustering structure expressed in the following data model.

Data Model 1. *The data matrix* $\mathbf{D} \in \mathbb{R}^{M_1 \times M_2}$ *can be represented as* $\mathbf{D} = [\mathbf{D}_1 \ ... \ \mathbf{D}_N]\mathbf{T}$, *where* $\mathbf{T}$ *is an*

The authors are with the Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816 USA (e-mails: mostafa@knights.ucf.edu, george.atia@ucf.edu).
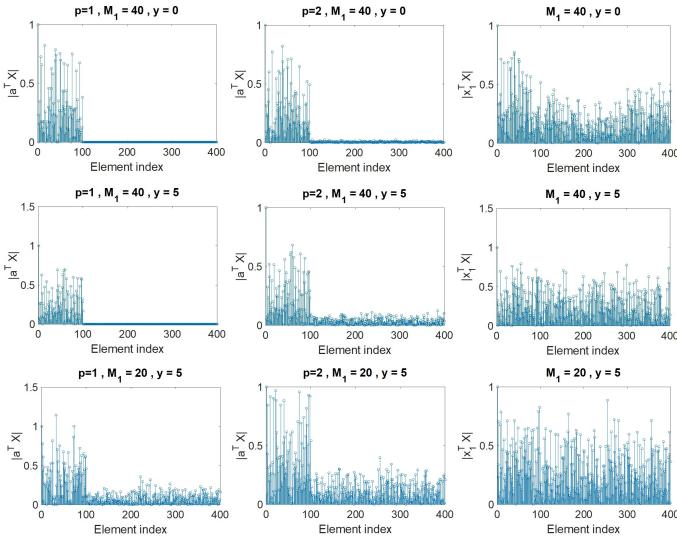
Fig. 1. Measures of similarity $|\mathbf{a}^T\mathbf{X}|$ and $|\mathbf{x}_i^T\mathbf{X}|$ adopted by DSC and TSC to identify a neighborhood set for the first data point. First row $M_1 = 40$ and $y = 0$, second row $M_1 = 40$ and $y = 5$, third row $M_1 = 20$ and $y = 5$.

arbitrary permutation matrix. The columns of $\mathbf{D}_i \in \mathbb{R}^{M_1 \times n_i}$ lie in $\mathcal{S}_i$, where $\mathcal{S}_i$ is an r_i-dimensional linear subspace, for $1 \le i \le N$, and, $\sum_{i=1}^{N} n_i = M_2$.

We define $\mathbf{Q} \in \mathbb{R}^{M_1 \times r}$ as an orthonormal basis for $\mathrm{col}(\mathbf{D})$, where r is the rank of $\mathbf{D}$. If the data is noisy, the matrix $\mathbf{Q}$ is formed using the dominant left singular vectors of $\mathbf{D}$. In addition, $\mathbf{X} \in \mathbb{R}^{r \times M_2}$ is defined as $\mathbf{X} = \mathbf{Q}^T\mathbf{D}$.

II. DIRECTION SEARCH CLUSTERING

The proposed approach consists of M_2 identical optimization problems, one per data point. The optimization problem

$$\min_{\mathbf{a}} \|\mathbf{a}^T\mathbf{X}\|_p \quad \text{s.t.} \quad \mathbf{a}^T\mathbf{x}_i = 1 \tag{1}$$

corresponding to $\mathbf{d}_i$, searches for a direction in the column space of the projected data $\mathbf{X}$ with non-zero projection on $\mathbf{x}_i$ and minimum projection on the rest of the data. In this paper, we use $p = 1$ or $p = 2$ for the ℓ_p-norm. The linear constraint enforces the optimal point of (1) to have strong coherence with $\mathbf{x}_i$. In practice, the data points within a subspace are mutually coherent, wherefore the optimal point of (1) will have large projection on other data points in the subspace containing $\mathbf{x}_i$. Accordingly, if we sample few of the columns of $\mathbf{X}$ corresponding to the elements of $|\mathbf{a}^T\mathbf{X}|$ with largest values, they will all lie in the subspace containing $\mathbf{x}_i$. Thus, we exploit the obtained directions to construct a neighborhood set for each data point in order to construct a similarity matrix, hence the name Direction search Subspace Clustering (DSC). Algorithm 1 describes the proposed DSC method. The first step finds all the directions in one shot via solving a $r \times M_2$ convex optimization problem. The similarity matrix is formed in the second step, then in the final step the spectral clustering algorithm is applied to the similarity matrix. For more information about Steps 2.2 and 3, the reader is referred to [5], [19].

