High Rank Matrix Completion with Side Information

Yugang Wang[1] **and Ehsan Elhamifar**[2]
[1] School of Mathematical Sciences, University of Electronic Science and Technology of China
[2] College of Computer and Information Science, Northeastern University

Abstract

We address the problem of high-rank matrix completion with side information. In contrast to existing work dealing with side information, which assume that the data matrix is low-rank, we consider the more general scenario where the columns of the data matrix are drawn from a union of low-dimensional subspaces, which can lead to a high rank matrix. Our goal is to complete the matrix while taking advantage of the side information. To do so, we use the self-expressive property of the data, searching for a sparse representation of each column of matrix as a combination of a few other columns. More specifically, we propose a factorization of the data matrix as the product of side information matrices with an unknown interaction matrix, under which each column of the data matrix can be reconstructed using a sparse combination of other columns. As our proposed optimization, searching for missing entries and sparse coefficients, is non-convex and NP-hard, we propose a lifting framework, where we couple sparse coefficients and missing values and define an equivalent optimization that is amenable to convex relaxation. We also propose a fast implementation of our convex framework using a Linearized Alternating Direction Method. By extensive experiments on both synthetic and real data, and, in particular, by studying the problem of multi-label learning, we demonstrate that our method outperforms existing techniques in both low-rank and high-rank data regimes.

Introduction

Matrix completion, which is the problem of estimating missing entries of an incomplete matrix, is a fundamental task in machine learning with numerous applications, including, collaborative filtering for recommender systems (Rennie and Srebro 2005; Sindhwani et al. 2010), multi-label learning (Natarajan and Dhillon 2014; Xu, Jin, and Zhou 2013; Argyriou, Evgeniou, and Pontil 2008), semi-supervised clustering (Chiang et al. 2014) and global positioning (Singer and Cucuringu 2010; Singer 2008; Biswas et al. 2006). Existing algorithms that deal with missing entries in data can be divided into two main categories. The first group of algorithms, such as Probabilistic PCA (PPCA) (Tipping and Bishop 1999b), Factor Analysis (FA) (Knott and Bartholomew 1999) and Convex Low-Rank Matrix Completion (Candes and Recht 2009; Keshavan, Montanari, and Oh 2010; Chen et al. 2011; Chiang, Hsieh, and Dhillon 2015), assumes that data lie in a single low-dimensional subspace and try to recover a completion of the data that has a minimum or a small fixed rank. The second group of algorithms, including Mixture of Probabilistic PCA (MPPCA) (Tipping and Bishop 1999a; Gruber and Weiss 2004), Mixture of Factor Analyzers (MFA) (Ghahramani, Hinton, and others 1996), K-GROUSE (Balzano et al. 2012), SSC-Lifting (Elhamifar 2016) and (Eriksson, Balzano, and Nowak 2012), addresses the more general and challenging scenario where data lie in a union of low-dimensional subspaces. The goals in this case are to recover missing entries and cluster data according to subspaces. The union of subspaces models many real-world problems, including motion and activity segmentation in videos, recommender systems and multi-label learning, where there exists multiple groups in data corresponding to different classes or categories, where each group is modeled by a single subspace. Since the union of low-dimensional subspaces is often high/full-rank, methods in the first category are not effective for data completion.

Matrix Completion with Side Information. In many real-world problems, we have access to additional information about the entries of the data matrix, referred to as *side information*, which can guide the matrix completion in order to obtain more accurate solutions (Adams, Dahl, and Murray 2010; Agarwal and Chen 2009; Menon et al. 2011a; Porteous, Asuncion, and Welling 2010). For example, in the classical Netflix problem, which aims to predict the unobserved entries of a users-movies rating matrix, besides the rating history, we have access to information/features of users, such as age, gender, etc., as well as information/features of movies, such as suspense, science fiction, etc. Also, in the multi-label learning problem, whose goal is to find all relevant labels to each sample, in addition to the incomplete observed labels, the features describing instances are often given at the same time. Indeed, such side information can be leveraged in matrix completion for better recovery performance, especially, when very few matrix entries are observed.

Despite its significance, the problem of matrix completion with side information has only been recently studied, where all existing techniques have addressed the problem when the data matrix is low-rank (Goldberg et al. 2010; Menon et al. 2011b; Natarajan and Dhillon 2014; Jain and Dhillon 2013;

Xu, Jin, and Zhou 2013; Chiang, Hsieh, and Dhillon 2015; Lu et al. 2016; Chiang, Hsieh, and Dhillon 2016; Liu and Li 2016). The methods in (Menon et al. 2011b; Natarajan and Dhillon 2014) cast the problem as finding a factorization of data matrix as the inner product of side information features with the product of two unknown matrices that must be recovered simultaneously, and employ a non-convex algorithm to recover the unknowns. Although experimental results have shown favorable results, the proposed methods rely on non-convex programming and depend on good initialization. The methods in (Jain and Dhillon 2013; Xu, Jin, and Zhou 2013) use side information feature matrices, $\boldsymbol{F}_c$ and $\boldsymbol{F}_r$, for the rows and columns of the data matrix, $\boldsymbol{Y}$, in a so called Inductive Matrix Completion (IMC) framework. More specifically, assuming that all columns and rows of the data matrix lie in spaces spanned, respectively, by the column vectors in $\boldsymbol{F}_c$ and $\boldsymbol{F}_r$, (Jain and Dhillon 2013; Xu, Jin, and Zhou 2013) consider a factorization of $\boldsymbol{Y}$ as $\boldsymbol{Y} = \boldsymbol{F}_r \boldsymbol{Q} \boldsymbol{F}_c^\top$ for an unknown low-rank inductive matrix $\boldsymbol{Q}$ and try to complete the data by finding $\boldsymbol{Q}$. While (Jain and Dhillon 2013) uses a low-rank matrix factorization, (Xu, Jin, and Zhou 2013) proposes a method, referred to as Maxide, that directly minimizes the rank of $\boldsymbol{Q}$ based on a singular value thresholding algorithm. The work in (Chiang, Hsieh, and Dhillon 2015) considers an extension of the IMC framework, referred to as DirtyIMC, to addresses the problem of matrix completion with noisy side information. More specifically, it considers the model $\boldsymbol{Y} = \boldsymbol{F}_r \boldsymbol{Q} \boldsymbol{F}_c^\top + \boldsymbol{R}$, where the residual matrix $\boldsymbol{R}$ is used to capture the component of the data that the side information cannot describe, and requires both $\boldsymbol{Q}$ and $\boldsymbol{R}$ to be low-rank, hence assuming a low-rank data matrix, $\boldsymbol{Y}$. The work in (Lu et al. 2016) considers a modification of IMC by assuming that the inductive matrix $\boldsymbol{Q}$ is sparse, instead of low-rank, to deal with the situation that $\boldsymbol{Q}$ in not necessarily low-rank. Hence, it minimizes the rank of $\boldsymbol{Y}$ while minimizing the sparsity of $\boldsymbol{Q}$. We refer to this method as Sparse Interactive Model (SIM) in our paper.

