Robust Subspace Clustering by Bi-sparsity Pursuit: Guarantees and Sequential Algorithm

Ashkan Panahi[1*] Xiao Bian[2] Hamid Krim[3] Liyi Dai[4]
[1,2,3] ECE Department, North Carolina State Univerity, Raleigh, NC
[4] U.S. Army Research Office, Research Triangle Park, NC
apanahi[1],xbian[2],ahk[3] @ncsu.edu [4]liyi.dai.civ@mail.mil

Abstract

We consider subspace clustering under sparse noise, for which a non-convex optimization framework based on sparse data representations has been recently developed. This setup is suitable for a large variety of applications with high dimensional data, such as image processing, which is naturally decomposed into a sparse unstructured foreground and a background residing in a union of low-dimensional subspaces. In this framework, we further discuss both performance and implementation of the key optimization problem. We provide an analysis of this optimization problem demonstrating that our approach is capable of recovering linear subspaces as a local optimal solution for sufficiently large data sets and sparse noise vectors. We also propose a sequential algorithmic solution, which is particularly useful for extremely large data sets and online vision applications such as video processing.

1. Introduction

Linear models underlie many successful approaches in machine learning, especially for large-scale problems. Principal component analysis (PCA) is perhaps the most well-known example, which identifies a generic low-dimensional subspace representing a given data set [1]. In recent years the premise of PCA has been extended to the case where the data set can be divided into individual groups with arbitrarily different underlying subspaces [2]. This notion is widely referred to as the union of subspaces (UoS) model and has been found significantly useful in many machine learning and signal processing applications. A central problem under the UoS model, is to uncover the data partitions and detect the underlying subspaces for a given data set. This problem is known as subspace clustering and has received considerable attention in the recent unsupervised machine learning literature [3, 4]. As such, it is a natural choice for learning multi-modal behavior of data, dimension reduction and a suitable alternative for conventional vector clustering methods such as k-means in unsupervised learning tasks [5, 6, 7, 8].

Subspace clustering is an NP-hard problem and there are several approaches to approximately solving it. One of the most popular methods in the recent literature is the Sparse Subspace Clustering (SSC) technique [9], which is based on the following optimization

$$\begin{aligned} &\min_{\mathbf{W}\in\mathbb{R}^{n\times n}} \|\mathbf{W}\|_{1,1} \\ &\text{subject to } \mathbf{L}\mathbf{W}=\mathbf{L}, \quad \forall i; (\mathbf{W})_{ii}=0, \end{aligned} \qquad (1)$$

where $\mathbf{L}$ is the input data set and $(.)_{ij}$ denotes the entry in the i^{th} row and j^{th} column. Each column of the matrix $\mathbf{W}$ provides a sparse representation of the data point in the corresponding column of $\mathbf{L}$ in terms of the other ones. The main rationale for SSC is that the data points from each subspace are more likely to be represented by other data points of the same subspace. Hence, the support of $\mathbf{W}$ in (1) reflects the desired partitions. The sparse subspace clustering method in (1) has a number of advantages over other existing methods. First, it is based on a convex optimization, which provides superior convergence properties as well as provable performance guarantees. Second, the number of variables in (1) is independent of the data dimension (number of rows of $\mathbf{X}$) which makes this approach suitable for applications with high-dimensional data. In fact, a closer look at (1) reveals that SSC relies on only the inner product of the data points, which also makes this method a good candidate for kernelization [10].

A natural extension of subspace clustering is to consider the UoS model under uncertainty. This problem is generally referred to as robust subspace clustering. Various approaches to robust subspace clustering exist in the literature [11, 12]. In [13, 14], Robust Subspace Recovery (RoSuRe) by bi-sparsity pursuit is introduced as a generalization of

[*] This work is supported in part by the U.S. Army Research Office under agreements W911NF-04-D-0003 and W911NF-16-2-0005.

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00147

SSC, and is based on the following optimization:

$$\min_{\mathbf{W}\in\mathbb{R}^{n\times n},\mathbf{E}\in\mathbb{R}^{m\times n}} \|\mathbf{W}\|_{1,1}+\lambda\|\mathbf{E}\|_{1,1}$$
$$\text{subject to}$$
$$(\mathbf{X}-\mathbf{E})\mathbf{W}=(\mathbf{X}-\mathbf{E}),\quad \forall i;(\mathbf{W})_{ii}=0 \qquad (2)$$

where $\mathbf{E}$ denotes a sparse error matrix and $\mathbf{X}=\mathbf{L}+\mathbf{E}$ is the noisy data. The RoSuRe approach is motivated by applications such as image processing where the data points are decomposed into a small unstructured part (foreground) modeled by $\mathbf{E}$, and a large remainder (background) residing in a union of low-dimensional subspaces. Although the optimization in (2) is non-convex, [14] verifies by numerical experiments that the non-linear optimization techniques such as linearized Alternating Direction Method of Multipliers (linearized ADMM) with well-tuned parameters, reliably provide the desired solution.

In this paper, our goals are twofold: 1) to explain the persistent behavior of the RoSuRe in large-scale problems, 2) to provide its sequential implementation, which is desirable for online machine learning applications such as video processing and surveillance. Our algorithm is inspired by the incremental proximal optimization methods [15], which are tightly related to the Stochastic Gradient Descent (SGD) technique [16, 17], and addresses the associated constraints by a quadratic penalty function. We show by numerical experiments on both synthetic and real-world data that our algorithm can run in real time, and is easily capable of decomposing videos into foreground and background. Related to the analysis, due to non-convexity of RoSuRe, we are unable to provide a global convergence analysis. Instead, we resort to local analysis, where the desired structure of partitions is captured by a local, and not necessarily global, optimal point. As such, our analysis is generally conditioned on a proper initialization. However, we empirically observe that random initializations are often satisfactory.