Algorithm 1 Direction search based Subspace Clustering (DSC)

Initialization: Set g equal to the cardinality of a neighborhood set. Set $\mathbf{W} \in \mathbb{R}^{M_2 \times M_2}$ equal to a zero matrix and set p equal to 1 or 2.
Normalize the ℓ_2-norm of the columns of $\mathbf{D}$ (i.e., set $\mathbf{d}_i$ equal to $\mathbf{d}_i/\|\mathbf{d}_i\|_2$). Form matrix $\mathbf{X} = \mathbf{Q}^T\mathbf{D}$.

1. Define $\mathbf{A}^*$ as the optimal points of

$$\min_{\mathbf{A}} \|\mathbf{X}^T\mathbf{A}\|_{1,p} \quad \text{subject to} \quad \mathrm{diag}(\mathbf{A}^T\mathbf{X}) = \mathbf{1}, \tag{2}$$

where $\mathbf{1} \in \mathbb{R}^{M_2 \times 1}$ is the vector of all ones.

2. For $i = 1$ to M_2
2.1 Set $\mathcal{I}$ equal to the index set of the largest g elements of $|\mathbf{a}_i^*\mathbf{X}|$.
2.2 $\mathbf{w}_\mathcal{I}^i = \exp(-2 * \mathrm{acos}(\mathbf{x}_i^T\mathbf{X}_\mathcal{I}))$, where acos and exp are the element-wise inverse cosine and exponential functions, respectively, $\mathbf{X}_\mathcal{I}$ the columns of $\mathbf{X}$ indexed by $\mathcal{I}$, $\mathbf{w}^i$ the ith row of $\mathbf{W}$, and $\mathbf{w}_\mathcal{I}^i$ the elements of $\mathbf{w}^i$ indexed by $\mathcal{I}$.
2. End For

3. Apply spectral clustering to the matrix $\mathbf{W} + \mathbf{W}^T$.

Sparse regularization: If the data matrix is low rank, each vector in $\mathrm{col}(\mathbf{Q})$ can be represented as a sparse combination of the columns of $\mathbf{D}$. For such setting, we can rewrite (2) as

$$\begin{aligned} \min_{\mathbf{A},\mathbf{Z}} \quad & \|\mathbf{A}^T\mathbf{X}\|_{1,p} + \gamma\|\mathbf{Z}\|_1 \\ \text{subject to} \quad & \mathbf{A} = \mathbf{X}\mathbf{Z} \\ & \mathrm{diag}(\mathbf{A}^T\mathbf{X}) = \mathbf{1}, \end{aligned} \tag{3}$$

where $\mathbf{Z} \in \mathbb{R}^{M_2 \times M_2}$. The sparse representation can further enhance the robustness of the proposed approach to noise. The singular vectors corresponding to the noise component do not admit sparse representations in the data – that is, are normally obtained through linear combinations of a large number of data points. Thus, enforcing a sparse representation for the optimal direction averts a solution for

$$\begin{aligned} \min_{\mathbf{a},\mathbf{z}} \quad & \|\mathbf{a}^T\mathbf{X}\|_p + \gamma\|\mathbf{z}\|_1 \\ \text{subject to} \quad & \mathbf{a} = \mathbf{X}\mathbf{z} \,, \quad \mathbf{a}^T\mathbf{x}_i = 1 \,, \end{aligned} \tag{4}$$

that lies in close proximity with the noise singular vectors.

A. Connection and contrast to TSC and iPursuit

We point out some similarities and fundamental differences between DSC and some of the more related approaches. DSC and TSC bear some resemblance from a structural standpoint, yet are conceptually very different concerning how data similarity is viewed and measured, and thus how neighborhoods are constructed. Specifically, underlying DSC is the convex program (2) whereby optimal directions are obtained in Step 1 to be used in Step 2.1 of Algorithm 1 to construct the similarity matrix. This is fundamentally different from the thresholding-based subspace clustering (TSC) algorithm [22], which uses the data points themselves as directions. Thus, in TSC the equivalent of set $\mathcal{I}$ is formed from the indices of the largest elements of $|\mathbf{d}_i^T\mathbf{D}|$. Hence, the performance of TSC greatly declines when the subspaces are in close proximity.