It is important to note that all the above work, which address the problem of matrix completion with side information, consider the setting where the data matrix is low-rank. On the other hand, as discussed earlier, in many real-world problems, the data matrix columns or rows lie in a union of low-dimensional subspaces which leads to a high-rank data matrix. As a result, existing techniques will not be effective, as we will demonstrate in our experiments.

Paper Contributions. In this paper, we address the challenging and general problem of high-rank matrix completion with side information. Building on (Elhamifar and Vidal 2013; 2009), we assume that each column of the data matrix can be efficiently represented as a sparse combination of a few other columns, which holds for both a single subspace as well as a union of subspaces. We cast the problem as recovering the missing entries and sparse representation coefficients, while taking advantage of the side information to complete the data matrix. More specifically, we propose a factorization of the data into a product of side information matrices with an unknown interaction matrix, under which each column of the data matrix can be reconstructed using a sparse combination of its other columns. As our proposed formulation is non-convex and NP-hard, building on (Elhamifar 2016), we propose a lifting framework, where we couple sparse coefficients and missing values and define an equivalent optimization that is amenable to convex relaxation. We derive a convex optimization and propose an efficient implementation of our framework using a Linearized Alternating Direction Method (LADM) (Yang and Yuan 2013), which is significantly faster than standard alternating direction methods, hence, allowing to efficiently deal with large data and high percentage of missing entries. Finally, by extensive experiments on synthetic and real data, in particular, by studying the problem of multi-label learning, we demonstrate that our method outperforms existing techniques in both low-rank and high-rank data regimes.

High-Rank Matrix Completion with Side Information

In this section, we propose a method to address the problem of high-rank matrix completion with side information. Assume that we are given a data matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times N}$, which is partially observed, where Ω and Ω^c denote, respectively, the set of indices of observed and missing entries of $\boldsymbol{Y}$. Assume that every row and column of $\boldsymbol{Y}$ is associated with an observed feature vector, providing side information. Let $\boldsymbol{F}_r \in \mathbb{R}^{n \times k_r}$ and $\boldsymbol{F}_c \in \mathbb{R}^{N \times k_c}$ denote side information matrices of feature vectors associated, respectively, with the rows and columns of $\boldsymbol{Y}$. We refer to $\boldsymbol{F}_r$ and $\boldsymbol{F}_c$ as row and column space side information matrices, as they provide additional information about the relationships between entries of the data matrix, which we will use for data completion.

In this paper, we consider a general high/full rank model for $\boldsymbol{Y}$ by assuming that the columns (or similarly rows) of $\boldsymbol{Y}$ lie in a union of low-dimensional subspaces. Our goal is to find missing entries of $\boldsymbol{Y}$, while taking advantage of the side information and respecting the underlying model of the data matrix. While the problem of matrix completion with side information has been studied before (Goldberg et al. 2010; Menon et al. 2011b; Natarajan and Dhillon 2014; Jain and Dhillon 2013; Xu, Jin, and Zhou 2013; Chiang, Hsieh, and Dhillon 2015; Lu et al. 2016; Chiang, Hsieh, and Dhillon 2016; Liu and Li 2016), all existing research has focused on the case where the data matrix is low-rank. On the other hand, in this paper, we study and address the more challenging problem of high-rank matrix completion with side information, which covers the low-rank setting as a special case.

To tackle the problem, similar to all conventional methods (Natarajan and Dhillon 2014; Jain and Dhillon 2013; Xu, Jin, and Zhou 2013; Chiang, Hsieh, and Dhillon 2015; Lu et al. 2016; Chiang, Hsieh, and Dhillon 2016; Liu and Li 2016), we assume that the columns and rows of $\boldsymbol{Y}$ lie in spaces spanned by the columns of $\boldsymbol{F}_r$ and $\boldsymbol{F}_c$, respectively. Thus, we can write $\boldsymbol{Y} = \boldsymbol{F}_r \boldsymbol{Q} \boldsymbol{F}_c^\top$, where $\boldsymbol{Q} \in \mathbb{R}^{k_r \times k_c}$ is a unknown interaction matrix. Let $\bar{\boldsymbol{Y}} \in \mathbb{R}^{n \times N}$ denote the zero-filled data matrix, where the missing entries are filled with zeros. Our goal is to find the complete matrix $\boldsymbol{Y}$, so that it can be decomposed as $\boldsymbol{Y} = \boldsymbol{F}_r \boldsymbol{Q} \boldsymbol{F}_c^\top$ and the entries of $\boldsymbol{Y}$ indexed by Ω coincide with the entries of $\bar{\boldsymbol{Y}}$,

