Mixture Regression as Subspace Clustering

Daniel Pimentel-Alarcón, Laura Balzano, Roummel Marcia, Robert Nowak, Rebecca Willett

Abstract—**In this paper we show that observations in a mixture can be modeled using a union of subspaces, and hence mixture regression can be posed as a subspace clustering problem. This allows to perform mixture regression even in the presence of missing data. We illustrate this using a state-of-the-art subspace clustering algorithm for incomplete data to perform mixed linear regression on gene functional data. Our approach outperforms existing methods on this task.**

I. Introduction

One often wants to determine how a collection of variables $\mathbf{y} \in \mathbb{R}^q$ (e.g., glucose or cholesterol levels) depends on an other collection of variables $\mathbf{x} \in \mathbb{R}^p$ (e.g., diet, income, or education), using a set of *samples* $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$. Linear regression, or more generally a **mixture of linear regressors** is arguably the most used model for this purpose. This model assumes that the dependency is of the form

$$\mathbf{y}_i \;=\; \sum_{k=1}^{K} \mathbb{1}_{\{k_i=k\}} \mathbf{B}_k \mathbf{x}_i \;+\; \boldsymbol{\epsilon}_i, \tag{1}$$

where $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function, $k_i \in \{1, \ldots, K\}$ is a *hidden* variable (e.g., blood type) indicating that the i^{th} example depends on the ${k_i}^{\text{th}}$ regressor $\mathbf{B}_{k_i} \in \mathbb{R}^{q \times p}$, and $\boldsymbol{\epsilon}_i \in \mathbb{R}^q$ represents noise. The goal is to estimate the k_i's and the $\mathbf{B}_k$'s from the training examples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$. The k_i's and the $\mathbf{B}_k$'s determine the dependency that we aim to find. Notice that if we knew the k_i's, then we could partition the data accordingly, and estimate the $\mathbf{B}_k$'s using standard regression. The challenge is that we do not know the k_i's.

Linear mixtures are good models for practical applications as diverse as health care, image processing and classification. Hence there exists a wide variety of mixture regression methods. The most widely used algorithm is perhaps Expectation-Maximization (EM) [1]. Unfortunately, not all datasets follow the Gaussian assumption that it requires, and due to the non-convexity of the method, it can only be guaranteed to converge to a local minimum. Furthermore, new datasets pose new challenges, like missing data. Hence developing new and efficient mixture regression techniques is still an active field of research [2]–[5].

On the other hand, **subspace clustering** is a powerful tool to analyze high-dimensional data. One is given columns lying in the union of several unknown low-dimensional subspaces, and aims to infer the underlying subspaces and cluster the columns according to the subspaces [6]. Subspace clustering has applications in computer vision [7], network estimation [8], [9] and recommender systems [10], [11], to name a few. Hence it has attracted increasing attention in recent years, producing theory and methods to handle outliers [12]–[16], noisy measurements [17], privacy concerns [18], data constraints [19], and missing data [20]–[25].

In this paper we show that observations following (1) lie in a union of subspaces. Furthermore, we will see that if the dataset at hand has low intrinsic dimension (as it is often the case), then the underlying subspaces are low-dimensional. In this case we can use subspace clustering algorithms to perform mixture regression, even if data is highly incomplete. In fact, our approach works in the difficult regime where each sample has multiple responses (i.e., $q > 1$), many of which may be unobserved, as well as many of the independent variables. This is often known as multi-label transductive learning with missing data [26].

We illustrate our approach on a real-life gene functional dataset using two methods: first, a state-of-the-art subspace clustering algorithm for incomplete data, group-sparse subspace clustering (GSSC) [25], and second, a state-of-the-art method for multi-label transductive learning with missing data, matrix completion with bias (MC-b) [26]. Our approach outperforms existing methods on this task, showing the potential of subspace clustering algorithms on mixture regression.