As an example, suppose the columns of $\mathbf{D} = [\mathbf{D}_1\mathbf{D}_2\mathbf{D}_3\mathbf{D}_4]$ lie in the union of 4 10-dimensional subspaces $\{\mathcal{S}_i\}_{i=1}^4$, each with a 100 data points, where $\mathcal{S}_i = \mathcal{M} \oplus \mathcal{R}_i$. $\mathcal{M}$ is a random y-dimensional subspace, and $\{\mathcal{R}_i\}_{i=1}^4$ are random

$(10 - y)$-dimensional subspaces. Thus, the dimension of the intersections between the subspaces is equal to y with high probability. We solve (1) with $i = 1$. The first two columns of Fig. 1 illustrate the values of $|\mathbf{a}^T\mathbf{X}|$ for $p = 1$ and $p = 2$ adopted by DSC as a measure of similarity to build the neighborhood set of the first data point, and the last column displays the values of $|\mathbf{x}_1^T\mathbf{X}|$ adopted by TSC. In the first row $M_1 = 40$ and $y = 0$, corresponding to independent subspaces that do not intersect. For, the second row $y = 5$ (i.e., closer subspaces) and in the last row $M_1 = 20$ and $y = 5$. As desired, the largest values of $|\mathbf{a}^T\mathbf{X}|$ used to form the set $\mathcal{I}$ in Step 2.1 consistently correspond to the first subspace. When $y = 0$, the subspaces are not very close to each other and TSC can build a correct neighborhood for $\mathbf{d}_1$ since the data columns corresponding to the largest elements of $|\mathbf{x}_1^T\mathbf{X}|$ all lie in the same subspace $\mathcal{S}_1$. However, in the second and third row where $y = 5$, TSC cannot form a proper neighborhood as the data points corresponding to the largest elements of $|\mathbf{x}_1^T\mathbf{X}|$ do not lie in the same cluster. Despite the close proximity of the subspaces, (1) finds a direction in the data span that is strongly coherent with the first subspace and has small projection on the other subspaces. This feature notably empowers DSC to distinguish the data clusters.

In [16], we developed an iterative subspace clustering approach termed iPursuit (short for innovation pursuit). Akin to DSC, iPursuit leverages some direction search module for subspace identification, albeit the approach is very different. To describe the connection to, and difference from, DSC we need the following definition.

Definition 1. *(Innovation subspace) Suppose $\mathbf{D}$ follows Data Model 1, $\mathbf{V}_i$ is an orthonormal basis for $\mathcal{S}_i$, $\mathbf{C}_i$ is an orthonormal basis for $\overset{N}{\underset{\substack{k=1 \\ k \neq i}}{\oplus}} \mathcal{S}_k$, and $\mathcal{S}_i \nsubseteq \overset{N}{\underset{\substack{k=1 \\ k \neq i}}{\oplus}} \mathcal{S}_k$ (i.e., $\mathcal{S}_i$ does not lie completely in the direct sum of the other subspaces). Then, the innovation subspace corresponding to $\mathcal{S}_i$, denoted $\mathrm{innov}(\mathcal{S}_i)$, is defined as the linear subspace in $\mathrm{col}(\mathbf{D})$ spanned by the columns of $(\mathbf{I} - \mathbf{C}_i\mathbf{C}_i^T)\mathbf{V}_i$.*