i.e., $R_\Omega(\boldsymbol{Y}) = R_\Omega(\bar{\boldsymbol{Y}})$. The function $R_\Omega(\boldsymbol{Y})$ returns a matrix whose (i,j)-th entry is $\boldsymbol{Y}_{i,j}$ when $(i,j) \in \Omega$ and is 0 otherwise. When the number of observed entries is small, in particular when $|\Omega| < k_r k_c$, without priors on $\boldsymbol{Q}$ and $\boldsymbol{Y}$ the problem has infinitely many solutions for unknown variables. Thus, in addition to utilizing side information, we need to impose appropriate priors on the unknown parameters and use the underlying structure of the data in order to perform data completion.

To take advantage of the fact that the columns of the data matrix $\boldsymbol{Y}$ lie in a union of subspaces, we use the Self-Expressive Model (SEM) (Elhamifar and Vidal 2013; Elhamifar 2016), which states that each column of the data matrix can be written as a sparse representation of the other columns, hence, $\boldsymbol{Y} = \boldsymbol{Y}\boldsymbol{C}$, where $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ denotes the self-representation coefficient matrix. In addition, we need to impose $\mathrm{diag}(\boldsymbol{C}) = \boldsymbol{0}$ to remove the trivial solution of writing each point as a combination of itself. Notice that the SEM covers both a single and a union of low-dimensional subspaces, since in each subspace of dimension d, every point can be written as a combination of only d other points, in general positions, from the same subspace. Thus, to solve the problem of high-rank matrix completion with side information, we propose to solve

$$\begin{aligned} &\min_{\boldsymbol{C},\boldsymbol{Q},\boldsymbol{Y}} \ \frac{1}{2}h(\boldsymbol{Q}) + \lambda\|\boldsymbol{C}\|_0 \\ &\text{s.t.} \ \ \boldsymbol{Y} = \boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top, \ \boldsymbol{Y} = \boldsymbol{Y}\boldsymbol{C}, \ \mathrm{diag}(\boldsymbol{C}) = \boldsymbol{0}, \\ &\qquad R_\Omega(\boldsymbol{Y}) = R_\Omega(\bar{\boldsymbol{Y}}), \end{aligned} \tag{1}$$

where $\|\boldsymbol{C}\|_0$ indicates the ℓ_0-norm of $\boldsymbol{C}$, counting the number of non-zero elements, and $h(\cdot)$ is an appropriate prior on the interaction matrix $\boldsymbol{Q}$, which will be discussed shortly. Notice that the optimization in (1) is NP-hard and non-convex, due to the product of unknown matrices $\boldsymbol{Y}$ and $\boldsymbol{C}$ as well as the sparsity regularization on $\boldsymbol{C}$. To tackle the problem, building on (Elhamifar 2016), we propose a lifting scheme, where we define new optimization variables, corresponding to the product of unknown coefficients and the missing values, and pose the problem as an optimization on the new variables that, together with appropriate constraints, will lead to an equivalent optimization to (1).

Lifting-Based Formulation

In this part, we develop an equivalent optimization for our formulation in (1) and propose an efficient convex relaxation. We denote by $\boldsymbol{y}_i, \bar{\boldsymbol{y}}_i \in \mathbb{R}^n$ the i-th column of $\boldsymbol{Y}$ and $\bar{\boldsymbol{Y}}$, respectively. We also denote by $\Omega_i, \Omega_i^c \subseteq \{1,\dots,n\}$ the set of, respectively, observed and missing entries of the i-th column of $\boldsymbol{Y}$. Denoting by $\boldsymbol{U}_{\Omega_i^c} \in \mathbb{R}^{n\times|\Omega_i^c|}$ a matrix formed by taking the columns of the identity matrix indexed by Ω_j^c, we can write

$$\boldsymbol{y}_i = \bar{\boldsymbol{y}}_i + \boldsymbol{U}_{\Omega_i^c}\boldsymbol{x}_i = \begin{bmatrix}\bar{\boldsymbol{y}}_i & \boldsymbol{U}_{\Omega_i^c}\end{bmatrix}\begin{bmatrix}1\\ \boldsymbol{x}_i\end{bmatrix}, \tag{2}$$

where $\boldsymbol{x}_i \in \mathbb{R}^{|\Omega_i^c|}$ denotes the vector of missing values of the i-th column of the data matrix. Denoting the element (i,j) of $\boldsymbol{C}$ by $c_{i,j}$, using equation (2), we can rewrite the self-expressiveness model, $\boldsymbol{y}_j = \sum_{i=1}^N c_{i,j}\boldsymbol{y}_i$, as

$$\boldsymbol{y}_j = \sum_{i=1}^N c_{i,j}\begin{bmatrix}\bar{\boldsymbol{y}}_i & \boldsymbol{U}_{\Omega_i^c}\end{bmatrix}\begin{bmatrix}1\\ \boldsymbol{x}_i\end{bmatrix} = \sum_{i=1}^N \begin{bmatrix}\bar{\boldsymbol{y}}_i & \boldsymbol{U}_{\Omega_i^c}\end{bmatrix}\begin{bmatrix}c_{i,j}\\ c_{i,j}\boldsymbol{x}_i\end{bmatrix}. \tag{3}$$

