GROUP-SPARSE SUBSPACE CLUSTERING WITH MISSING DATA

D. Pimentel-Alarcón[1], *L. Balzano*[2], *R. Marcia*[3], *R. Nowak*[1], *R. Willett*[1]

[1] University of Wisconsin - Madison, [2] University of Michigan - Ann Arbor,
[3] University of California - Merced

ABSTRACT

This paper explores algorithms for subspace clustering with missing data. In many high-dimensional data analysis settings, data points lie in or near a union of subspaces. Subspace clustering is the process of estimating these subspaces and assigning each data point to one of them. However, in many modern applications the data are severely corrupted by missing values. This paper describes two novel methods for subspace clustering with missing data: (a) *group-sparse subspace clustering* (GSSC), which is based on group-sparsity and alternating minimization, and (b) *mixture subspace clustering* (MSC), which models each data point as a convex combination of its projections onto all subspaces in the union. Both of these algorithms are shown to converge to a local minimum, and experimental results show that they outperform the previous state-of-the-art, with GSSC yielding the highest overall clustering accuracy.

Index Terms— Low-rank matrix completion, low-dimensional models, lasso, sparsity, subspace clustering, missing data, alternating optimization, compressed sensing.

1. INTRODUCTION

Subspace clustering is a powerful data analysis tool for modeling high-dimensional data. One is given a data matrix $\mathbf{X}$ whose columns lie in the union of several unknown low-dimensional subspaces of $\mathbb{R}^d$. The goal is to infer the underlying subspaces and cluster the columns of $\mathbf{X}$ according to the subspaces [1]. This problem has attracted increasing attention in recent years, producing theory and methods to handle outliers [2–6], noisy measurements [7], privacy concerns [8], and data constraints [9], among other issues. However, subspace clustering with missing data (SCMD), where one aims at the same goal but assumes that $\mathbf{X}$ is only partially observed, remains a challenging task. This scenario arises in a wide variety of modern applications, ranging from computer vision [10] to network estimation [11, 12] and recommender systems [13, 14].

SCMD can equivalently be thought of as a generalization of low-rank matrix completion [15] to the case where columns lie in several subspaces, rather than just one. Hence some approaches to SCMD use a combination of subspace clustering and matrix completion algorithms. For example, if the data lies in a union of subspaces, but remains low-rank, one option is to first complete the matrix and then cluster the data according to their nearest subspaces. In practice, though, the number and dimensions of the subspaces may be sufficiently large such that the data matrix is full-rank, and hence low-rank matrix completion methods are not applicable.

Other approaches to SCMD aim to generalize existing methods of subspace clustering to the missing data case. For example, [16] proposes to fill all missing entries with a sensible value (e.g., zeros or means), and then use one of the most popular algorithms for subspace clustering: sparse subspace clustering [17]. Unfortunately, this method provides no guarantees and may require a large amount of information, either in terms of number of columns or number of observations per column. In contrast, [18] gives conditions to provably perform SCMD using local neighborhoods; this method may also require a large amount of information. The main caveat of these approaches is that they aim to cluster a few columns at a time, and this requires many columns to be observed on the same rows, which is highly unlikely for severely incomplete datasets.

Rather than clustering a few columns at a time, other approaches aim to directly find a collection of subspaces that fits the entire dataset. One way to do this is by modeling a union of subspaces as a Gaussian mixture and to use an EM algorithm [19]. Unfortunately many datasets do not follow the Gaussian assumption. Furthermore, EM algorithms can only be guaranteed to converge to a local minimum. Another approach with similar drawbacks [20] generalizes the well-known k-subspaces clustering method to handle missing data. Hence there is interest in new SCMD algorithms that can avoid these drawbacks.

The main contribution of this paper is a novel algorithm tailored specifically for SCMD, based on group-sparsity and alternating minimization. We call this algorithm *group-sparse subspace clustering* (GSSC). Additionally, we present a second algorithm, which we call *mixture subspace clustering* (MSC). MSC is similar in principle and performance to EM, but for general data (not just Gaussian). Our main result shows that these algorithms will converge to a local minimum, and our experiments show how they outperform state of the art methods.