We point out that **it has been previously noted** that mixture regression can be modeled using unions of subspaces [5]. However, this observation has received little attention. The reason is that the classical setup of regression deals with the case where there is only one response per observation (i.e., $\mathbf{y}_i \in \mathbb{R}$). As we will see in Section II, this yields observations on p-dimensional subspaces in $\mathbb{R}^{p+1}$, i.e., subspaces whose dimension is only one less than the ambient dimension (hyperplanes). Unfortunately, practical approaches to subspace clustering have poor performance in this setting. Furthermore, this leaves no room for missing data, as one fundamental requirement for clustering of p-dimensional subspaces is to observe at least $p+1$ entries per column [24], which would imply observing *all* data.

Fortunately, in many modern applications, each observation has $q > 1$ responses (i.e., $\mathbf{y}_i \in \mathbb{R}^q$), and the $\mathbf{x}_i$'s often have additional structure. As we will see in Section III, this yields observations lying near subspaces of dimension much lower than the ambient dimension, $q+p$. This allows a large portion of missing data, and there exist practical subspace clustering algorithms that work well under these settings [20]–[25].

Organization of the Paper

In Section II we show that observations in a mixture can be modeled using a union of subspaces. In Section III we show that if the $\mathbf{x}_i$'s have some additional structure, these subspaces are low-dimensional, whence one can use subspace clustering algorithms to perform mixture regression, even if data is missing. In Section IV we present our experiments on a real-life biological dataset. Section V includes a brief description of the methods we used for our experiments.

II. Mixtures Lie in Unions of Subspaces

In this section we show that observations in a mixture lie near a union of subspaces. First observe that in the noiseless setting, i.e., if $\boldsymbol{\epsilon}_i = \mathbf{0}$, (1) can be written as

$$\mathbf{y}_i \;=\; \mathbf{B}_{k_i}\mathbf{x}_i. \tag{2}$$

Define

$$\mathbf{z}_i \;:=\; \begin{bmatrix}\mathbf{y}_i\\ \mathbf{x}_i\end{bmatrix} \in \mathbb{R}^{q+p}. \tag{3}$$

Letting $\mathbf{I}_q$ denote the identity matrix of size $q \times q$, we can rewrite (2) as

$$\begin{bmatrix}-\mathbf{I}_q & \mathbf{B}_{k_i}\end{bmatrix}\mathbf{z}_i \;=\; \mathbf{0}. \tag{4}$$

Recall that $\mathbf{B}_{k_i} \in \mathbb{R}^{q\times p}$. Let $\mathbf{Z}_k$ be the matrix corresponding to the k^{th} regressor, i.e., the matrix formed with $\{\mathbf{z}_i : k_i = k\}$ as columns. It follows that $\mathbf{Z}_k$ lies in $\ker[-\mathbf{I}_q \;\; \mathbf{B}_{k_i}]$, which is a p-dimensional subspace in $\mathbb{R}^{q+p}$ (the q rows in $[-\mathbf{I}_q \;\; \mathbf{B}_{k_i}]$ are linearly independent because of the identity block).

Theoretically, one could use a subspace clustering algorithm to determine the hidden labels k_i's. Once this is known, we could partition the data accordingly, and learn the $\mathbf{B}_k$'s using standard regression. Alternatively, subspace clustering algorithms typically yield the underlying subspaces, in this case given by $\ker[-\mathbf{I}_q \;\; \mathbf{B}_k]$, whence the $\mathbf{B}_k$'s can be learned by simple inspection.

However, if q is small (for example, in classical mixture settings $q = 1$), then the columns in $\mathbf{Z}_k$ lie in a p-dimensional subspace in $\mathbb{R}^{p+1}$, i.e., a hyperplane, and practical subspace clustering algorithms have poor performance in this setting. Furthermore, since $p+1$ observations per column are theoretically necessary for subspace clustering [24], this leaves no room for missing data.

Fortunately, in many modern applications q is often large, and the $\mathbf{x}_i$'s have additional structure. As we will see in the next section, this results in observations lying on low-dimensional subspaces. This allows a large portion of missing data, and there exist practical subspace clustering algorithms that work well under these settings [20]–[25].