Given the fact that $\boldsymbol{y}_j$ is the j-th column of $\boldsymbol{Y}$, using the above, we can write

$$\boldsymbol{Y} = \sum_{i=1}^N \begin{bmatrix}\bar{\boldsymbol{y}}_i & \boldsymbol{U}_{\Omega_i^c}\end{bmatrix}\begin{bmatrix}c_{i,1} & \cdots & c_{i,N}\\ c_{i,1}\boldsymbol{x}_i & \cdots & c_{i,N}\boldsymbol{x}_i\end{bmatrix} = \boldsymbol{D}\boldsymbol{A}, \tag{4}$$

where the dictionary $\boldsymbol{D}$ is a given and known matrix and is defined, using the zero-filled data and the columns of the identity matrix indexed by missing entries of points, as

$$\boldsymbol{D} = \begin{bmatrix}\bar{\boldsymbol{y}}_1 & \boldsymbol{U}_{\Omega_1^c} \mid \cdots \mid \bar{\boldsymbol{y}}_N & \boldsymbol{U}_{\Omega_N^c}\end{bmatrix}, \tag{5}$$

while the matrix $\boldsymbol{A}$, which consists of missing entries and self-representation coefficients, is unknown and defined as

$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{A}_1^\top & \cdots & \boldsymbol{A}_N^\top\end{bmatrix}^\top, \ \boldsymbol{A}_i = \begin{bmatrix}c_{i,1} & \cdots & c_{i,N}\\ c_{i,1}\boldsymbol{x}_i & \cdots & c_{i,N}\boldsymbol{x}_i\end{bmatrix}. \tag{6}$$

It is important to notice that each $\boldsymbol{A}_i$ is a rank-1 matrix, since we can write it is as the outer product of vectors $\begin{bmatrix}1 & \boldsymbol{x}_i^\top\end{bmatrix}^\top$ and $[c_{i,1} \ \cdots \ c_{i,N}]$. This helps us to pave the way for an equivalent optimization to (1) that is amenable to convex relaxation. More specifically, if we define each $\boldsymbol{A}_i$ as

$$\boldsymbol{A}_i = \begin{bmatrix}c_{i,1} & \cdots & c_{i,N}\\ \boldsymbol{\alpha}_{i,1} & \cdots & \boldsymbol{\alpha}_{i,N}\end{bmatrix}, \tag{7}$$

and impose that rank of $\boldsymbol{A}_i$ be one, we obtain $\boldsymbol{\alpha}_{i,j} = c_{i,j}\boldsymbol{x}_i$. As a result, we can write our proposed optimization in (1) as the equivalent optimization program

$$\begin{aligned} &\min_{\boldsymbol{A},\boldsymbol{Q},\boldsymbol{Y}} \ \frac{1}{2}h(\boldsymbol{Q}) + \lambda\|\boldsymbol{A}^{(c)}\|_0 \\ &\text{s.t.} \ \ \boldsymbol{Y} = \boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top, \ \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{A}, \ \mathrm{diag}(\boldsymbol{A}^{(c)}) = \boldsymbol{0}, \\ &\qquad \mathrm{rk}(\boldsymbol{A}_i) = 1, \ \forall i, \ \ R_\Omega(\boldsymbol{Y}) = R_\Omega(\bar{\boldsymbol{Y}}). \end{aligned} \tag{8}$$

Here, $\boldsymbol{A}^{(c)}$ denotes a submatrix of $\boldsymbol{A}$, whose i-th row corresponds to the first row of $\boldsymbol{A}_i$, i.e., $\boldsymbol{A}^{(c)} = \boldsymbol{C}$. Notice that we have transferred the non-convexity of the product of unknown coefficients and missing entries in (1) to a set of rank-1 constraints on the new optimization variables. As we show next, our new formulation in (8) allows to explore efficient methods based on convex relaxation to solve the problem.

Convex Relaxation and Extensions

To obtain a convex optimization, first we use the convex surrogate of the ℓ_0-norm, and minimize the ℓ_1-norm $\|\boldsymbol{A}^{(c)}\|_1$. In the paper, we select the prior on $\boldsymbol{Q}$ to be $h(\boldsymbol{Q}) = 1/2\|\boldsymbol{Q}\|_F^2$. Also, we use the nuclear norm relaxation of the rank constraints, i.e., we use $\|\boldsymbol{A}_i\|_* \le \eta$, for $\eta > 0$, where the nuclear norm $\|\boldsymbol{A}_i\|_*$ is defined as the sum of the singular values of $\boldsymbol{A}_i$. To reduce the number of constraints, we bring the nuclear norm constraints to the objective function via a regularization parameter $\rho > 0$ and propose to solve

$$\begin{aligned} &\min_{\boldsymbol{A},\boldsymbol{Q},\boldsymbol{Y}} \ \frac{1}{2}\|\boldsymbol{Q}\|_F^2 + \lambda\|\boldsymbol{A}^{(c)}\|_1 + \rho\sum_{i=1}^N \|\boldsymbol{A}_i\|_* \\ &\text{s.t.} \ \ \boldsymbol{Y} = \boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top, \ \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{A}, \ \mathrm{diag}(\boldsymbol{A}^{(c)}) = \boldsymbol{0}, \\ &\qquad R_\Omega(\boldsymbol{Y}) = R_\Omega(\bar{\boldsymbol{Y}}). \end{aligned} \tag{9}$$

Notice that the above optimization is convex, which can be solved efficiently using convex programming techniques.