Organization of the paper

In Section 2 we formally state the problem and our main results, which we show in Section 3. In Section 4 we present experiments that show the performance of our algorithms, and in Section 5 we give a brief discussion of our methods.

2. MODEL AND MAIN RESULTS

Let $\mathcal{U}^\star := \{S_k^\star\}_{k=1}^K$ denote a collection of K unknown r-dimensional subspaces of $\mathbb{R}^d$. Let $\mathbf{X}$ be a $d \times N$ data matrix whose columns lie in the union of the subspaces in $\mathcal{U}^\star$. Let $\mathbf{\Omega}$ denote a $d \times N$ matrix with binary entries, and let $\mathbf{X}_\mathbf{\Omega}$ denote the incomplete version of $\mathbf{X}$, observed only in the nonzero entries of $\mathbf{\Omega}$. The goal is to infer $\mathcal{U}^\star$ from $\mathbf{X}_\mathbf{\Omega}$ and cluster the columns of $\mathbf{X}_\mathbf{\Omega}$ accordingly.

2.1. Group-Sparse Subspace Clustering (GSSC)

We now present our main contribution: a novel SCMD algorithm based on group-sparsity and alternating minimization.

Algorithm 1: Group-Sparse Subspace Clustering

Input: $\mathbf{X}_{\mathbf{\Omega}}, K, r, \lambda$.

Initialize $\widehat{\mathbf{U}} \in \mathbb{R}^{d \times Kr}$ (e.g., using SSC-EWZF).

repeat

$$\widehat{\mathbf{V}} = \arg\min_{\mathbf{V}} \|\mathbf{\Omega}(\mathbf{X} - \widehat{\mathbf{U}}\mathbf{V})\|_F^2 + \lambda \sum_{j,k=1}^{N,K} \|\mathbf{v}_{jk}\|_2.$$

$$\widehat{\mathbf{U}} = \arg\min_{\mathbf{U}\,:\,\|\mathbf{U}\|_F \leq 1} \|\mathbf{\Omega}(\mathbf{X} - \mathbf{U}\widehat{\mathbf{V}})\|_F.$$

until convergence;

Output: $\widehat{\mathbf{U}}, \widehat{\mathbf{V}}$.

Algorithm 2: Mixture Subspace Clustering

Input: $\mathbf{X}_{\mathbf{\Omega}}, K, \lambda$.

Initialize $\widehat{\mathbf{P}} \in \mathbb{R}^{KN \times N}$ (e.g., using SSC-EWZF).

repeat

$$\widehat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \|\mathbf{\Omega}(\mathbf{X} - \mathbf{Y}\widehat{\mathbf{P}})\|_F^2 + \lambda \sum_{k=1}^{K} \|\mathbf{Y}_k\|_*.$$

$$\widehat{\mathbf{P}} = \arg\min_{\mathbf{P} \in \Delta} \|\mathbf{\Omega}(\mathbf{X} - \widehat{\mathbf{Y}}\mathbf{P})\|_F.$$

until convergence;

Output: $\widehat{\mathbf{Y}}, \widehat{\mathbf{P}}$.

Let $\{\mathbf{U}_k^\star\}_{k=1}^K$ denote bases of the subspaces in $\mathcal{U}^\star$ and let $\mathbf{x}_j$ denote the j^{th} column of $\mathbf{X}$. If $\mathbf{x}_j$ lies in $S_k^\star$, then we can write

$$\mathbf{x}_j \;=\; \mathbf{U}_k^\star \mathbf{v}_j^\star, \tag{1}$$

where $\mathbf{v}_j^\star$ is a vector in $\mathbb{R}^r$ containing the coefficients of $\mathbf{x}_j$ in the basis $\mathbf{U}_k^\star$. Next let $\mathbf{V}_j^\star$ be a vector in $\mathbb{R}^{Kr}$ formed by K blocks of size $r \times 1$. If $\mathbf{x}_j \in S_k^\star$, let the k^{th} block of $\mathbf{V}_j^\star$ be $\mathbf{v}_j^\star$, and the remaining blocks be zero. For example, if $\mathbf{x}_j \in S_1^\star$, then

$$\mathbf{V}_j^\star \;=\; \begin{bmatrix} \mathbf{v}_j^\star \\ \hline \mathbf{0} \\ \hline \vdots \\ \hline \mathbf{0} \end{bmatrix} \begin{matrix} \}r \\ \}r \\ \\ \}r. \end{matrix}$$