1.1. Related Work

In comparison to the classical clustering methods such as K-means, subspace clustering is a relatively new topic in machine learning. The UoS model dates back to the seminal work of Berger and Sinclar [18]. The CLIQUE algorithm in [3] is one of the first proposals for solving the subspace clustering problem. Since then, many other techniques have been introduced in the context of clustering with dimensionality reduction. Extensive reviews of these methods can be found in [4, 19, 20] as well as in [21]. The book [22] also includes a chapter on subspace clustering. The paper [9] was one of the first works, discussing the idea of sparse subspace clustering. A generic analysis of SSC is given in [23], where the geometric concept of subspace affinity was introduced. Our analysis employs subspace affinity and other similar geometric ideas to the ones in this paper. More recently, the problem of robust subspace clustering was considered in different studies such as [11, 12, 24]. In this context, the RoSuRe algorithm was first introduced in [14] and immediately received attention in the background/forground separation problem [25]. Another attempt for utilizing subspace structure for this problem is found in [26]. Our focus is on the RoSuRe optimization framework. Although the other robust subspace clustering studies provide guarantees of similar nature to ours, their underlying models are different from RoSuRe and direct comparison is not straightforward. Related to implementation, a number of studies address online algorithms for robust principal component analysis (RPCA) and subspace tracking [27, 28, 29]. Moreover, there have been few attempts for subspace clustering with sequential data [30], but we are not aware of any previous proposal for sequential implementation of RoSuRe.

2. Algorithmic and Numerical Implementation

2.1. Solution by Quadratic Penalty Function

One way to solve (2) is to employ a quadratic penalty function to account for the constraint to yield the following unconstrained optimization:

$$\min \|\mathbf{W}\|_{1,1}+\lambda\|\mathbf{E}'\|_{1,1}+\frac{\mu}{2}\,\|(\mathbf{X}-\mathbf{E}')(\mathbf{W}-\mathbf{I})\|_{\mathrm{F}}^{2}\,, \quad (3)$$

where the minimization is taken over $\mathbf{W}\in\mathbb{R}_{\mathrm{d}}^{n\times n}$ and $\mathbf{E}'\in\mathbb{R}^{m\times n}$. Note that the optimal solution of (3) is identical to that of (2) when μ increases to infinity. However, solving (3) with a large value of μ requires a careful choice of the initialization, which leads one to rather proceed by means of gradually increasing μ. As a result, the solution in an early stage with a small value of μ, serves as a proper initial point for the later stages with larger μ. For simplicity, we fix μ to a moderately large value, which may lead to an adequately precise solution with no substantial effort for initialization. Due to non-smoothness in the objective function of (3), a proximal optimization method is used to solve (3). Each iteration of the resulting algorithm can be written as the following [13, 31]:

$$\text{Procedure } P: \quad \mathbf{W}_{t+1}=\mathcal{T}^{\mathrm{d}}_{\frac{\eta}{\mu}}\left[\mathbf{W}_t-\eta(\mathbf{X}-\mathbf{E}_t)^T\mathbf{Z}_t\right],$$
$$\mathbf{E}_{t+1}=\mathcal{T}_{\frac{\eta\lambda}{\mu}}\left[\mathbf{E}_t+\eta\mathbf{Z}_t(\mathbf{W}_t-\mathbf{I})^T\right], \quad (4)$$

where η is the step size, and we call $\mathbf{Z}_t=(\mathbf{X}-\mathbf{E}_t)(\mathbf{W}_t-\mathbf{I})$ the feasibility gap matrix as it represents the feasibility gap in (2). The operator $\mathcal{T}^{\mathrm{d}}_{.}(\,.\,)$ applies soft thresholding to every non-diagonal element, and sets the diagonal elements to zero. Notice that $\mathcal{T}_{.}(\,.\,)$ denotes elementwise soft thresholding.

2.2. An Online Sequential Algorithm

In this paper, we seek an online solution of the optimization in (2), where the data points $\mathbf{x}_i$ are provided sequentially, and the computational complexity is limited by the data rate. Note that the problem size increases by acquiring new observations, thus increasing the number of parameters to update in later iterations. This may quickly become intractable for problems of large size n. We hence propose a simplified approach, which, in general, leads to a good approximate solution of (2).

Our approach is to apply P in (4) at each iteration of the proposed algorithm. As the data sequentially streams in, each new data point is appended to the matrix $\mathbf{X}$, with the size of matrices $\mathbf{E}_t, \mathbf{W}_t$ simultaneously increasing. To control the complexity of the algorithm, we also delete from $\mathbf{X}$ old data points which have less contribution in the subsequent iterations. We propose two different criteria for selecting the old data points in the sequel. Finally, removing a data point from $\mathbf{X}$ entails re-calculating $\mathbf{W}$ and consequently reducing its dimension, as we elaborate below. The algorithmic procedure is detailed in Algorithm 1. Note that the proposed approach calls for dynamically variable size matrices, highlighted by $\mathbf{X}_t$ and n_t respectively denoting the matrix $\mathbf{X}$ and the cardinality of its column set at iteration t.

Algorithm 1 Sequential Bi-Sparsity Pursuit

```
Input Parameters $\alpha, \mu$ and $\eta$ and the maximum size $n_{\text{m}}$.
Initialize $\mathbf{W}_0 = [\,]$, $\mathbf{E}_0 = [\,]$, $\mathbf{X}_0 = [\,]$ and $t = 0$.
loop
    Obtain a new observation $\mathbf{x}_{t+1}$ and update $\mathbf{X}_{t+1} = [\mathbf{X}_t \; \mathbf{x}_{t+1}]$.
    Update
        $$\mathbf{W}_t \leftarrow \begin{bmatrix} \mathbf{W}_t & \mathbf{g}_1 \\ \mathbf{0} & g \end{bmatrix}, \quad \mathbf{E}_t \leftarrow [\mathbf{E}_t \; \mathbf{g}_2]$$
    where $\mathbf{g}_1, \mathbf{g}_2$ are standard Gaussian vectors, normalized to give $\mathcal{E}(\|\mathbf{g}_1\|^2) = \mathcal{E}(\|\mathbf{g}_2\|^2) = 1$ and $g$ is a standard Gaussian variable.
    Apply Procedure P in (4) and set $t \leftarrow t + 1$.
    if the maximum size of $\mathbf{X}_t$ is reached then
        Calculate the uncertainty and centrality levels $c_t^i, u_t^i$ and select the index $i$ with the smallest value $c_t^i + \alpha u_t^i$ (Quality Criterion), or set $i = 1$ (FIFO).
        Apply the update in (5), remove the $i^{\text{th}}$ column of $\mathbf{E}_t$ and $\mathbf{X}_t$ as well as the $i^{\text{th}}$ row and column of $\mathbf{W}_t$.
    end if
end loop
```