In real-world problems, however, observed data entries are corrupted by noise (Soltanolkotabi, Elhamifar, and Candes 2014). In other words, we have $R_\Omega(\bar{\boldsymbol{Y}}) = R_\Omega(\boldsymbol{Y}+\boldsymbol{E}_1)$, for an error matrix $\boldsymbol{E}_1$. Moreover, since error-free data may not necessarily lie perfectly on subspaces, we should account for deviations from the SEM model, i.e., $\bar{\boldsymbol{Y}} = \boldsymbol{Y}+\boldsymbol{E}_2$, for an error matrix $\boldsymbol{E}_2$. We assume that the energy of the noise terms are bounded, i.e., $\|\boldsymbol{E}_1\|_F^2 \leq \zeta_1$ and $\|\boldsymbol{E}_2\|_F^2 \leq \zeta_2$ for $\zeta_1 > 0$ and $\zeta_2 > 0$. Thus, using the Lagrange multipliers, we propose to solve

$$\begin{aligned}\min_{\boldsymbol{A},\boldsymbol{Q},\boldsymbol{Y}} \ & \frac{1}{2}\|\boldsymbol{Q}\|_F^2 + \lambda\|\boldsymbol{A}^{(c)}\|_1 + \rho\sum_{i=1}^{N}\|\boldsymbol{A}_i\|_* \\ & + \frac{\mu_1}{2}\|R_\Omega(\boldsymbol{Y}-\bar{\boldsymbol{Y}})\|_F^2 + \frac{\mu_2}{2}\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{A}\|_F^2 \\ \text{s.t.} \ & \boldsymbol{Y} = \boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top, \ \operatorname{diag}(\boldsymbol{A}^{(c)}) = \boldsymbol{0}\end{aligned} \tag{10}$$

where $\mu_1, \mu_2 > 0$ are positive regularization parameters. Next, we develop an efficient algorithm to solve our proposed optimization in (10).

Linearized ADM Algorithm

In this section, we develop a Linearized Alternating Direction Method (LADM) (Yang and Yuan 2013) to efficiently solve (10). To do so, we introduce two auxiliary matrices $\boldsymbol{Z}$ and $\boldsymbol{V}$, which we enforce to be equal to $\boldsymbol{A}$, and define the notations of submatrices $\boldsymbol{V}_i$ and $\boldsymbol{Z}^{(c)}$ similar to (7) and (8), respectively. We consider the following optimization,

$$\begin{aligned}\min_{\boldsymbol{A},\boldsymbol{Z},\boldsymbol{V},\boldsymbol{Q},\boldsymbol{Y}} \ & \frac{1}{2}\|\boldsymbol{Q}\|_F^2 + \lambda\|\boldsymbol{Z}^{(c)}\|_1 + \rho\sum_{i=1}^{n}\|\boldsymbol{V}_i\|_* \\ & + \frac{\mu_1}{2}\|R_\Omega(\boldsymbol{Y}-\bar{\boldsymbol{Y}})\|_F^2 + \frac{\mu_2}{2}\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{A}\|_F^2 \\ \text{s.t.} \ & \boldsymbol{Y} = \boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top, \operatorname{diag}(\boldsymbol{Z}^{(c)}) = \boldsymbol{0}, \ \boldsymbol{A} = \boldsymbol{Z}, \ \boldsymbol{A} = \boldsymbol{V}.\end{aligned} \tag{11}$$

which is equivalent to (10). To solve the above optimization via the LADM approach, we first form the augment Lagrangian function of (11), which is

$$\begin{aligned}\boldsymbol{L}(\boldsymbol{Q},\boldsymbol{Y},&\boldsymbol{A},\boldsymbol{Z},\boldsymbol{V}) = \\ & + \frac{1}{2}\|\boldsymbol{Q}\|_F^2 + \lambda\|\boldsymbol{Z}^{(c)}\|_1 + \rho\sum_{i=1}^{n}||\boldsymbol{V}_i||_* \\ & + \frac{\mu_1}{2}\|R_\Omega(\boldsymbol{Y}-\bar{\boldsymbol{Y}})\|_F^2 + \frac{\mu_2}{2}\|\boldsymbol{Y}-\boldsymbol{D}\boldsymbol{A}\|_F^2 \\ & + \langle\boldsymbol{\Gamma}_1, \boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top - \boldsymbol{Y}\rangle + \langle\boldsymbol{\Gamma}_2, \boldsymbol{Z}-\boldsymbol{A}\rangle \\ & + \langle\boldsymbol{\Gamma}_3, \boldsymbol{V}-\boldsymbol{A}\rangle + \frac{\beta}{2}\|\boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top - \boldsymbol{Y}\|_F^2 \\ & + \frac{\beta}{2}\|\boldsymbol{Z}-\boldsymbol{A}\|_F^2 + \frac{\beta}{2}\|\boldsymbol{V}-\boldsymbol{A}\|_F^2,\end{aligned} \tag{12}$$

where $\boldsymbol{\Gamma}_1$, $\boldsymbol{\Gamma}_2$ and $\boldsymbol{\Gamma}_3$ are Lagrange multiplier matrices associated with the constraints of (11), and $\beta > 0$ is the augmented Lagrangian penalty parameter.

In an ADM framework, we take an iterative approach, where starting from initialization of optimization matrices, we fix all unknown matrices except one, solve for the unknown matrix, and repeat this procedure for every optimization matrix. It is straightforward to verify that the updates of $\boldsymbol{A}$ and $\boldsymbol{Q}$ depends on the updates of $\boldsymbol{Y}$, $\boldsymbol{Z}$ and $\boldsymbol{V}$ and vice versa. To derive the updates, we perform the following.