Letting $\mathbf{U}^\star := [\mathbf{U}_1^\star \;\cdots\; \mathbf{U}_K^\star]$, we can rewrite (1) as $\mathbf{x}_j = \mathbf{U}^\star \mathbf{V}_j^\star$, and letting $\mathbf{V}^\star := [\mathbf{V}_1^\star \;\cdots\; \mathbf{V}_N^\star]$, it follows that

$$\mathbf{X} \;=\; \mathbf{U}^\star\, \mathbf{V}^\star.$$

Notice that $\mathbf{V}^\star$ is group-sparse, because if $\mathbf{x}_j$ lies in $S_k^\star$, then the j^{th} column of $\mathbf{V}^\star$ may only have nonzero entries in the k^{th} block, that is, in rows $(k-1)r+1, \ldots, kr$. This way, the group-sparsity pattern of $\mathbf{V}^\star$ encodes the information of the clustering of the columns of $\mathbf{X}$. On the other hand, $\mathbf{U}^\star$ determines $\mathcal{U}^\star$. Hence the goal is to estimate $\mathbf{U}^\star$ and $\mathbf{V}^\star$ from $\mathbf{X}_{\mathbf{\Omega}}$.

To this end, let $\mathbf{U} \in \mathbb{R}^{d \times Kr}$ and $\mathbf{V} \in \mathbb{R}^{Kr \times N}$. Let $\mathbf{V}_j$ denote the j^{th} column of $\mathbf{V}$. Then split $\mathbf{V}_j$ into k blocks of size $r \times 1$, and let $\mathbf{v}_{jk} \in \mathbb{R}^r$ denote the k^{th} block, i.e.,

$$\mathbf{V}_j \;=\; \begin{bmatrix} \mathbf{v}_{j1} \\ \hline \vdots \\ \hline \mathbf{v}_{jK} \end{bmatrix} \begin{matrix} \}r \\ \\ \}r. \end{matrix}$$

We would like to find matrices $\mathbf{U}$ and $\mathbf{V}$ that best explain the observed entries of $\mathbf{X}$, while encouraging the structure of $\mathbf{V}$ to resemble the group-sparse structure of $\mathbf{V}^\star$. This can be done by adding a group-lasso penalty (weighted by a parameter $\lambda \geq 0$) to encourage the columns of $\mathbf{V}$ to be group-sparse. Hence, ideally, we would like to find solutions to the following optimization problem:

$$\arg\min_{\substack{\mathbf{U},\mathbf{V}: \\ \|\mathbf{U}\|_F \leq 1}} \|\mathbf{\Omega}(\mathbf{X} - \mathbf{U}\mathbf{V})\|_F^2 + \lambda \sum_{j,k=1}^{N,K} \|\mathbf{v}_{jk}\|_2, \tag{2}$$

where $\mathbf{\Omega}(\cdot)$ denotes the Hadamard product with $\mathbf{\Omega}$. Unfortunately, (2) is non-convex, but with either variable $\mathbf{U}$ or $\mathbf{V}$ fixed, the function is convex in the other. GSSC aims to find a solution to (2) through alternating minimization. This is detailed in Algorithm 1.

Alternating minimization methods are generally not guaranteed to converge to a critical point [21, 22]. The main result of this paper is the following theorem. It states that the sequence generated by Algorithm 1 will contain a subsequence that converges to a critical point. The proof is given in Section 3.