2.2.1 Recalculation after Deletion of a Data Point

To proceed, recall that the elements of $\mathbf{W}$ reflect the self representations of the denoised data vectors. Denote by $\mathbf{l}_t^i$ the i^{th} column of the denoised data matrix $\mathbf{L}_t = \mathbf{X}_t - \mathbf{E}_t$. Then, the definition of the feasibility gap matrix in (4) yields

$$\mathbf{l}_t^i = \sum_{k=1}^{n_t} w_{ki}(t)\mathbf{l}_t^k + \mathbf{z}_t^i,$$

where $w_{ki}(t)$ is the (k, i) element of $\mathbf{W}_t$, and $\mathbf{z}_t^i$ is the i^{th} column of $\mathbf{Z}_t$. Now, suppose that the i^{th} column of $\mathbf{X}_t$ is to be removed. For any other column $j \neq i$ we have that

$$\begin{aligned} \mathbf{l}_t^j &= \sum_{k \neq i} w_{kj}(t)\mathbf{l}_t^k + w_{ij}(t)\mathbf{l}_t^i + \mathbf{z}_t^j \\ &= \sum_{k \neq i} w_{kj}(t)\mathbf{l}_t^k + w_{ij}(t)\left(\sum_{k=1}^{n_t} w_{ki}(t)\mathbf{l}_t^k + \mathbf{z}_t^i\right) + \mathbf{z}_t^j \\ &= \sum_{k \neq i,j} \left[w_{kj}(t) + w_{ki}(t)w_{ij}(t)\right]\mathbf{l}_t^k + w_{ji}(t)w_{ij}(t)\mathbf{l}_t^j + w_{ij}(t)\mathbf{z}_t^i + \mathbf{z}_t^j. \end{aligned}$$

The first term in the above equation is the new representation, while the next three terms are the updated feasibility gap. This leads to the following iterative update:

$$\begin{aligned} w_{jk}(t) &\leftarrow \begin{cases} w_{kj}(t) + w_{ki}(t)w_{ij}(t) & k \neq j \\ 0 & k = j \end{cases}, \\ \mathbf{z}_t^j &\leftarrow w_{ji}(t)w_{ij}(t)\mathbf{l}_t^j + w_{ij}(t)\mathbf{z}_t^i + \mathbf{z}_t^j. \end{aligned} \tag{5}$$

In summary, when deleting column i from $\mathbf{X}_t$ and $\mathbf{E}_t$, we proceed to update $\mathbf{W}_t$ in (5) and also delete the i^{th} column and i^{th} row from $\mathbf{W}_t$. The omitted columns from $\mathbf{E}_t$ and $\mathbf{W}_t$ (before recalculation by (5)) are the final acceptable estimates for the dropped data point. They will be key components in configuring the similarity graph and the final clustering step, which is performed in a single batch. Notice that the error vectors from $\mathbf{E}_t$ does not need any further processing and are provided in real time.

2.2.2 Selecting Deleted Data Point

We consider the deletion of a single column i of $\mathbf{X}$ when the maximum allowable size n_{m} of $\mathbf{X}_t$ is attained. We discuss two different approaches to the selection of i:

First In First Out (FIFO): We simply remove the oldest sample in $\mathbf{X}_t$. If data is appended from right to $\mathbf{X}_t$, we remove the leftmost sample from $\mathbf{X}_t$ ($i = 1$). This is suitable for preserving the order of the samples in the output stream, e.g in video processing applications.

Quality Criteria: When the order of the output stream is not a concern, we may use the following criteria:

Uncertainty : The i^{th} column in $\mathbf{X}$ is said to be uncertain if applying the update in (4) substantially changes its corresponding representation (column of $\mathbf{W}$). Observe that the change in the i^{th} column of $\mathbf{W}$ is limited by the magnitude $u_t^i = \|\mathbf{z}_t^i\|$ of the i^{th} column of the feasibility gap matrix, which is the *uncertainty level* of the i^{th} column.

Centrality The i^{th} column of $\mathbf{X}$ is said to be noncentral in the representation $\mathbf{W}$ if the deletion step in (5) of this column does not substantially increase the objective value in (3). Given that the objective value in (3) can be written as $\|\mathbf{W}\|_{1,1}+\lambda\|\mathbf{E}\|_{1,1}+{}^{\mu}\!/_{2}\|\mathbf{Z}\|_{\mathrm{F}}^2$, we can simply find the following upper bound for the change of objective according to (5):

$$c_t^i = \sum_{j\neq i,k\neq i,j\neq k} |w_{ki}(t)||w_{ij}(t)|$$

$$+2\mu\sum_{j\neq i} w_{ji}^2(t)w_{ij}^2(t)L_{\max}^t + 2\mu\sum_{j\neq i} w_{ij}(t)\|\mathbf{z}_t^i\|_2^2,$$

where $L_{\max}^t$ is the maximum value of $\|\mathbf{l}_t^j\|$ for $j = 1, 2, \ldots$. We call c_t^i the *centrality level* of the i^{th} column.

These characteristics constitute a natural set of qualitative criteria to select the proper column in $\mathbf{X}_t$. Specifically, we select the column with the smallest $c_t^i + \alpha u_t^i$ where α is a design parameter.

3. Convergence Guarantees

The purpose of our analysis is to ensure that the optimization in (2) is capable of correctly recognizing clusters under sparse noise. To that end, we consider the following standard generative model: Consider a collection of K linear subspaces $\{\mathcal{L}_k \subseteq \mathbb{R}^m\}_{k=1}^K$, where m is the dimension of data samples and d_k denotes the dimension of $\mathcal{L}_k$. Take data samples $\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_n \in \mathbb{R}^m$ where each data sample belongs to exactly one of the subspaces $\mathcal{L}_k$. Suppose that each sample $\mathbf{l}_i$ is corrupted by a sparse additive noise $\mathbf{e}_i$. We observe the corrupted vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ which can be simply written as

$$\mathbf{x}_i = \mathbf{l}_i + \mathbf{e}_i, \quad i = 1, 2, \ldots, n. \tag{6}$$

Our objective is to guarantee that the optimization in (2) has a local optimal point satisfying the following *extended subspace detection* (ESD) properties:

1. The solution $\mathbf{E}$ coincides with the true noise, i.e. $\mathbf{E} = [\mathbf{e}_1\,\mathbf{e}_2\,\ldots\mathbf{e}_n]$.
2. For any pair $\mathbf{l}_i, \mathbf{l}_j$ of samples belonging to different subspaces, $(\mathbf{W})_{ij} = 0$.