– At iteration $k+1$, minimizing $\boldsymbol{L}$ with respect to $\boldsymbol{A}$, while fixing other matrices, leads to

$$\begin{aligned}\boldsymbol{A}^{k+1} = (\mu_2\boldsymbol{D}^\top\boldsymbol{D} + 2\beta\boldsymbol{I})^{-1}(\mu_2\boldsymbol{D}^\top\boldsymbol{Y}^k \\ +\beta\boldsymbol{Z}^k + \beta\boldsymbol{V}^k + \boldsymbol{\Gamma}_2^k + \boldsymbol{\Gamma}_3^k),\end{aligned} \tag{13}$$

where $\boldsymbol{I}$ is identity matrix. Indeed, as long as the number of points and the percentage of missing entries is small, we use the above formulation. However, when the number of points and/or the percentage of missing entries increases, the size of $\boldsymbol{D}^\top\boldsymbol{D}$ will be large and computation of the inverse, which is cubic in the size of the matrix, will be computationally prohibitive. Thus, we propose to solve $\boldsymbol{A}$ using the Linearized ADM technique, which also enjoys convergence to optimal solution (Yang and Yuan 2013). To do so, we approximate the term involving $\boldsymbol{A}$ with its linearized form as

$$\frac{1}{2}\|\boldsymbol{D}\boldsymbol{A}-\boldsymbol{Y}^k\|_F^2 \approx \frac{1}{2\tau_1}\|\boldsymbol{A}-(\boldsymbol{A}^k-\tau_1\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{A}^k-\boldsymbol{Y}^k)\|_F^2,$$

where $\tau_1 > 0$ is the proximal parameter. Hence, we can compute the solution of $\boldsymbol{A}$ as

$$\begin{aligned}\boldsymbol{A}^{k+1} = \frac{\tau_1}{2\beta\tau_1+\mu_2}\Big(\frac{\mu_2}{\tau_1}\boldsymbol{A}^k - \mu_2\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{A}^k-\boldsymbol{Y}^k) \\ + \beta(\boldsymbol{Z}^k+\boldsymbol{V}^k) + \boldsymbol{\Gamma}_2^k + \boldsymbol{\Gamma}_3^k\Big).\end{aligned} \tag{14}$$

– To solve for $\boldsymbol{Q}$, while fixing other unknowns, we need to minimize the Lagrangian with respect to the terms involving $\boldsymbol{Q}$, i.e., need to solve the following subproblem

$$\min_{\boldsymbol{Q}} \frac{1}{2}\|\boldsymbol{Q}\|_F^2 + \frac{\beta}{2}\|\boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top - \boldsymbol{Y}^k + \frac{\boldsymbol{\Gamma}_1^k}{\beta}\|_F^2. \tag{15}$$

While we can solve for $\boldsymbol{Q}$ in closed-form, the solution requires to compute the inverse of $(\boldsymbol{F}_r\otimes\boldsymbol{F}_c)^\top(\boldsymbol{F}_r\otimes\boldsymbol{F}_c)$, where $\otimes$ denotes the the kronecker product. Since this computation is expensive, we use a linearization of (15) and solve the following subproblem

$$\min_{\boldsymbol{Q}} \frac{1}{2}\|\boldsymbol{Q}\|_F^2 + \frac{\beta}{2\tau_2}\|\boldsymbol{Q}-(\boldsymbol{Q}^k-\tau_2\boldsymbol{g}^k)\|_F^2, \tag{16}$$

where $\tau_2 > 0$ is a proximal parameter, and $\boldsymbol{g}^k$ is the gradient of $\frac{1}{2}\|\boldsymbol{F}_r\boldsymbol{Q}\boldsymbol{F}_c^\top - \boldsymbol{Y}^k + \frac{\boldsymbol{\Gamma}_1^k}{\beta}\|_F^2$ at $\boldsymbol{Q}^k$, which is given by

$$\boldsymbol{g}^k = \boldsymbol{F}_r^\top\boldsymbol{F}_r\boldsymbol{Q}^k\boldsymbol{F}_c^\top\boldsymbol{F}_c - \boldsymbol{F}_r^\top(\boldsymbol{Y}^k - \frac{\boldsymbol{\Gamma}_1^k}{\beta})\boldsymbol{F}_c. \tag{17}$$

As a result, solving (16), we obtain the following closed-form solution for $\boldsymbol{Q}^{k+1}$,

$$\boldsymbol{Q}^{k+1} = \frac{\tau_2}{\beta+\tau_2}(\boldsymbol{Q}^k - \tau_2\boldsymbol{g}^k). \tag{18}$$

Algorithm 1

Input: Incomplete data matrix $\bar{\boldsymbol{Y}}$, side information matrices $\boldsymbol{F}_r$, $\boldsymbol{F}_c$ and set of indices of observed entries Ω.
Output: $\boldsymbol{Y}, \boldsymbol{A}, \boldsymbol{Q}$.

1: Set $k = 0$. Initialize matrices $\boldsymbol{Q}^0$, $\boldsymbol{Y}^0$, $\boldsymbol{A}^0$, $\boldsymbol{Z}^0$, $\boldsymbol{V}^0$, $\boldsymbol{\Gamma}_1^0$, $\boldsymbol{\Gamma}_2^0, \boldsymbol{\Gamma}_3^0$ as zero matrices.
2: Update $\boldsymbol{A}^{k+1}$ using the equation (13) or (14).
3: Update $\boldsymbol{Q}^{k+1}$ using the equations (18).
4: Update $\boldsymbol{Y}^{k+1}$ using the equation (19).
5: Update $\boldsymbol{Z}^{k+1}$ using (20) and (21).
6: Update $\boldsymbol{V}^{k+1}$ using (20) and (21).
7: Update the multipliers $\boldsymbol{\Gamma}_1^{k+1}, \boldsymbol{\Gamma}_2^{k+1}, \boldsymbol{\Gamma}_3^{k+1}$ using (22).
8: Set $k = k + 1$. While not converged, go to step 2.
9: **return** $\boldsymbol{Y}, \boldsymbol{A}, \boldsymbol{Q}$.