III. Mixture Regression as Subspace Clustering

In many applications of regression, the vectors $\mathbf{x}_i$ lie in r-dimensional subspaces of $\mathbb{R}^p$, $r < p$. In this section we will show that if this is the case, then the $\mathbf{z}_i$'s (as defined in (3)) also lie in subspaces of dimension r (as opposed to p), whence subspace clustering (and hence mixture regression) can be performed with even fewer observations (as little as $r+1$ instead of $p+1$). In other words, if the vectors $\mathbf{x}_i$ lie near low-dimensional subspaces, one can perform mixture regression with even fewer data.

To see this, let $\mathbf{X}_k$ denote the matrix containing the columns corresponding to the k^{th} regressor. More precisely, let $\mathbf{X}_k$ be the matrix formed with $\{\mathbf{x}_i : k_i = k\}$ as columns. Suppose $\mathbf{X}_k$ is rank-r, with $r < p$. Let $\mathbf{W}_k \in \mathbb{R}^{p\times r}$ be a basis of the r-dimensional subspace spanned by the columns in $\mathbf{X}_k$. We can assume without loss of generality that $\mathbf{W}_k$ is in the following column-echelon form (otherwise we can simply permute the rows of $\mathbf{X}_k$):

$$\mathbf{W}_k \;=\; \begin{bmatrix}\mathbf{W}'_k\\ \mathbf{I}_r\end{bmatrix}. \tag{5}$$

Let $\boldsymbol{\theta}_i \in \mathbb{R}^r$ be the coefficient vector of $\mathbf{x}_i$ in the basis $\mathbf{W}_{k_i}$, and let $\mathbf{x}_{\boldsymbol{\upsilon}_i} \in \mathbb{R}^{p-r}$ and $\mathbf{x}_{\omega_i} \in \mathbb{R}^r$ be the vectors with the first $p-r$ and the last r entries of $\mathbf{x}_i$ respectively, such that

$$\mathbf{x}_i \;=\; \begin{bmatrix}\mathbf{x}_{\boldsymbol{\upsilon}_i}\\ \mathbf{x}_{\omega_i}\end{bmatrix} \;=\; \begin{bmatrix}\mathbf{W}'_{k_i}\\ \mathbf{I}_r\end{bmatrix}\boldsymbol{\theta}_i.$$

From the bottom block we can see that $\boldsymbol{\theta}_i = \mathbf{x}_{\omega_i}$, and from the top block we can see that $\mathbf{x}_{\boldsymbol{\upsilon}_i} = \mathbf{W}'_{k_i}\mathbf{x}_{\omega_i}$. Equivalently,

$$\begin{bmatrix}-\mathbf{I}_{p-r} & \mathbf{W}'_{k_i}\end{bmatrix}\mathbf{x}_i \;=\; \mathbf{0}.$$

Putting this together with (4), we obtain:

$$\left[\begin{array}{c|c|c}-\mathbf{I}_q & \multicolumn{2}{c}{\mathbf{B}_{k_i}}\\ \hline \mathbf{0} & -\mathbf{I}_{p-r} & \mathbf{W}'_{k_i}\end{array}\right]\mathbf{z}_i \;=\; \mathbf{0}. \tag{6}$$

Define $\boldsymbol{\Gamma}_{k_i}$ as the $(q+p-r)\times(q+p)$ matrix in the last equation. It follows that $\mathbf{z}_i$ lies in $\ker\boldsymbol{\Gamma}_{k_i}$, which is an r-dimensional subspace in $\mathbb{R}^{q+p}$ (the $q+p-r$ rows in $\boldsymbol{\Gamma}_{k_i}$ are linearly independent because of the identity blocks). We conclude that the columns in $\mathbf{Z}_k$ lie in an r-dimensional subspace of $\mathbb{R}^{q+p}$.

The smaller r the better, because then data lies in lower dimensional subspaces, which are *smaller* and easier to estimate. Also, the smaller r, the more room for missing data, because $r+1$ observations per column are information-theoretically necessary and sufficient for r-dimensional subspace clustering [24].