In [16], it was shown that if $\mathcal{S}_1 \nsubseteq \oplus_{i=2}^N \mathcal{S}_i$, and $\mathbf{q} \in \mathbb{R}^{M_1}$ is sufficiently close to $\mathrm{innov}(\mathcal{S}_1)$, then the optimal point of

$$\min_{\mathbf{a}} \|\mathbf{a}^T\mathbf{X}\|_1 \quad \text{s.t.} \quad \mathbf{a}^T\mathbf{q} = 1 \,, \tag{5}$$

lies in $\mathrm{innov}(\mathcal{S}_1)$. Therefore, iPursuit exploits this result, combined with the fact that $\mathrm{innov}(\mathcal{S}_i)$ is orthogonal to $\oplus_{i=2}^N \mathcal{S}_i$ per Definition 1, to directly separate out the different subspaces successively. In contrast, DSC is a spectral-clustering-based approach which uses the outcome of direction search to build a similarity matrix. The main restriction of iPursuit is that it requires every subspace to carry innovation relative to the other subspaces. In other words, iPursuit *requires that no subspace lies in the direct sum of the other subspaces*. DSC does not have such restrictions. For illustration, the first row of Fig. 1 indeed shows the orthogonality of the optimal direction to $\oplus_{i=2}^4 \mathcal{S}_i$ when $M_1 = 40$. However, when $M_1 = 20$ in the last row of Fig. 1, the requirement of iPursuit is violated and iPursuit cannot yield correct clustering. On the other hand, DSC samples few columns corresponding to the largest
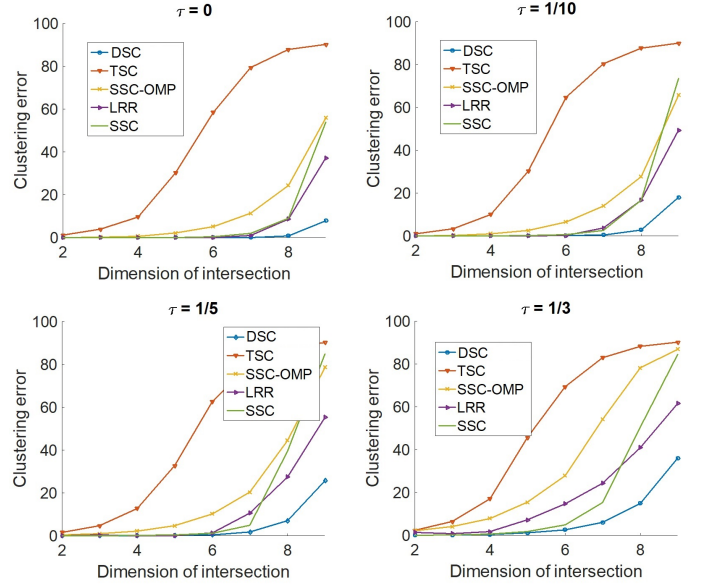

Fig. 2. Performance of the algorithms versus the dimension of intersection for different noise levels.

elements of $|\mathbf{a}^T\mathbf{X}|$, which all lie in the first cluster. Therefore, DSC can form a proper neighborhood set even if the subspaces do not have relative innovations.

B. Solving the proposed optimization problem

In this section, we use an Alternating Direction Method of Multipliers (ADMM) [28] to develop an efficient algorithm for solving (3) which is a generalized form of (2). The optimization problem (3) is equivalent to

$$\begin{aligned}
\min_{\mathbf{A},\mathbf{Z}} \quad & \|\mathbf{T}\|_{1,p} + \gamma\|\mathbf{Z}\|_1 \\
\text{subject to} \quad & \mathbf{A} = \mathbf{X}\,\mathbf{U}, \ \mathbf{U} = \mathbf{Z} \\
& \mathbf{T} = \mathbf{X}^T\mathbf{A}, \quad \text{and} \ \ \mathrm{diag}\left(\mathbf{A}^T\mathbf{X}\right) = \mathbf{1} \,.
\end{aligned} \tag{6}$$