Theorem 1. *Let $\{\widehat{\mathbf{U}}_t, \widehat{\mathbf{V}}_t\}_{t \geq 0}$ be the sequence of estimates generated by Algorithm 1. Then $\{\widehat{\mathbf{U}}_t, \widehat{\mathbf{V}}_t\}_{t \geq 0}$ has an accumulation point. Moreover, any accumulation point of $\{\widehat{\mathbf{U}}_t, \widehat{\mathbf{V}}_t\}_{t \geq 0}$ is a critical point of problem* (2).

2.2. Mixture Subspace Clustering (MSC)

Our second algorithm represents each column of $\mathbf{X}$ as a convex combination of the projection of that column onto each of the K subspaces. Hence we must estimate these projections and the weights associated with the convex combination for each column.

Let $\mathbf{Y}_k \in \mathbb{R}^{d \times N}$ denote the approximation of $\mathbf{X}$ in the k^{th} subspace, and let $\mathbf{Y} := [\mathbf{Y}_1 \;\cdots\; \mathbf{Y}_K] \in \mathbb{R}^{d \times KN}$. Let $p_{jk} \in [0, 1]$ denote the weight assigned to the subspace-k approximation of the j^{th} column of $\mathbf{X}$; we require $\sum_k p_{jk} = 1$. Let $\mathbf{P}_k \in \mathbb{R}^{N \times N}$ be a diagonal matrix with diagonal elements equal to $p_{1k}, \ldots, p_{Nk}$. Finally, let $\mathbf{P} := [\mathbf{P}_1 \;\cdots\; \mathbf{P}_K]^\top \in \mathbb{R}^{KN \times N}$, and let Δ denote the space of K stacked $N \times N$ diagonal matrices whose columns sum to 1 and whose entries are non-negative (i.e., matrices with the same form as $\mathbf{P}$).

Our mixture subspace clustering (MSC) algorithm estimates $\mathbf{P}$ and $\mathbf{Y}$ by solving the following optimization problem:

$$\arg\min_{\substack{\mathbf{Y},\mathbf{P}: \\ \mathbf{P} \in \Delta}} \|\mathbf{\Omega}(\mathbf{X} - \mathbf{Y}\mathbf{P})\|_F^2 + \lambda \sum_{k=1}^{K} \|\mathbf{Y}_k\|_*. \tag{3}$$

The alternating minimization procedure we use is detailed in Algorithm 2.

Similar to Theorem 1, we can show that the sequence generated by Algorithm 2 will contain a subsequence that converges to a critical point. The proof is analogous to that given in Section 3.

Theorem 2. *Let $\{\widehat{\mathbf{Y}}_t, \widehat{\mathbf{P}}_t\}_{t\geq 0}$ be the sequence of estimates generated by Algorithm 2. Then $\{\widehat{\mathbf{Y}}_t, \widehat{\mathbf{P}}_t\}_{t\geq 0}$ has an accumulation point. Moreover, any accumulation point of $\{\widehat{\mathbf{Y}}_t, \widehat{\mathbf{P}}_t\}_{t\geq 0}$ is a critical point of problem* (3)*.*

3. PROOF

In this section we give the proof of Theorem 1. The proof of Theorem 2 follows by similar arguments. Recall that $\mathbf{v}_{jk}$ denotes the k^{th} block of size $r \times 1$ of the j^{th} column of $\mathbf{V}$. It follows that

$$\left(\sum_{j,k=1}^{N,K} \|\mathbf{v}_{jk}\|_2 \right)^2 \geq \sum_{j,k=1}^{N,K} \|\mathbf{v}_{jk}\|_2^2 = \|\mathbf{V}\|_F^2.$$