We call such a local optimal solution an *ESD point*.

3.1. Previous Work: Noiseless Case

The noiseless case is discussed in [23], by using $\mathbf{E} = \mathbf{0}$ and reducing (2) to (1) with $\mathbf{X} = \mathbf{L} = [\mathbf{l}_1\,\mathbf{l}_2\ldots\mathbf{l}_n]$. The analysis of (1) is performed in three different ways: a) deterministic, where both subspaces and samples are specified, b) partially probabilistic, where the subspaces are specified but the samples are random and c) probabilistic, where both subspaces and samples are random. Our results in this paper are similar in spirit to the deterministic and partially probabilistic ones in [23]. For the sake of clarity, we only focus on the partially probabilistic results in this section and postpone the deterministic case to Section 4.

The partially probabilistic result in [23] concerns a case where the data points are uniformly sampled from the unit spheres in the subspaces, and the number of samples n_k from the k^{th} subspace is proportional to its dimension d_k. i.e. $n_k = \rho d_k$, where ρ is a constant known as a *sampling density*. The result is based on the following geometric notion of subspace affinity:

Definition 1. The affinity $\operatorname{aff}(\mathcal{L}_1, \mathcal{L}_2)$ between two subspaces $\mathcal{L}_1, \mathcal{L}_2$ is defined as

$$\operatorname{aff}(\mathcal{L}_1, \mathcal{L}_2) = \sqrt{\sum_k \cos^2(\theta_k)},$$

where $\theta_1, \theta_2, \ldots$ are the principal angles between $\mathcal{L}_1, \mathcal{L}_2$[1].

For convenience, we only state a simplified result where the dimension d_k of the subspaces are equal to d, and every subspace has an equal number of samples given by $n_k = \rho d = n/K$. Then [23] states (with some further simplifications) that there exists a constant $C = C(\rho)$ such that the solution of (1) satisfies the subspace detection property with probability at least $1 - {}^{1}\!/_{n^{10}}$ if for any two subspaces $\mathcal{L}_i, \mathcal{L}_j$:

$$\frac{\operatorname{aff}(\mathcal{L}_i, \mathcal{L}_j)}{\sqrt{d}} \leq \frac{1}{C\log n}. \tag{7}$$

This result essentially shows that as d (and consequently $n = K\rho d$) grows to infinity, subspace clustering is successful under (7) with exceeding probability. Conversely, it is more difficult to detect lower dimensional subspaces with a fixed amount of affinity. The choice of power 10 is also arbitrary, and can be replaced by any other positive number with a different choice of L.

3.2. Partially Probabilistic Guarantee with Adversarial Noise

Our study is in the same spirit as the noiseless analysis in Section 3.1. However, introducing a sparse noise in (2) restricts some of the conditions. In particular, we require the following additional factors in our analysis:

Definition 2. We say that a subspace $\mathcal{L} \subseteq \mathbb{R}^m$ is $(r, \epsilon)-$balanced if for any r distinct indices $i_1, i_2, \ldots, i_r$ in $[m]$ and values $x_1, x_2, \ldots, x_r$ in the interval $[-1\ 1]$, there exists a vector $\mathbf{y} = (y_1, y_2, \ldots, y_m) \in \mathcal{L}$ such that $y_{i_k} = x_k$ for $k = 1, 2, \ldots, r$ and $\forall j \notin \{i_1, i_2, \ldots, i_r\}$, $|y_j| < \epsilon$.

[1] For more details see [23].

Definition 3. We define ambiguity of a subspace $\mathcal{L} \subseteq \mathbb{R}^m$ as the maximal number of zero entries in a nonzero vector of $\mathcal{L}$.

In addition, a difficulty with the analysis of (2) is its nonconvexity and consequently propensity for local minima. Our analysis guarantees existence of a local minimum point with the desired ESD property. Our numerical results further suggest that in large-scale problems with local search methods, this local minimum is likely to be attained by a random initialization and a careful choice of design parameters.

To proceed, we provide the following natural characteristics for the noise vectors:

Definition 4. For a sequence $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ of sparse noise vectors we define the *noise level* m_{e} as the maximum number of non-zero entries the noise vectors. We also define the *incidence number* n_{e} as the maximum number of noise vectors with nonzero values in some common entry.

We express a simplified result which is comparable to the condition in (7). A general case is found in the supplement.

Theorem 1. *Consider a constant number K of subspaces with equal dimension d in $\mathbb{R}^m$. Supposed that for an absolute constant ρ, a data set of size ρd is sampled independently and randomly from each subspace $\mathcal{L}_k$ by orthogonally projecting a standard Gaussian vector onto $\mathcal{L}_k$. Denote the noise level and the incidence number of the sparse noise by m_e and n_e, respectively. Further, denote the maximal ambiguity of the subspaces $\mathcal{L}_k$ by s. Then, the optimization in* (2) *with a sufficiently large value of λ has an ESD point with probability* $1 - {}^1\!/n^{10}$[2] *if*

$$\frac{\mathrm{aff}(\mathcal{L}_i,\mathcal{L}_j)}{\sqrt{d}} \le \frac{1}{C \log n},$$
$$n_e \le \frac{1}{C}\left(\frac{n}{\log n}\right)^{\frac{1}{4}}, \quad m > \rho d m_e + s, \tag{8}$$

where C is a constant only depending on ρ, and the orthogonal complement of $\mathcal{L}_1 \oplus \mathcal{L}_2 \oplus \dots \oplus \mathcal{L}_K$[3] *is $(m_e, \frac{1}{C\sqrt{d}})-$balanced.*

Proof. The proof is discussed in Section 4. □

3.3. Discussion

As clearly evident, our result above is tightly connected to the geometric analysis of [23], with substantial differences that we clarify in the sequel. An important observation is that the "embedding" dimension m is absent in (7), since the space $\mathbf{R}^m$ only serves as an "embedding" or "representation" space for the problem in (7). As evident in (8), the induced noise changes the problem. In fact, the condition $m > \rho d m_{\mathrm{e}} + s$ suggests that in spite of a linear relation between n and d, the embedding dimension m should grow super-linearly for an interesting range of increasing noise level m_e. While in many typical cases, the ambiguity s is in the order of the dimension d, the required dimension m effectively amounts in this case, to the product of the noise level and the dimension.