The Lagrangian function of (6) can be written as

$$\begin{aligned}
\mathcal{L}(\mathbf{T}, \mathbf{A}, \mathbf{U}, \mathbf{Z}) &= \|\mathbf{T}\|_{1,p} + \gamma\|\mathbf{Z}\|_1 + \frac{\mu}{2}\|\mathbf{A} - \mathbf{X}\mathbf{U}\|_F^2 + \\
& \frac{\mu}{2}\left(\mathrm{diag}\left(\mathbf{A}^T\mathbf{X}\right) - \mathbf{1}\right)_2^2 + \frac{\mu}{2}\|\mathbf{T} - \mathbf{X}^T\mathbf{A}\|_F^2 + \frac{\mu}{2}\|\mathbf{Z} - \mathbf{U}\|_F^2 \\
& + \mathrm{tr}\left(\mathbf{Y}_1^T(\mathbf{A} - \mathbf{X}\mathbf{U})\right) + \mathrm{tr}\left(\mathbf{y}_2^T\left(\mathrm{diag}\left(\mathbf{A}^T\mathbf{X}\right) - \mathbf{1}\right)\right) \\
& + \mathrm{tr}\left(\mathbf{Y}_3^T(\mathbf{T} - \mathbf{X}^T\mathbf{A})\right) + \mathrm{tr}\left(\mathbf{Y}_4^T(\mathbf{Z} - \mathbf{U})\right) \,,
\end{aligned} \tag{7}$$

where μ is the regularization parameter. The ADMM approach is an iterative procedure. Define $(\mathbf{A}_k, \mathbf{U}_k, \mathbf{Z}_k, \mathbf{T}_k)$ as the optimization variables and $(\mathbf{Y}_1^k, \mathbf{y}_2^k, \mathbf{Y}_3^k, \mathbf{Y}_4^k)$ as the Lagrange multipliers at the kth iteration. Define $\mathbf{G}_1 = \mu^{-1}(\mathbf{I} + 2\mathbf{X}\mathbf{X}^T)^{-1}$, $\mathbf{G}_2 = \mu^{-1}(\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}$, and define the element-wise function $\mathcal{T}_\epsilon(c)$ as $\mathcal{T}_\epsilon(c) = \mathrm{sgn}(c)\max(|c| - \epsilon, 0)$. Define a column-wise operator $\mathbf{H} = \mathcal{C}_\epsilon(\mathbf{C})$ as follows: set $\mathbf{h}_i$ equal to zero if $\|\mathbf{c}_i\|_2 \leq \epsilon$, otherwise set $\mathbf{h}_i = \mathbf{c}_i - \epsilon\,\mathbf{c}_i/\|\mathbf{c}_i\|_2$. Each

iteration consists of the following steps:

$$
\begin{aligned}
\mathbf{A}_{k+1} = \mathbf{G}_1 \big(& \mu \mathbf{X} \mathbf{U}_k + \mu \mathbf{X} + \mu \mathbf{X} \mathbf{T}_k - \mathbf{Y}_1^k \\
& - \mathbf{X} \left(\mathrm{diag}(\mathbf{y}_2^k) \right) + \mathbf{X} \mathbf{Y}_3^k \big) \\
\text{if } p = 1 : \mathbf{T}_{k+1} & = \mathcal{T}_{\mu^{-1}}(\mathbf{X}^T \mathbf{A}_{k+1} - \mu^{-1} \mathbf{Y}_3^k) \\
\text{if } p = 2 : \mathbf{T}_{k+1} & = \mathcal{C}_{\mu^{-1}}(\mathbf{X}^T \mathbf{A}_{k+1} - \mu^{-1} \mathbf{Y}_3^k) \\
\mathbf{Z}_{k+1} & = \mathcal{T}_{\mu^{-1}\gamma}(\mathbf{U}_k - \mu^{-1} \mathbf{Y}_4^k) \\
\mathbf{U}_{k+1} & = \mathbf{G}_2 \left(\mu \mathbf{X}^T \mathbf{A}_{k+1} + \mu \mathbf{Z}_{k+1} + \mathbf{X}^T \mathbf{Y}_1^k + \mathbf{Y}_4^k \right) \\
\mathbf{Y}1^{k+1} & = \mathbf{Y}_1^k + \mu(\mathbf{A}_{k+1} - \mathbf{X} \mathbf{U}_{k+1}) \\
\mathbf{y}_2^{k+1} & = \mathbf{y}_2^k + \mu \left(\mathrm{diag} \left(\mathbf{A}_{k+1}^T \mathbf{X} \right) - \mathbf{1} \right) \\
\mathbf{Y}_3^{k+1} & = \mathbf{Y}_3^k + \mu(\mathbf{T}_{k+1} - \mathbf{X}^T \mathbf{A}_{k+1}) \\
\mathbf{Y}_4^{k+1} & = \mathbf{Y}_4^k + \mu(\mathbf{Z}_{k+1} - \mathbf{U}_{k+1}) .
\end{aligned}
\tag{8}
$$

These steps are repeated until the algorithm converges or the number of iterations exceeds a predefined threshold.