Next define

$$\Phi(\mathbf{U}, \mathbf{V}) := \|\mathbf{\Omega}(\mathbf{X} - \mathbf{U}\mathbf{V})\|_F^2 + \lambda \sum_{j,k=1}^{N,K} \|\mathbf{v}_{jk}\|_2,$$

and observe that if $\|\mathbf{V}\|_F > \frac{1}{\lambda}\Phi(0,0) = \frac{1}{\lambda}\|\mathbf{\Omega}(\mathbf{X})\|_F^2$, then $\Phi(\mathbf{U}, \mathbf{V}) \geq \lambda \sum_{j,k=1}^{N,K} \|\mathbf{v}_{jk}\|_2 \geq \lambda \|\mathbf{V}\|_F > \Phi(0,0)$, which implies $\mathbf{V}$ is not a minimizer of (2). It follows that the solution to (2) is the same as the solution to

$$\underset{\substack{\mathbf{U},\mathbf{V}: \\ \|\mathbf{U}\|_F \leq 1}}{\arg\min}\ \Phi(\mathbf{U}, \mathbf{V}) \quad \text{s.t.} \quad \|\mathbf{V}\|_F \leq \frac{1}{\lambda}\|\mathbf{\Omega}(\mathbf{X})\|_F^2, \tag{4}$$

and that $\widehat{\mathbf{V}}$ in Algorithm 1 satisfies $\|\widehat{\mathbf{V}}\|_F \leq \frac{1}{\lambda}\|\mathbf{\Omega}(\mathbf{X})\|_F^2$. Thus, the sequence $\{\widehat{\mathbf{U}}_t, \widehat{\mathbf{V}}_t\}_{t\geq 0}$ generated by Algorithm 1 will be bounded. By the Bolzano-Weierstrass theorem, this sequence will contain a convergent subsequence. The limit of this subsequence will be an accumulation point of $\{\widehat{\mathbf{U}}_t, \widehat{\mathbf{V}}_t\}_{t\geq 0}$. Next, we demonstrate that any accumulation point will be a critical point. To this end, we will show that the conditions of Lemma 3.2 in [23] are satisfied for (4). Let

$$f(\mathbf{U}, \mathbf{V}) = \|\mathbf{\Omega}(\mathbf{X} - \mathbf{U}\mathbf{V})\|_F^2,$$
$$g_1(\mathbf{V}) = \sum_{j,k=1}^{N,K} \|\mathbf{v}_{jk}\|_2, \qquad g_2(\mathbf{U}) = 0.$$

Next notice that

(a) The function g_1 is closed, because g_1 is a continuous function with closed domain (see Sec. A.3.3 in [24]). In addition, g_1 is proper convex because g_1 is a norm (see [25]) and norms are convex and nonnegative. Also, g_1 is sub differentiable because it is differentiable everywhere except whenever $\mathbf{v}_{jk} = \mathbf{0}$, and because at $\mathbf{v}_{jk} = \mathbf{0}$, there exists a $\mathbf{W} \in \mathbb{R}^{dKr \times N}$ such that for every $\mathbf{Z} \in \text{dom } g_1$,

$$g_1(\mathbf{Z}) = \sum_{j,k=1}^{N,K} \|\mathbf{z}_{jk}\|_2 \geq \sum_{j,k=1}^{N,K} \|\mathbf{v}_{jk}\|_2 + \sum_{j,k=1}^{N,K} \mathbf{w}_{jk}^{\mathsf{T}}(\mathbf{z}_{jk} - \mathbf{v}_{jk}),$$

where $\mathbf{w}_{jk}$ and $\mathbf{z}_{jk}$ denote the blocks in $\mathbf{W}$ and $\mathbf{Z}$ corresponding to $\mathbf{v}_{jk}$. Specifically, $\mathbf{w}_{jk} = \mathbf{0}$ if $\mathbf{v}_{jk} = \mathbf{0}$ and $\mathbf{w}_{jk} = \mathbf{v}_{jk}/\|\mathbf{v}_{jk}\|_2$ otherwise. Finally, the function g_2 is also closed, proper convex, and sub differentiable because it is a constant.

(b) f is continuously differentiable because f is a polynomial (of degree 4).