Another interesting observation is that our analysis necessitates a highly balanced orthogonal complement of $\mathcal{L} = \mathcal{L}_1 \oplus \mathcal{L}_2 \oplus \dots \oplus \mathcal{L}_K$. This emphasizes that the dimension m should be much larger than the "total used" dimension of $\mathcal{L}$ which is in the range (in fact bounded by) of $K\rho d$, where K is the number of subspaces. In our application of interest, this is not a restrictive fact as the dimension m is the number of pixels in an image, which easily attains hundreds of thousands, while the $\rho d K = nK$ hardly exceeds few hundreds.

A more restrictive factor in Theorem 1 is the growth rate for the noise incidence number $n_{\mathrm{e}} = o(n^{\frac{1}{4}})$. For this, recall that the noise in our analysis has an adversarial nature. In other words, we consider a worst-case-scenario with respect to the error. Notice that a random noise corresponds to $n_{\mathrm{e}} = O(n)$, which is not supported by our analysis. However, we can lift this restriction by considering a fully probabilistic analysis. We postpone a careful study of this case to a future paper, as the nature of the vision problem considered here, does not assume random sparse noise (foreground). In the vision setup, our analysis shows that the foreground should be relatively smaller than the background for a guaranteed performance, which often holds.

Finally, we note that we establish local optimality of the solution for any sufficiently large value of λ. This clearly shows that the local optimal point of interest is not always the global optimal point as the noise term in the global optimal point can be arbitrarily (and undesirably) small with an extremely large value of λ. This suggests that in practice, our desired local minimum point can be difficult to attain by an uncommonly large value of λ. The study of this phenomenon is also postponed to future work.

4. Technical Details of Convergence Analysis

Our strategy is similar to [23]: We first introduce a fully deterministic analysis, and next verify its conditions for the probabilistic model of the data and next randomize it to obtain the result in Theorem 1. We explain this procedure in this section.

4.1. Deterministic Guarantee

Without loss of generality, assume that the samples are ordered such that each sample from $\mathcal{L}_l$ appears in the sequence $\{\mathbf{l}_i\}_{i=1}^n$ before every sample from $\mathcal{L}_k$ with $k > l$.

[2] The choice of power 10 is arbitrary.

[3] The sum $\mathcal{L}_1 \oplus \mathcal{L}_2 \oplus \dots \oplus \mathcal{L}_K$ of subspaces is simply the subspace obtained by taking the linear span of their union.

Next, define $\mathbf{L} = [\mathbf{l}_1\ \mathbf{l}_2, \dots\ \mathbf{l}_n]$ and take $\mathbf{L}_k$ as the sub matrix of $\mathbf{L}$ consisting of all data points $\mathbf{l}_k$ from the subspace $\mathcal{L}_k$ and denote

$$\mathbf{W}_{0,k} = \arg \min_{\mathbf{W}\ |\ \mathbf{L}_k\mathbf{W}=\mathbf{L}_k} \|\mathbf{W}\|_{1,1}. \tag{9}$$

Take $\mathbf{Z}_{0,k}$ as the dual vectors at the optimal point of (9). In other words,

$$\mathbf{Z}_{0,k} = \arg \max_{\mathbf{Z}\ |\ \|\mathbf{Z}^T\mathbf{L}_k\|_{\infty,\infty}\leq 1} \langle \mathbf{Z}, \mathbf{L}_k \rangle.$$

Without loss of generality, we assume that each column of $\mathbf{Z}_{0,k}$ belongs to $\mathcal{L}_k$ (otherwise the projection of the columns onto $\mathcal{L}_k$ is also a valid dual vector). Define $\mathbf{W}_0$ as a block diagonal matrix where its k^{th} diagonal block is $\mathbf{W}_{0,k}$ and take $\mathbf{Z}_0 = [\mathbf{Z}_{0,1}\ \mathbf{Z}_{0,2} \dots \mathbf{Z}_{0,K}]$. The supports of $\mathbf{W}_0$ and $\mathbf{E}$ are respectively denoted by Ω and $\mathcal{E}$. Denote by Λ the collection of all indexes (i, j), for which $\mathbf{l}_i$ and $\mathbf{l}_j$ do not belong to the same subspace. We denote by $\mathcal{P}_\Omega$, $\mathcal{P}_\mathcal{E}$ and $\mathcal{P}_\Lambda$ the projection operators, which respectively stack the elements of their arguments on Ω, $\mathcal{E}$ and Λ in the returned vector. Similarly, $\mathcal{P}_{\Omega^c}$ and $\mathcal{P}_{\mathcal{E}^c}$ stack the off-support elements of their arguments. By finally defining $\hat{\mathbf{W}}_0 = \mathbf{W}_0 - \mathbf{I}$, we state the following result:

Theorem 2. *The pair* ($\mathbf{W} = \mathbf{W}_0, \mathbf{E}' = \mathbf{E}$) *is a local optimal solution of Eq. (2 in Paper), hence an ESD point, if the following conditions hold:*

Local Identifiability: *For any pair of matrices* $\Delta\mathbf{W}, \Delta\mathbf{E}$ *satisfying* $\mathbf{L}\Delta\mathbf{W} - \Delta\mathbf{E}\hat{\mathbf{W}} = 0$*, if* $\mathcal{P}_{\mathcal{E}^c}\Delta\mathbf{E} = \mathbf{0}$ *and* $\mathcal{P}_\Lambda\Delta\mathbf{W} = \mathbf{0}$*, then* $\Delta\mathbf{E} = \mathbf{0}$ *and* $\Delta\mathbf{W} = \mathbf{0}$.