The complexity of the initialization step (obtaining the matrices $\mathbf{G}_1$ and $\mathbf{G}_2$) is roughly $\mathcal{O}(M_2^2 r)$. The order of complexity of each iteration is also $\mathcal{O}(r M_2^2)$. Thus, the overall complexity is $\mathcal{O}(T M_2^2 r + M_2 M_1 r)$, where T is the number of iterations and the second term corresponds to the complexity of calculating the matrix $\mathbf{X}$.

III. NUMERICAL SIMULATIONS

In this section, we study the performance of DSC with both synthetic and real data. In the experiments with synthetic data, the data lies in a union of subspaces $\{\mathcal{S}_i\}_{i=1}^N$ where $\mathcal{S}_i = \mathcal{M} \oplus \mathcal{R}_i$. The subspace $\mathcal{M}$ is a random y-dimensional subspace and $\{\mathcal{S}_i\}_{i=1}^N$ are random d-dimensional subspaces. Hence, the dimension of the intersection between the subspaces is equal to y. The data points are distributed uniformly at random within the subspaces, i.e., a data point lying in an r_i-dimensional subspace $\mathcal{S}_i$ is generated as $\mathbf{V}_i \mathbf{g}$, where the elements of $\mathbf{g} \in \mathbb{R}^{r_i}$ are sampled independently from a standard normal distribution $\mathcal{N}(0,1)$ and $\mathbf{V}_i$ is an orthonormal basis for $\mathcal{S}_i$. If $\mathbf{M}_2'$ is the number of misclassified data points, the clustering error is defined as $100 \times \frac{M_2'}{M_2}$. DSC is compared against SSC [14], LRR [23], TSC [22], SSC-OMP [20], and SCC [26]. In the simulations with synthetic data, the performance of DSC with $p = 1$ and $p = 2$ are similar. However, in the face clustering example, DSC yields better performance with $p = 2$. Thus, we report all the results with $p = 2$. In all experiments, $\mu = 3.3$ and $\gamma = 0.01$.

A. Noisy data

In this section, we study the performance of DSC with noisy data. The data points lie in a union of 20 10-dimensional linear subspaces and $M_1 = 40$. There are 100 data points in each cluster. The noisy data matrix $\mathbf{D}_e \in \mathbb{R}^{40 \times 2000}$ is obtained as $\mathbf{D}_e = \mathbf{D} + \alpha \mathbf{E}$, where $\mathbf{D}$ follows Data Model 1, the elements of $\mathbf{E}$ are sampled from $\mathcal{N}(0,1)$, and α determines the relative power of the noise component. Fig. 2 shows the performance of the different algorithms versus the dimension of intersection for $\tau := \frac{\|\alpha \mathbf{E}\|_F}{\|\mathbf{D}\|_F}$ equal to 0, 1/10, 1/5 and 1/3. It is worth noting that in this experiment not all subspaces have relative innovations, which excludes iPursuit as a feasible choice. As
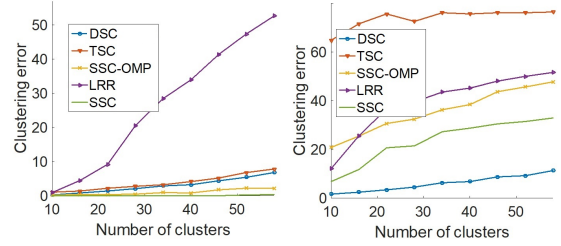

Fig. 3. Performance with different number of data clusters. **Left:** the dimension of intersection $y = 0$, **Right:** $y = 4$.

TABLE I
CLUSTERING ERROR (%) OF DIFFERENT ALGORITHMS ON THE EXTENDED YALE B DATASET.