(c) $\nabla_{\mathbf{U}} f$ is Lipschitz continuous with respect to $\mathbf{U}$, with Lipschitz constant $c = (\frac{1}{\lambda}\|\mathbf{\Omega}(\mathbf{X})\|_F^2)^2$, because for any $\mathbf{A} \in \mathbb{R}^{d \times Kr}$:

$$\begin{aligned}
&\|\nabla_{\mathbf{U}} f(\mathbf{U} + \mathbf{A}, \mathbf{V}) - \nabla_{\mathbf{U}} f(\mathbf{U}, \mathbf{V})\|_F \\
&= \|\mathbf{\Omega}(\mathbf{X} - (\mathbf{U} + \mathbf{A})\mathbf{V})\mathbf{V}^{\mathsf{T}} - \mathbf{\Omega}(\mathbf{X} - \mathbf{U}\mathbf{V})\mathbf{V}^{\mathsf{T}}\|_F \\
&= \| - \mathbf{\Omega}(\mathbf{A}\mathbf{V})\mathbf{V}^{\mathsf{T}}\|_F \leq \|\mathbf{\Omega}(\mathbf{A}\mathbf{V})\|_F \|\mathbf{V}\|_F \\
&\leq \|\mathbf{A}\mathbf{V}\|_F \|\mathbf{V}\|_F \leq \|\mathbf{A}\|_F \|\mathbf{V}\|_F^2 \leq c\|\mathbf{A}\|_F.
\end{aligned}$$

(d) $\nabla_{\mathbf{V}} f$ is Lipschitz continuous with respect to $\mathbf{V}$ for similar reasons as above, i.e., for any $\mathbf{A} \in \mathbb{R}^{Kr \times N}$:

$$\begin{aligned}
&\|\nabla_{\mathbf{V}} f(\mathbf{U}, \mathbf{V} + \mathbf{A}) - \nabla_{\mathbf{V}} f(\mathbf{U}, \mathbf{V})\|_F \\
&\leq \|\mathbf{U}\mathbf{A}\|_F \|\mathbf{U}\|_F \leq \|\mathbf{A}\|_F \|\mathbf{U}\|_F^2 \leq \|\mathbf{A}\|_F,
\end{aligned}$$

where the last equality follows because $\|\mathbf{U}\|_F \leq 1$.

(e) The function $\Phi(\mathbf{U}, \mathbf{V})$ and the objective functions in the two subproblems in Algorithm 1 are continuous with closed and bounded domains. By the Extreme Value Theorem, $\Phi(\mathbf{U}, \mathbf{V})$ must attain a minimum, and therefore, they must have a minimizer.

Conditions (a)-(e) are the assumptions of Lemma 3.2 in [23]. Theorem 1 follows directly by this result. □

4. EXPERIMENTS

In this section we present a series of experiments to study the performance of GSSC and MSC and compare them with the following SCMD algorithms:

- EM. This algorithm models data in a union of subspaces as a Gaussian mixture with low-rank covariance matrices, which are estimated using an expectation maximization algorithm [19].
- SSC-EWZF (sparse subspace clustering by entry-wise zero fill). This algorithm fills all missing entries with zeros and then uses sparse subspace clustering (SSC) to cluster the filled columns [16].
- MC+SSC (matrix completion plus sparse subspace clustering). This algorithm first completes the missing values using low-rank matrix completion methods and then uses SSC to cluster the completed columns [16].

We found that these methods, representative of the distinct approaches to SCMD, typically performed as well or better than several other SCMD algorithms. We point out that GSSC, MSC, MC+SSC and SSC-EWZF involve a penalty parameter λ, which was selected by cross-validation.

In our experiments we compare the behavior of these algorithms as a function of the ambient dimension d, the number of subspaces K, the dimension of each subspace r, the number of observed entries per column ℓ, and the number of columns per subspace N_k.