IV. Experiments

In this section we present an experiment to illustrate our approach on a real-life application. To this end, we will use the yeast dataset studied by Elisseeff and Weston [27]. This dataset contains $n = 2417$ samples. Each sample consists of a vector $\mathbf{x}_i \in \mathbb{R}^p$ containing $p = 113$ activation levels in a gene, and a vector $\mathbf{y}'_i \in \{-1,1\}^q$ indicating whether such gene belongs to each of $q = 14$ functional classes. The goal is to predict each gene's membership in the functional classes, i.e., predict $\mathbf{y}'_i$. Since $\mathbf{y}'_i \in \{-1,1\}^q$, we will model $\mathbf{y}'_i$ as $\operatorname{sign}(\mathbf{y}_i)$, with $\mathbf{y}_i$ as in (1).

To further illustrate the potential of subspace clustering techniques, we will complicate this task by adding missing data. More precisely, let $\mathbf{X} \in \mathbb{R}^{p\times n}$ and $\mathbf{Y}' \in \mathbb{R}^{q\times n}$ be the matrices formed with $\{\mathbf{x}_i\}_{i=1}^n$ and $\{\mathbf{y}'_i\}_{i=1}^n$ as columns. We will only observe a fraction ω of the entries in $\mathbf{X}$ and $\mathbf{Y}'$ (selected uniformly at random), perform mixture regression using subspace clustering, and measure the percentage of unobserved entries in $\mathbf{Y}'$ that were predicted correctly.

Algorithm	$\omega = 40\%$	$\omega = 60\%$	$\omega = 80\%$
GSSC+MC-b	**14.3(0.2)**	**10.9(0.2)**	**7.4(0.4)**
MC-b	16.1(0.3)	12.2(0.3)	8.7(0.4)
MC-1	16.7(0.3)	13.0(0.2)	8.5(0.4)
FPC+SVM	21.5(0.3)	20.8(0.3)	20.3(0.3)
EM1+SVM	22.0(0.2)	21.2(0.2)	20.4(0.2)
Mean+SVM	21.7(0.2)	21.1(0.2)	20.5(0.4)
Zero+SVM	21.6(0.2)	21.1(0.2)	20.5(0.4)

TABLE I: Percentage of incorrectly predicted entries in $\mathbf{Y}$ for different percentages of observed entries in the yeast dataset [27]. Mean (and standard deviation) over 10 trials. Our approach (GSSC+MC-b) outperforms the state-of-the-art method MC-b.

As mentioned in Section I, if we knew the k_i's, we could partition the data accordingly, and do standard regression separately. This is essentially what we did. We first used a state-of-the-art subspace clustering algorithm for missing data, group-sparse subspace clustering (GSSC) [25] to recover the k_i's. This partitioned the data into $\{\mathbf{X}_k\}_{k=1}^K$ and $\{\mathbf{Y}'_k\}_{k=1}^K$. Then for each k, we ran matrix completion with bias (MC-b) [26], a state-of-the-art method for multi-label transductive learning with missing data, which essentially performs standard regression on this type of data. See Section V for more details about these methods.

Table 5 in [26] shows the mean (and standard deviation) performance of several baselines from the literature on this task and this dataset, over 10 independent trials (where the randomness is in the observed entries) for different percentages ω of observed entries. The contesting methods are combinations of completion methods and standard support vector machines (SVM): matrix completion on $\mathbf{X}$ alone using FPC [10] (FPC+SVM), expectation-maximization with k Gaussians to impute $\mathbf{X}$ (EM(k)+SVM), mean-filling (Mean+SVM) and zero-filling (Zero+SVM). Table I complements such table with our results, showing that our approach (GSSC+MC-b) outperforms the state-of-the-art method MC-b as well as all other methods in the comparison.

V. Methods

In this section we briefly describe the two methods that we used in Section IV to perform mixture regression using subspace clustering. The first method is group-sparse subspace clustering (GSSC) [25], a state-of-the-art subspace clustering algorithm for incomplete data, group-sparse subspace clustering (GSSC). The second method is matrix completion with bias (MC-b) [26], a state-of-the-art method for multi-label transductive learning with missing data, which essentially performs standard regression on datasets with both, binary and real valued data, like the yeast dataset [27] analyzed in Section IV.