# of subjects	DSC	SSC	SSC-P	SCC	TSC
5	**2.56**	4.24	29.04	62.62	25.62
10	**4.88**	9.53	32.76	74.13	40.46
15	**4.71**	15.66	34.21	77.02	44.79
20	**6.45**	19.95	33.67	78.50	45.30
25	**8.53**	24.76	50.19	79.37	46.46
38	**8.84**	27.47	50.37	88.86	47.12

shown, the proposed approach notably outperforms the other spectral-clustering-based algorithms in all four scenarios.

B. Clustering error versus N

Here, we investigate the performance of the algorithms when there is a large number of clusters. The data follows Data Model 1, the dimension of each subspace is equal to 6, and $M_1 = 20$. There are 60 data points in each cluster. Fig. 3 shows the clustering error versus the number of subspaces. In the left plot $y = 0$ and all algorithms expect for LRR yield accurate clustering. In the right plot $y = 4$, in which case the clustering error of all algorithms except for DSC notably increases with the number of subspaces.

C. Face clustering

Face clustering is a challenging and practical application of subspace clustering [14]. We use the Extended Yale B dataset, which contains 64 images for each of 38 individuals in frontal view and different illumination conditions [29]. The faces corresponding to each subject can be approximated with a low-dimensional subspace. Thus, a data set containing face images from multiple subjects can be modeled as a union of subspaces.

We apply DSC to face clustering and present results for a different number of clusters in Table I. The performance is also compared with SSC, SCC, and TSC. Heretofore, SSC yielded the best known result for this problem. For each number of clusters shown (except 38), we ran the algorithms over 50 different random combinations of subjects from the 38 clusters. To expedite the runtime, we project the data on the span of the first 500 left singular vectors, which does not affect the performance of the algorithms (expect SSC). For SSC, we report the results without projection (SSC) and with projection (SSC-P). As shown, DSC yields accurate clustering and notably outperforms the performance achieved by SSC.

REFERENCES

[1] C. M. Bishop, *Pattern recognition and machine learning.* springer, 2006.

[2] M. Rahmani and G. Atia, "Randomized robust subspace recovery and outlier detection for high dimensional data matrices," *IEEE Transactions on Signal Processing*, vol. 65, no. 6, March 2017.

[3] T. Zhang and G. Lerman, "A novel m-estimator for robust PCA," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 749–808, 2014.

[4] G. Lerman, M. B. McCoy, J. A. Tropp, and T. Zhang, "Robust computation of linear models by convex relaxation," *Foundations of Computational Mathematics*, vol. 15, no. 2, pp. 363–410, 2015.

[5] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 2, no. 28, pp. 52–68, 2011.

[6] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. 1–11.

[7] B. McWilliams and G. Montana, "Subspace clustering of high-dimensional data: a predictive approach," *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 736–772, 2014.

[8] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, p. 1, 2009.

[9] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[10] A. Y. Yang, S. R. Rao, and Y. Ma, "Robust statistical estimation and segmentation of multiple subspaces," in *Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2006, pp. 99–99.

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

[12] Y. Sugaya and K. Kanatani, "Geometric structure of degeneracy for multi-body motion segmentation," in *Statistical Methods in Video Processing*. Springer, 2004, pp. 13–25.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] 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.

[15] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.

[16] M. Rahmani and G. K. Atia, "Innovation pursuit: A new approach to subspace clustering," *IEEE Transactions on Signal Processing*, 2017.

[17] P. S. Bradley and O. L. Mangasarian, "k-plane clustering," *Journal of Global Optimization*, vol. 16, no. 1, pp. 23–32, 2000.

[18] T. Zhang, A. Szlam, and G. Lerman, "Median k-flats for hybrid linear modeling with many outliers," in *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, 2009, pp. 234–241.

[19] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[20] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2487–2517, 2013.

[21] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4238–4246.

[22] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *arXiv preprint arXiv:1307.4891*, 2013.

[23] 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.

[24] M. Soltanolkotabi, E. J. Candes *et al.*, "A geometric analysis of subspace clustering with outliers," *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.

[25] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When lrr meets ssc," in *Advances in Neural Information Processing Systems*, 2013, pp. 64–72.

[26] G. Chen and G. Lerman, "Spectral curvature clustering (scc)," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 317–330, 2009.

[27] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," in *Advances in Neural Inf. Processing Systems*, 2014, pp. 2753–2761.

[28] S. 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.

[29] 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.