In each trial, we generated K subspaces, each spanned by r vectors in $\mathbb{R}^d$ drawn i.i.d. from the standard Gaussian distribution. We then generated N_k columns from each subspace, with coefficients

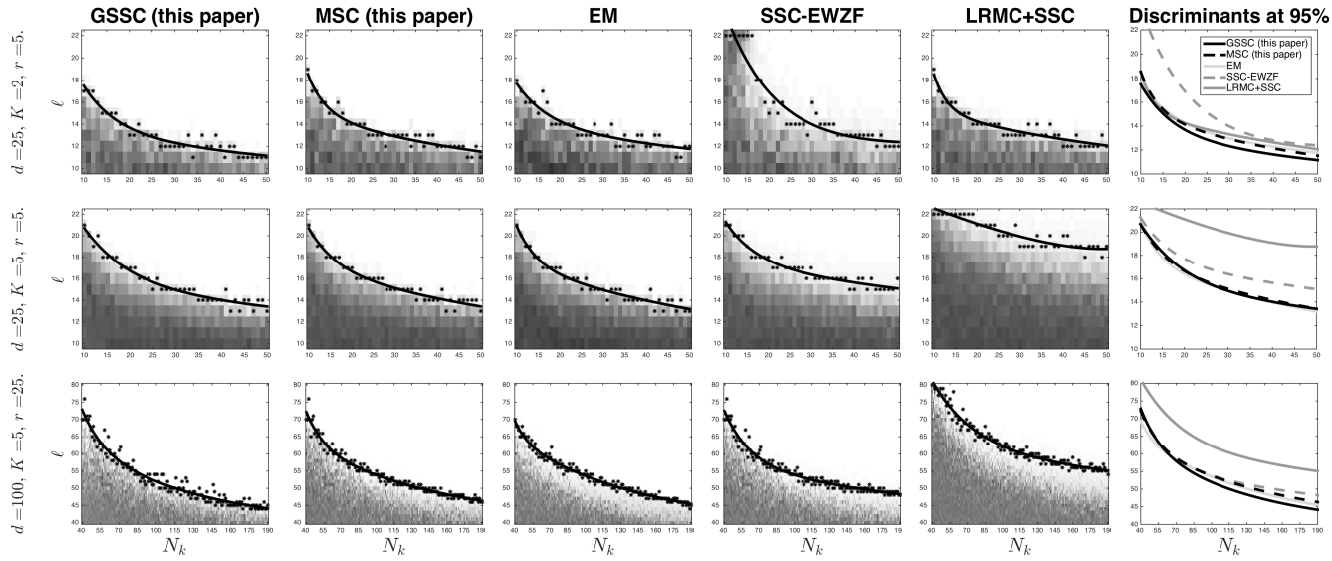

Fig. 1: Proportion of correctly classified points (average over 10 trials) of several SCMD algorithms as a function of the number of columns N_k and the number of observations per column ℓ for different values of the ambient dimension d, the number of subspaces K, and the dimension of each subspace r. White represents 100% accuracy, and the darkest pixel in each figure represents $1/K$, which amounts to random guessing. All pixels above the black point in each column have at least 95% accuracy. The curve is the best exponential fit to these points, and we plot them all simultaneously in the rightmost plot. These curves represent the discriminant between 95% accuracy (above curve) and less than 95% accuracy (below curve). The lower the curve the better.

also drawn i.i.d. from the standard Gaussian distribution. Next, we observed only ℓ entries per column, selected uniformly at random and independently across columns. Finally, we clustered the incomplete columns using the subspace clustering methods above.

Each cluster of columns defines an estimated subspace $\hat{S}$. We assign the columns corresponding to $\hat{S}$ to the subspace in $\mathcal{U}^\star$ that is closest to $\hat{S}$. To measure accuracy we count the number of columns that were correctly assigned. The results are shown in Figure 1. In the settings where $\mathbf{X}$ remained low-rank ($d = 25$, $K = 2$, $r = 5$), we used the output of MC+SSC as input to the rest of the algorithms. In the rest of the settings, we use the output of SSC-EWZF. One can see that our methods consistently outperform the state of the art methods SSC-EWZF and MC+SSC [16]. For all problem sizes, our methods result in a better clustering accuracy for a lower number of observed points ℓ.