First let $\mathbf{Z} \in \mathbb{R}^{(q+p)\times n}$ be the matrix formed by stacking $\mathbf{Y}' \in \{-1,1\}^{q\times n}$ and $\mathbf{X} \in \mathbb{R}^{p\times n}$. GSSC will assume that the columns in $\mathbf{Z}$ lie in a union of subspaces, one subspace for each regressor $\mathbf{B}_k$ in (1). By subspace clustering the columns in $\mathbf{Z}$, GSSC aims to recover the labels $\{k_i\}_{i=1}^n$ indicating which columns correspond to which regressors.

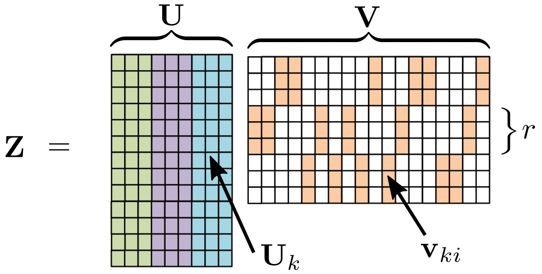

Fig. 1: Factorization aimed by the group-sparse subspace clustering algorithm (GSSC).

The main idea behind GSSC is to find matrices $\mathbf{U} \in \mathbb{R}^{(q+p)\times Kr}$ and $\mathbf{V} \in \mathbb{R}^{Kr\times n}$ such that $\mathbf{UV}$ equals $\mathbf{Z}$ in all the observed entries. Essentially, $\mathbf{U}$ contains K blocks, each of size $(q+p)\times r$. The aim is that the k^{th} block, $\mathbf{U}_k$, gives a basis of the k^{th} subspace. Similarly, each column in $\mathbf{V}$ contains K blocks. The aim is that the k^{th} block in the i^{th} column, $\mathbf{v}_{ki}$, is nonzero if and only if the i^{th} column lies in the k^{th} subspace (see Figure 1 for some intuition). This is pursued through alternating minimization of $\mathbf{U}$ and $\mathbf{V}$, and a group-sparsity penalty on $\mathbf{V}$. GSSC is summarized in Algorithm 1, where $\|\cdot\|_F$ denote the Frobenius norm, $\odot$ denotes the Hadamard (element-wise) product, and $\mathbf{\Omega}$ is the $(q+p)\times n$ matrix indicating the observed entries in $\mathbf{Z}$.

Notice that GSSC requires knowing in advance the number of subspaces K and their dimensions r. Using 5-fold cross validation we obtained $K = 16$ and $r = 19$. We also used 5-fold cross validation to obtain $\lambda = 10^{-3}$. We initialize GSSC with the output of SSC-EWZF (sparse subspace clustering by entry-wise zero fill) [23], which is very similar to the well-known sparse subspace clustering algorithm (SSC) [28], except that the coefficients of each column are obtained using only its observed rows, filling all unobserved entries in these rows of the remaining columns with zeros.

After running GSSC, the estimated labels $\{k_i\}_{i=1}^n$ produce a partition of the data into $\{\mathbf{X}_k\}_{k=1}^K$ and $\{\mathbf{Y}'_k\}_{k=1}^K$. Now for each k we run matrix completion with bias (MC-b) [26], a state-of-the-art method that essentially performs standard regression on this type of data.