Strong Dual Verifiers Property: *There exists a vector* $\mathbf{Z} \in \mathbb{R}^{m\times N}$ *and a number* $0 < \delta < 1$ *such that*

$$\left(\mathbf{L}^T\mathbf{Z}\right)_{ij} \begin{cases} = \operatorname{sgn}(W_{ij}) & W_{ij}, \neq 0 \\ \in [-1\ 1] & W_{ij} = 0,\ (i,j) \notin \Lambda, \\ \in [-\delta\ \delta] & (i,j) \in \Lambda, \end{cases}$$

and

$$\left(\mathbf{Z}\hat{\mathbf{W}}_0^T\right)_{ij} \begin{cases} = -\lambda\operatorname{sgn}(E_{ij}) & E_{ij} \neq 0, \\ \in [-\lambda\delta\ \lambda\delta] & E_{ij} = 0, \end{cases}$$

Proof. The proof can be found in the supplement. □

4.1.1 Simplified Dual Verifiers Property

A refinement of the result in Theorem 2 follows by providing the dual verifier matrix $\mathbf{Z}$. Denote the orthogonal complement of a subspace $\mathcal{L}$ by $\mathcal{L}^\perp$, and introduce the following,

Definition 5. We say that a square matrix $\mathbf{W} = (W_{ij}) \in \mathbb{R}^{p\times p}$ is (α, β, q)-regular if for every $i \in \{1, 2, \dots, p\}$, and every subset $J \subseteq \{1, 2, \dots, p\}$ with $|J| \leq q$, we have $\sum_{j=1}^{p} |W_{i,j}| \leq \alpha,\ \sum_{j\in J} |W_{i,j}| \leq \beta$.

We now state the following lemma:

Lemma 1. *Suppose the number of nonzero entries in each row and column of* $\mathbf{E}$ *is bounded by* n_e *and* m_e*, respectively. Furthermore, the matrix* $\mathbf{W}_0$ *is* (α, β, n_e)*-regular, and the subspace* $(\mathcal{L}_1 \oplus \mathcal{L}_2 \oplus \dots \oplus \mathcal{L}_K)^\perp$ *is nonzero and* (m_e, ϵ)*-balanced. Set* $\eta = \max_{i,j} |[\mathbf{Z}_0\mathbf{W}_o^T]_{i,j}|$ *the largest absolute value in* $\mathbf{Z}_0\mathbf{W}_0^T$*, and Suppose that for any* $(i, j) \in \Lambda$*,* $|\mathbf{l}_i^T\mathbf{z}_j| < \delta$ *where* $\delta < 1$ *is a constant. Then, there exists a matrix* $\mathbf{Z}$ *satisfying the strong dual verifier condition in Theorem 2 if* $\epsilon\alpha + \beta < {}^{\delta}/_{2}$ *and* $\lambda > {}^{\eta}/_{(\delta-2(\alpha\epsilon+\beta))}$.

Proof. The proof can be found in the supplement. □

4.1.2 Final Step in Proof of Theorem 1

To prove theorem 1, we need to show that the local identifiability condition in Theorem 2 and the conditions of Lemma 1 are satisfied with high probability. We perform this by introducing the following result:

Lemma 2. *Suppose that* $\mathbf{a} \in \mathbb{R}^m$ *and each column of a matrix* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *are independently generated by taking a standard Gaussian vector and projecting it to a subspace* S *with dimension* d*. Assume that* $n = \rho d$ *for a constant* $\rho > 0$*. Denote respectively by* $\mathbf{w} \in \mathbb{R}^n$ *and* $\mathbf{z} \in \mathbb{R}^m$ *the solution and its corresponding dual vector of the following optimization:*

$$\min_{\mathbf{w}\in\mathbb{R}^n} \|\mathbf{w}\|_1 \text{ subject to } \mathbf{a} = \mathbf{A}\mathbf{w}$$

Then,

a) There exists a constant ω *only depending on* ρ *such that*

$$\Pr\left(\left|\frac{\|\mathbf{w}\|_1}{\sqrt{d}} - \omega\right| > \epsilon\right) < \frac{L}{\epsilon^2}e^{-cn\epsilon^2}$$

b) We have

$$\Pr\left(\|\mathbf{w}\|_\infty > \delta\right) \leq \frac{Ln}{\delta^4}e^{-cn\delta^4}$$

c) If $\mathbf{g}$ *is obtained by independently generating a standard Gaussian vector and projecting it onto a subspace* T*, then*

$$\Pr\left(\mathbf{z}^T\mathbf{g} > \frac{\operatorname{taff}(S,T)}{\sqrt{d}}\right) \leq Le^{-ct}$$

Proof. The proof of part (a) is given in [32]. Part (b) is also similarly obtained by the approach in [32] and noticing that if $\|\mathbf{w}\|_\infty > \delta$ the objective function increases according to the approach in [32] and hence it cannot be the global solution. The third part is also given in [23]. □

Once this result is obtained, we can conclude Theorem 1 as follows: We take $\delta = L\left(\frac{\log n}{n}\right)^{\frac{1}{4}}$ and $t = L\log n$ in

Lemma 2 , and $\epsilon = \frac{1}{L\sqrt{d}}$, $\delta = {}^{3}\!/_{4}, \alpha = \frac{1}{8\epsilon}$ and $\beta = {}^{1}\!/_{8}$ in Lemma 1 we obtain that with probability larger than $1 - \frac{1}{n^{10}}$ for all k:

$$\|\mathbf{w}_k\|_1 \leq \omega\sqrt{d}, \quad \|\mathbf{w}_k\|_\infty < \frac{1}{L}\left(\frac{\log n}{n}\right)^{\frac{1}{4}},$$

$$|\mathbf{z}_k \mathbf{l}_l| \leq \frac{\mathrm{aff}(\mathcal{L}_1, \mathcal{L}_2)\log n}{L\sqrt{d}},$$