5. CONCLUSIONS

In this paper we study algorithms for subspace clustering with missing data. We propose two novel methods for this task with local convergence guarantees: GSSC and MSC. Our experimental results show that these methods outperform previous state-of-the-art. Each of these methods has different advantages. For instance, MSC does not require one to know r, the dimension of the subspaces, while GSSC relies on this knowledge. On the other hand, GSSC encourages the columns of $\mathbf{X}$ to be associated with a single subspace, while MSC selects $\mathbf{P}$ on a collection of simplices, which precludes ℓ_1 sparsity regularization of the weights. We point out that for simplicity of presentation we had each subspace have the same dimension. Nonetheless, both MSC and GSSC admit a union of subspaces with different dimensions.

References

[1] R. Vidal, *Subspace clustering*, IEEE Signal Processing Magazine, 2011.

[2] G. Liu, Z. Lin and Y. Yu, *Robust subspace segmentation by low-rank representation*, International Conference on Machine Learning, 2010.

[3] 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, 2013.

[4] M. Soltanolkotabi, E. Elhamifar and E. Candès, *Robust subspace clustering*, Annals of Statistics, 2014.

[5] C. Qu and H. Xu, *Subspace clustering with irrelevant features via robust Dantzig selector*, Advances in Neural Information Processing Systems, 2015.

[6] X. Peng, Z. Yi and H. Tang, *Robust subspace clustering via thresholding ridge regression*, AAAI Conference on Artificial Intelligence, 2015.

[7] Y. Wang and H. Xu, *Noisy sparse subspace clustering*, International Conference on Machine Learning, 2013.

[8] Y. Wang, Y. Wang and A. Singh, *Differentially private subspace clustering*, Advances in Neural Information Processing Systems, 2015.

[9] H. Hu, J. Feng and J. Zhou, *Exploiting unsupervised and supervised constraints for subspace clustering*, IEEE Pattern Analysis and Machine Intelligence, 2015.

[10] K. Kanatani, *Motion segmentation by subspace separation and model selection*, IEEE International Conference in Computer Vision, 2001.

[11] B. Eriksson, P. Barford, J. Sommers and R. Nowak, *Domain-Impute: Inferring unseen components in the Internet*, IEEE INFOCOM Mini-Conference, 2011.

[12] G. Mateos and K. Rajawat, *Dynamic network cartography: Advances in network health monitoring*, IEEE Signal Processing Magazine, 2013.

[13] J. Rennie and N. Srebro, *Fast maximum margin matrix factorization for collaborative prediction*, International Conference on Machine Learning, 2005.

[14] A. Zhang, N. Fawaz, S. Ioannidis and A. Montanari, *Guess who rated this movie: Identifying users through subspace clustering*, Conference on Uncertainty in Artificial Intelligence, 2012.

[15] E. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics, 2009.

[16] C. Yang, D. Robinson and R. Vidal, *Sparse subspace clustering with missing entries*, International Conference on Machine Learning, 2015.

[17] E. Elhamifar and R. Vidal, *Sparse subspace clustering: algorithm, theory, and applications*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013.

[18] B. Eriksson, L. Balzano and R. Nowak, *High-rank matrix completion and subspace clustering with missing data*, Artificial Intelligence and Statistics, 2012.

[19] D. Pimentel-Alarcón, L. Balzano and R. Nowak, *On the sample complexity of subspace clustering with missing data*, IEEE Statistical Signal Processing, 2014.

[20] L. Balzano, R. Nowak, A. Szlam and B. Recht, *k-Subspaces with missing data*, IEEE Statistical Signal Processing, 2012.

[21] M. Powell, *On search directions for minimization algorithms*, Mathematical Programming, 1973.

[22] J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[23] A. Beck, *On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes*, SIAM Journal on Optimization, 2015.

[24] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[25] C. Ding, D. Zhou, X. He and H. Zha, *R1-PCA: Rotational invariant L1-norm principal component analysis for robust subspace factorization*, International Conference on Machine Learning, 2006.