– To solve for $\boldsymbol{Y}$, we minimize the Lagrangian with respect to $\boldsymbol{Y}$ while fixing other optimization matrices. The solution of the corresponding least square problem leads to

$$\boldsymbol{Y}_{ij}^{k+1} = \begin{cases} \left[\frac{\mu_1 \boldsymbol{Y}^k + \mu_2 \boldsymbol{D}\boldsymbol{A}^k + \beta \boldsymbol{F}_r \boldsymbol{Q}^k \boldsymbol{F}_c^\top + \boldsymbol{\Gamma}_1^k}{\mu_1 + \mu_2 + \beta}\right]_{ij}, & (i,j) \in \Omega, \\ \left[\frac{\mu_2 \boldsymbol{D}\boldsymbol{A}^k + \beta \boldsymbol{F}_r \boldsymbol{Q}^k \boldsymbol{F}_c^\top + \boldsymbol{\Gamma}_1^k}{\mu_2 + \beta}\right]_{ij}, & (i,j) \notin \Omega. \end{cases} \tag{19}$$

– At iteration $k+1$, minimizing the augmented Lagrangian with respect to $\boldsymbol{Z}$, while fixing other matrices, leads to the soft thresholding operation,

$$\boldsymbol{Z}^{(c),k+1} = \max\left(\left|\boldsymbol{A}^k - \frac{\boldsymbol{\Gamma}_2^k}{\beta}\right| - \frac{\lambda}{\beta}, 0\right) \odot \operatorname{sign}(\boldsymbol{A}^k - \frac{\boldsymbol{\Gamma}_2^k}{\beta}), \tag{20}$$

for the entires of $\boldsymbol{Z}^{(c)}$, where $\odot$ denotes the Hadamard product. On the other hand, the solution for other entries of $\boldsymbol{Z}$ is given by

$$\boldsymbol{Z}^{k+1} = \boldsymbol{A}^k - \frac{\boldsymbol{\Gamma}_2^k}{\beta} \tag{21}$$

– At iteration $k+1$, minimizing the augmented Lagrangian with respect to $\boldsymbol{V}$, while fixing other matrices, leads to

$$\boldsymbol{V}_i^{k+1} = \operatorname{svt}(\boldsymbol{A}_i^k - \frac{[\boldsymbol{\Gamma}_3^k]_i}{\beta}, \frac{\rho}{\beta}), \ \forall i, \tag{22}$$

where $\operatorname{svt}(\boldsymbol{M}, t)$ is the singular value thresholding operation (Cai, Candès, and Shen 2010).

– Finally we update the Lagrange multiplier matrices $\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \boldsymbol{\Gamma}_3$ by a coordinate ascent method,

$$\begin{aligned} \boldsymbol{\Gamma}_1^{k+1} &= \boldsymbol{\Gamma}_1^k + \beta\left(\boldsymbol{F}_r \boldsymbol{Q}^{k+1} \boldsymbol{F}_c^\top - \boldsymbol{Y}^{k+1}\right), \\ \boldsymbol{\Gamma}_2^{k+1} &= \boldsymbol{\Gamma}_2^k + \beta\left(\boldsymbol{Z}^{k+1} - \boldsymbol{A}^{k+1}\right), \\ \boldsymbol{\Gamma}_3^{k+1} &= \boldsymbol{\Gamma}_3^k + \beta\left(\boldsymbol{V}^{k+1} - \boldsymbol{A}^{k+1}\right). \end{aligned} \tag{23}$$

Algorithm 1 shows the steps of our framework.

Experiments

In this section, we evaluate the performance of our proposed algorithm for high-rank matrix completion against the existing methods on both synthetic and real data and in both low-rank and high-rank settings. We compare our method with Maxide (Xu, Jin, and Zhou 2013), IMC (Natarajan and Dhillon 2014), dirtyIMC (Chiang, Hsieh, and Dhillon 2015) and SIM (Lu et al. 2016).[1] All the methods address the problem of low-rank matrix completion with side information. In contrast, our framework addresses the more challenging and general problem of high-rank matrix completion with side information. Given a ground-truth data matrix, $\boldsymbol{Y}^*$, we drop the values of δ fraction of the entries uniformly at random and change δ from 0.1 to 0.9. Given the recovered completed data, $\boldsymbol{Y}$, we compute the relative matrix completion error as

$$\text{RE} = \|\boldsymbol{Y} - \boldsymbol{Y}^*\|_F / \|\boldsymbol{Y}^*\|_F. \tag{24}$$

In the experiments, for our proposed method, we set $\lambda = 10$, $\rho = 2 \times 10^2$, $\mu_1 = \mu_2 = 10^5$, $\beta = 10^2$, $\tau_1 = 10^{-5}$, $\tau_2 = 10^{-2}$. Experimentally, we observed robust performance with respect to the change of these parameters.

Synthetic Experiments

We first evaluate different methods on synthetic low-rank and high-rank data. We let $k_r = k_c = k$ and generate $\boldsymbol{F}_c^\top$ so that the columns lie in a union of L low-dimensional subspaces, where the dimension of each subspace is d. To do so, we generate L random d-dimensional bases in $\mathbb{R}^k$ and generate N_g random data points in each subspace, forming matrices $\{\boldsymbol{B}_i \in \mathbb{R}^{k \times N_g}\}_{i=1}^L$. We then form $\boldsymbol{F}_c^\top = [\boldsymbol{B}_1 \ \cdots \ \boldsymbol{B}_L] \in \mathbb{R}^{k \times LN_g}$. We also draw $\boldsymbol{F}_r \in \mathbb{R}^{n \times k}$ and $\boldsymbol{Q} \in \mathbb{R}^{k \times k}$ from a standard Normal distribution. We set $\boldsymbol{Y}^* = 100 \times \boldsymbol{F}_r \boldsymbol{Q} \boldsymbol{F}_c^\top + \boldsymbol{E}$, where the entries of the noise matrix $\boldsymbol{E}$ are drawn from the standard normal distribution. Since $\boldsymbol{F}_c^\top$ has a union of low-rank structure, the columns of $\boldsymbol{Y}^*$ also lie in a union of low-rank subspaces. We set $n = 100$ for all the synthetic experiments. For the low-rank regime, we set $L = 3, d = 4, k = 12, Ng = 30$, hence, $L \times d = 12 \ll n = 100$. For high-rank regime, we set $L = 10, d = 10, r = 100, Ng = 50$, hence, $L \times d = n = 100$, i.e., a full-rank data matrix.