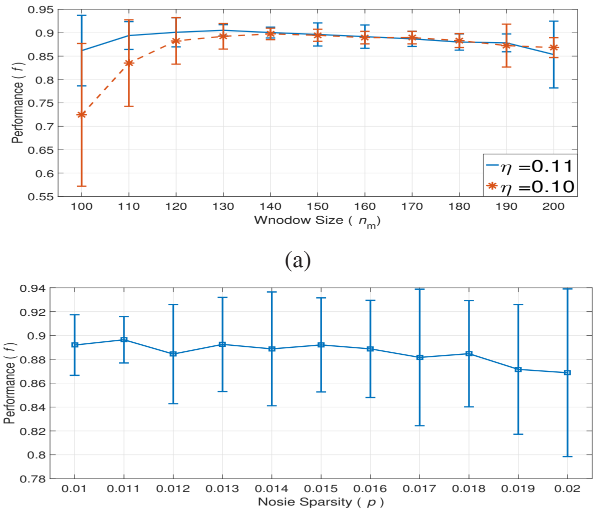
where $\mathbf{w}_k$ is the k^{th} column of $\mathbf{W}$ and $\mathbf{l}_k, \mathbf{l}_l$ belong to $\mathcal{L}_1, \mathcal{L}_2$, respectively. Then, it is simple to see that under the assumptions of Theorem 1, the conditions of Lemma 1 and, as a result, the strong dual verifiers condition holds. Finally, to check local identifiability notice that $\mathbf{L}\Delta\mathbf{W} = \Delta\mathbf{E}\hat{\mathbf{W}}$ implies that each column of $\Delta\mathbf{E}_k\mathbf{W}_{0,k}$ is in $\mathcal{L}_k$, where $\Delta\mathbf{E}_k$ is a block of $\Delta\mathbf{E}$ corresponding to $\mathbf{L}_k$. This means that the number of nonzero elements in $\Delta\mathbf{E}_k\mathbf{W}_{0,k}$, which is bounded by $\rho d m_e$, is larger than $m - s$ which contradicts the assumption. This completes the proof.

5. Numerical Results

To investigate the performance of the proposed online learning algorithm, we consider two different experimental scenarios. In the first experiment, we consider synthetic data obtained by the probabilistic model in Section 3. In the second experiment, we apply our algorithm to a real-world video sequence. The superiority of non-sequential RoSuRe to the state of the art methods in these setups is readily shown in [13, 14]. Our numerical results show that the sequential implementation can obtain similar performance to the batch procedure in [13, 14] and hence improves the state-of-the-art, from both performance and computation perspectives.

5.1. Synthetic Experiment

We consider three $d = 5$ dimensional subspaces $\mathcal{L}_1, \mathcal{L}_2$ and $\mathcal{L}_3$ in an $m = 200$ dimensional space, which are represented as the column spaces of there 200×5 matrices $\mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{A}_3$, respectively. We generate the entries of $\mathbf{A}_k$ for $k = 1, 2$ and 3 independently and randomly from a standard Gaussian distribution and then normalize their columns to have unit magnitude. Each noiseless data sample $\mathbf{l}_t$ is generated by selecting a subspace $k = 1, 2, 3$ with probabilities $0.35, 0.35, 0.3$, respectively, and multiplying $\mathbf{A}_k$ to a $5-$dimensional standard Gaussian vector normalized by ${}^{1}\!/\!\sqrt{5}$. The observed vector is generated by adding an independent identically distributed (i.i.d) sparse noise vector $\mathbf{e}_t$. Each entry of $\mathbf{e}_t$ is either zero with a probability p, or randomly generated by a standard Gaussian distribution. In our algorithm, we set $\lambda = 1$ and $\mu = 10$, and use the quality criterion in Section 2.2.2 with $\alpha = 20$.



(a)

(b)

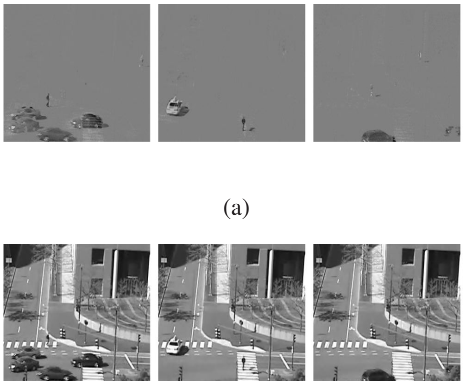
Figure 1: a)Performance versus window size for different values of η and $p = 0.02$. b) Performance versus window size for $n_{\mathrm{m}} = 100, \eta = 0.11$.

We consider a scenario with $1000 + n_{\mathrm{m}}$ samples, where n_{m} is the window size (maximum number of samples under processes). After deletion of each column from $\mathbf{W}_t$, it is stored in a matrix $\mathbf{W}_{\mathrm{final}}$ for calculating the clusters. We obtain the clusters after completion of the online learning stage. For this purpose, we define a symmetric affinity matrix $\mathbf{H}$, where

$$(\mathbf{H})_{i,j} = \begin{cases} 1 & (\mathbf{W}_{\mathrm{final}})_{i,j} + (\mathbf{W}_{\mathrm{final}})_{j,i} \geq 0.01 \\ 0 & \text{otherwise} \end{cases},$$

and perform spectral clustering on $\mathbf{H}$ [33]. We calculate the percentage f of correctly clustered data points for different values of p, η and n_{m}. Notice that we do not count the undeleted samples in $\mathbf{X}_t$ and hence the fraction f is always calculated over 1000 deleted samples.

Figure 1a depicts performance f averaged over 100 independent trials versus window size for two different values of $\eta = 0.1$ and $\eta = 0.11$. The error bars show the sample variance over the trials. Smaller variance reflects a more stable behavior in our algorithm. We observe that increasing the window size and/or the step size does not always improve the performance of our algorithm. This is because increasing the window size may reduce the speed of convergence, while increasing the step size can lead to a divergent solution. Small values of the step size may also lead to unresolvable clusters in $\mathbf{H}$. Another observation in Figure 1a is that different ranges of window size requires readjustment of the step size to provide best performance and stability. In particular, window sizes of size 140 and larger perform better with $\eta = 0.10$, while smaller windows are more accurate with $\eta = 0.11$. Figure 1b also shows average performance



(a)

(b)

Figure 2: a)The resulting foreground for frames 1, 60, 180. b) The original frames.

versus noise sparsity level p with $\eta = 0.11$ and $n_{\mathrm{m}} = 100$. Clearly, a smaller noise leads to a higher performance.