The main idea behind MC-b is to find the low-rank matrix that best approximates the observed entries. This is pursued

Algorithm 1: Group-Sparse Subspace Clustering (GSSC)

```
Input: $\mathbf{Z}_{\mathbf{\Omega}}, K, r$, parameter $\lambda$.
  Initialize $\hat{\mathbf{U}} \in \mathbb{R}^{(q+p)\times Kr}$ (e.g., using SSC-EWZF).
  repeat
      $\hat{\mathbf{V}} = \arg\min_{\mathbf{V}} \|\mathbf{\Omega} \odot (\mathbf{Z} - \hat{\mathbf{U}}\mathbf{V})\|_F^2 + \lambda \sum_{k=1,i=1}^{K,n} \|\mathbf{v}_{ki}\|_2$.
      $\hat{\mathbf{U}} = \arg\min_{\mathbf{U}\,:\,\|\mathbf{U}\|_F \le 1} \|\mathbf{\Omega} \odot (\mathbf{Z} - \mathbf{U}\hat{\mathbf{V}})\|_F$.
  until convergence;
Output: $\hat{\mathbf{U}}, \hat{\mathbf{V}}$.
```

Algorithm 2: Matrix Completion with bias (MC-b)

```
Input: $\mathbf{Z}_{\mathbf{\Omega}_k}$, parameters $\lambda, \{\mu_\ell\}_{\ell=1}^{L}$, step-sizes $\tau_{\mathbf{b}}, \tau_{\mathbf{Z}}$.
  Initialize $\hat{\mathbf{Z}}_k$ (e.g., as $\mathbf{\Omega}_k \odot \mathbf{Z}_k$) and $\hat{\mathbf{b}}_k$ (e.g., as $\mathbf{0}$).
  for $\ell = 1, 2, \ldots, L$ do
      repeat
          Compute $\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k - \tau_{\mathbf{b}} \nabla \mathbf{b}$.
          Compute $\hat{\mathbf{Z}}_k = \hat{\mathbf{Z}}_k - \tau_{\mathbf{Z}} \nabla \mathbf{Z}$.
          Compute SVD of $\hat{\mathbf{Z}}_k = \mathbf{LDR}$.
          Compute $\hat{\mathbf{Z}}_k = \mathbf{L} \max(\mathbf{D} - \tau_{\mathbf{Z}} \mu_\ell, \mathbf{0}) \mathbf{R}$.
      until convergence;
Output: $\hat{\mathbf{Z}}_k$.
```

by minimizing

$$\min_{\hat{\mathbf{X}}_k, \hat{\mathbf{Y}}_k, \hat{\mathbf{b}}_k} \mu \left\| \begin{bmatrix} \hat{\mathbf{Y}}_k \\ \hat{\mathbf{X}}_k \end{bmatrix} \right\|_* + \frac{1}{\|\mathbf{\Omega}_{\mathbf{X}_k}\|_0} \sum_{(j,i) \in \mathbf{\Omega}_{\mathbf{X}_k}} (x_{ji} - \hat{x}_{ji})^2$$
$$+ \frac{\lambda}{\|\mathbf{\Omega}_{\mathbf{Y}_k}\|_0} \sum_{(j,i) \in \mathbf{\Omega}_{\mathbf{Y}_k}} \log\left(1 + e^{-y'_{ji}(\hat{y}_{ji} + \hat{b}_{k_j})}\right) \quad (7)$$

where $\|\cdot\|_*$ denotes the nuclear norm, given by the sum of singular values, $\|\cdot\|_0$ denotes the zero-norm, given by the number of nonzero entries, and $\mathbf{\Omega}_{\mathbf{X}_k}$ and $\mathbf{\Omega}_{\mathbf{Y}_k}$ indicate the observed entries in $\mathbf{X}_k$ and $\mathbf{Y}'_k$. The main intuition behind (7) is that the first term will favor low-rank matrices, the quadratic loss in the last term will penalize the error in the real-valued observed entries in $\mathbf{X}_k$, and the logistic loss in the second term will penalize the error in the binary-valued observed entries in $\mathbf{Y}'_k$.

We follow the approach in [26] and use an adaptation of the fixed point continuation (FPC) method in [29] to solve (7). This method essentially consists of gradient steps in $\hat{\mathbf{X}}_k$, $\hat{\mathbf{Y}}_k$ and $\hat{\mathbf{b}}_k$, followed by a shrinkage operator in $\hat{\mathbf{Z}}_k$, the matrix formed by stacking $\hat{\mathbf{Y}}_k$ and $\hat{\mathbf{X}}_k$. MC-b is summarized in Algorithm 2, where the gradient steps are given by

$$\begin{aligned}
\nabla \mathbf{X} &= \frac{1}{|\mathbf{\Omega}_{\mathbf{X}_k}|} \mathbf{\Omega}_{\mathbf{X}_k} \odot (\mathbf{X}_k - \hat{\mathbf{X}}_k), \\
\nabla \mathbf{Y}_{ji} &= \begin{cases} \frac{\lambda}{|\mathbf{\Omega}_{\mathbf{Y}_k}|} \frac{-y'_{ji}}{1+e^{y'_{ji}(\hat{y}_{ji}+\hat{b}_j)}} & \text{if } (j,i) \in \mathbf{\Omega}_{\mathbf{Y}_k} \\ 0 & \text{otherwise,} \end{cases} \\
\nabla \mathbf{b}_{k_j} &= \frac{\lambda}{\|\mathbf{\Omega}_{\mathbf{Y}_k}\|} \sum_{(j,i) \in \mathbf{\Omega}_{\mathbf{Y}_k}} \frac{-y'_{ji}}{1 + e^{y'_{ji}(\hat{y}_{ji}+\hat{b}_j)}}, \\
\nabla \mathbf{Z} &= \begin{bmatrix} \nabla \mathbf{Y} \\ \nabla \mathbf{X} \end{bmatrix}.
\end{aligned}$$

Following the settings in [26] for the yeast dataset [27] studied in Section IV, we set $\{\mu_\ell\}_{\ell=1}^{L}$ starting with $\mu_1 = d_1 \eta$, where d_1 is the largest singular value of $\mathbf{Z}_k \odot \mathbf{\Omega}$ and $\eta = 1/4$, and then decreasing according to the decay parameter η until $\mu_L = 10^{-5}$. We also set $\tau_{\mathbf{Z}} = \min(3.8\|\mathbf{\Omega}_{\mathbf{Y}_k}\|_0/\lambda, \|\mathbf{\Omega}_{\mathbf{X}_k}\|_0)$ and $\tau_{\mathbf{b}} = 3.8\|\mathbf{\Omega}_{\mathbf{Y}_k}\|_0/\lambda n$, with $\lambda = 1$. We claimed convergence whenever there was a change in the objective function (7) smaller than 10^{-5}.

References

[1] A. Dempster, N. Laird and D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. of the royal stat. soc., 1977.

[2] A. Chaganty and P. Liang, *Spectral experts for estimating mixtures of linear regressions*, Int. Conf. on Machine Learning, 2013.

[3] D. Hsu and S. Kakade, *Learning mixtures of spherical gaussians: moment methods and spectral decompositions*, Conference on Innovations in Theoretical Computer Science, ACM, 2013.

[4] X. Yi, C. Caramanis and S. Sanghavi, *Alternating minimization for mixed linear regression*, International Conference on Machine Learning, 2014.

[5] K. Zhong, P. Jain and I. Dhillon, *Mixed linear regression with multiple components*, Advances in neural information processing systems, 2016.

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

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

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

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

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

[11] 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.

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

[13] 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.

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

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

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

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

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

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

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

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

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

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

[24] D. Pimentel-Alarcón and R. Nowak, *The information-theoretic requirements of subspace clustering with missing data*, International Conference on Machine Learning, 2016.

[25] D. Pimentel-Alarcón, L. Balzano, R. Marcia, R. Nowak and R. Willett, *Group-sparse subspace clustering with missing data*, IEEE Statistical Signal Processing, 2016.

[26] A. Goldberg, X. Zhu, B. Recht, J. Xu and R. Nowak, *Transduction with matrix completion: three birds with one stone*, Advances in Neural Information Processing Systems, 2010.

[27] A. Elisseeff and J. Weston, *A kernel method for multi-labelled classification*, Advances in Neural Information Processing Systems, 2001.

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

[29] S. Ma, D. Goldfarb and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Mathematical Programming Series, 2009.