5.2. Foreground-Background Decomposition

5.2.1 MIT Traffic Dataset

We consider the MIT traffic data set [34], where the goal is to decompose the video sequence into a foreground and a background. The foreground can be used for surveillance purposes. We consider 300 frames of a 30-second video (10 frames per second), down-sampled to the resolution 240×360 and set $\mu = 10^3$, $\eta = 0.02$ and window size $n_{\mathrm{m}} = 100$. We also utilize FIFO deletion rule in Section 2.2.2. The algorithm runs on MATLAB R2016a with a 3GHZ CPU with rate 0.43 seconds per frame (2 frames/second). We also attain the rate 0.24 seconds per frame with window size 50, which slightly degrades the performance. Note that the overall complexity of our algorithm is $O(Tmn_{\mathrm{m}}^2)$ where T is the number of frames, while the complexity of the batch process is $O(mT^2)$ per iteration with a considerable number of iterations for convergence. This shows the remarkable advantage of the sequential method over the batch process. Figure 2 depicts few slides of the resulting foreground ($\mathbf{E}$), which shows an excellent identification of fast moving objects such as vehicles and pedestrians.

5.2.2 Microsoft Wallflower Dataset

In a different experiment, we compare our proposed method to the Grassmannian Robust Adaptive Subspace Tracking (GRASTA) Algorithm in [28] by considering the ”bootstrap” segment from the wallpaper dataset [35]. The parameters for the proposed algorithm is similar to the previous section with window length 100. For GRASTA, we track a $d = 5$ dimensional subspace. We set a maximum of 100 iterations in the internal loop of representation learning for a similar computational complexity to the proposed approach.

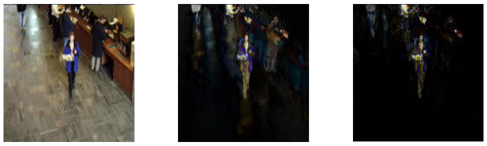


Figure 3: Wallflower bootstrap dataset frame 300. From left: Original frame, GRASTA and proposed method

	Prop. 100	Prop. 50	GRASTA	Wallflower
FP	379	357	391	356
FN	1471	9544	3658	2025
F_1 index	0.6112	0.3416	0.5559	
Best F_1	0.6505	0.6375	0.6714	

Figure 4: Number of false positive (FP) and false negative (FN) samples of different techniques as well as F index.

Figure 3 shows the result at the frame number 300, for which a hand segmented figure exists and is used for quantitavely studying the result in Figure 4. Clearly, GRASTA leads to a larger part of background incorporated in the foreground. In Figure 4, two threshod values for each method is used. One value is adjusted to provide around 350 false positive pixels. The other is adjusted for the best F value. Interestingly, GRASTA can lead to a slightly better F value but it requires a large threshold which is not practical.

6. Conclusion

We considered the problem of robust subspace clustering under sparse noise by bi-sparsity pursuit and its application to a video foreground/background decomposition problem. We presented an analysis of bi-sparsity pursuit, which provides bounds on the level of noise based on the coherence between subspaces, presented by the affinity measure. This ties our analysis to the previous studies of (noiseless) sparsity-based subspace clustering techniques. We also proposed a sequential implementation of the underlying optimization, which is suitable for online and real-time video processing applications. The results on the real-world data shows that we can easily attain a real-time implementation of our algorithm by improving computational resources, better programming and parallelization. As seen in the results, the performance of our algorithm highly depends on the step size. Hence, theoretical analysis of the effect of step size and the possibility of adaptive step size selection should be considered in a future study.

References

[1] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[2] 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.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*. ACM, 1998, vol. 27, no. 2.

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

[5] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, vol. 1. IEEE, 2003, pp. I–I.

[6] I. S. Kohane, A. J. Butte, and A. Kho, *Microarrays for an integrative genomics*. MIT press, 2002.

[7] S. Raychaudhuri, P. D. Sutphin, J. T. Chang, and R. B. Altman, "Basic microarray analysis: grouping and feature reduction," *TRENDS in Biotechnology*, vol. 19, no. 5, pp. 189–193, 2001.

[8] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using powerfactorization and gpca," *International Journal of Computer Vision*, vol. 79, no. 1, pp. 85–105, 2008.

[9] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2790–2797.

[10] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2849–2853.

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

[12] M. Soltanolkotabi, E. Elhamifar, E. J. Candes *et al.*, "Robust subspace clustering," *The Annals of Statistics*, vol. 42, no. 2, pp. 669–699, 2014.

[13] X. Bian and H. Krim, "Bi-sparsity pursuit for robust subspace recovery," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3535–3539.

[14] ——, "Robust subspace recovery via bi-sparsity pursuit," *arXiv preprint arXiv:1403.8067*, 2014.

[15] D. P. Bertsekas, "Incremental proximal methods for large scale convex optimization," *Mathematical programming*, vol. 129, no. 2, p. 163, 2011.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[17] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[18] R. L. Berger and D. F. Sinclair, "Testing hypotheses concerning unions of linear subspaces," *Journal of the American Statistical Association*, vol. 79, no. 385, pp. 158–163, 1984.

[19] 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.

[20] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *Acm Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[21] G. Gan and J. Wu, "Subspace clustering for high dimensional categorical data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 2, pp. 87–94, 2004.

[22] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[23] 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.

[24] Q. Qiu and G. Sapiro, "Learning robust subspace clustering," *arXiv preprint arXiv:1308.0273*, 2013.

[25] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Computer Science Review*, 2016.

[26] S. Javed, A. Mahmood, T. Bouwmans, and S. K. Jung, "Background–foreground modeling based on spatiotemporal sparse subspace clustering," *IEEE Transactions on Image Processing*, vol. 26, no. 12, pp. 5840–5854, 2017.

[27] P. Rodriguez and B. Wohlberg, "A matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3414–3416.

[28] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1568–1575.

[29] F. Seidel, C. Hage, and M. Kleinsteuber, "prost: a smoothed ℓ_p-norm robust online subspace tracking method for background subtraction in video," *Machine vision and applications*, vol. 25, no. 5, pp. 1227–1240, 2014.

[30] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1019–1026.

[31] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 2015, vol. 228.

[32] C. Thrampoulidis, A. Panahi, D. Guo, and B. Hassibi, "Precise error analysis of the lasso," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3467–3471.

[33] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," in *NIPS*, vol. 14, no. 2, 2001, pp. 849–856.

[34] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 31, no. 3, pp. 539–555, 2009.

[35] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 255–261.