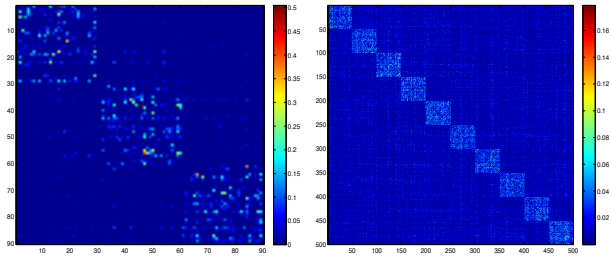
Table 1 shows the average relative matrix completion errors of different methods over 10 random trials. Notice that in both low-rank and high-rank regimes, our method outperforms existing methods across all fractions of missing entries δ. More specifically, the errors of IMC and SIM are higher than other methods as they rely on the low-rank assumption on the data matrix. Our proposed method and Maxide obtain smaller errors for all δ, while our method achieves better performance, due to being able to deal with the more general setting of the high-rank data. The performance of DirtyIMC significantly degrades when $\delta \geq 0.7$ for the low-rank and $\delta \geq 0.2$ for the high-rank regime. Notice also that the results of all methods degrade in the high-rank case compared to the low-rank regime. However, the performance of our method is less affected, thanks to its ability to handle high-rank data matrices.

Figure 1 shows the recovered sparse coefficient matrices $\boldsymbol{C}$, recovered by our method, for synthetic low-rank (with three subspaces) and high-rank (with ten subspaces)

[1] For Maxide, IMC and SIM, we used the publically available codes. We implemented dirtyIMC, since the code was not available.

Table 1: Relative completion errors for low-rank and high-rank data matrices.

	δ	SIM	dirtyIMC	IMC	Maxide	Proposed
Low-Rank	0.1	4.01×10^{-2}	2.39×10^{-4}	1.09×10^{-1}	3.23×10^{-4}	$\mathbf{6.16 \times 10^{-5}}$
	0.2	6.69×10^{-2}	2.61×10^{-4}	2.07×10^{-1}	6.57×10^{-4}	$\mathbf{6.47 \times 10^{-5}}$
	0.3	9.76×10^{-2}	2.94×10^{-4}	3.08×10^{-1}	2.54×10^{-4}	$\mathbf{7.12 \times 10^{-5}}$
	0.4	1.33×10^{-1}	3.38×10^{-4}	4.03×10^{-1}	2.86×10^{-4}	$\mathbf{7.63 \times 10^{-5}}$
	0.5	1.88×10^{-1}	4.29×10^{-4}	5.06×10^{-1}	4.83×10^{-4}	$\mathbf{8.34 \times 10^{-5}}$
	0.6	2.80×10^{-1}	7.17×10^{-4}	6.03×10^{-1}	1.15×10^{-4}	$\mathbf{9.31 \times 10^{-5}}$
	0.7	5.32×10^{-1}	1.87×10^{-2}	7.03×10^{-1}	7.58×10^{-4}	$\mathbf{1.08 \times 10^{-4}}$
	0.8	1.17×10^{0}	5.23×10^{-1}	8.04×10^{-1}	3.32×10^{-4}	$\mathbf{1.36 \times 10^{-4}}$
	0.9	2.64×10^{0}	5.55×10^{-1}	9.03×10^{-1}	7.45×10^{-4}	$\mathbf{2.23 \times 10^{-4}}$
High-Rank	0.1	1.21×10^{-1}	8.25×10^{-2}	1.75×10^{-1}	8.47×10^{-3}	$\mathbf{1.32 \times 10^{-4}}$
	0.2	2.09×10^{-1}	1.43×10^{-1}	2.75×10^{-1}	7.72×10^{-3}	$\mathbf{2.47 \times 10^{-4}}$
	0.3	3.18×10^{-1}	2.14×10^{-1}	3.73×10^{-1}	8.22×10^{-3}	$\mathbf{4.01 \times 10^{-4}}$
	0.4	4.25×10^{-1}	3.02×10^{-1}	4.64×10^{-1}	9.60×10^{-3}	$\mathbf{6.35 \times 10^{-4}}$
	0.5	5.31×10^{-1}	4.12×10^{-1}	5.55×10^{-1}	1.24×10^{-2}	$\mathbf{8.48 \times 10^{-4}}$
	0.6	6.34×10^{-1}	5.49×10^{-1}	6.44×10^{-1}	1.69×10^{-2}	$\mathbf{1.31 \times 10^{-3}}$
	0.7	7.39×10^{-1}	7.12×10^{-1}	7.34×10^{-1}	4.14×10^{-2}	$\mathbf{3.92 \times 10^{-3}}$
	0.8	8.42×10^{-1}	8.77×10^{-1}	8.24×10^{-1}	1.90×10^{-1}	$\mathbf{1.58 \times 10^{-2}}$
	0.9	9.39×10^{-1}	1.00×10^{0}	9.12×10^{-1}	5.12×10^{-1}	$\mathbf{3.67 \times 10^{-2}}$



(a) (b)

Figure 1: The coefficient matrix $\boldsymbol{C}$ obtained by our method for (a) low-rank and (b) high-rank matrices, where $\delta = 0.1$.

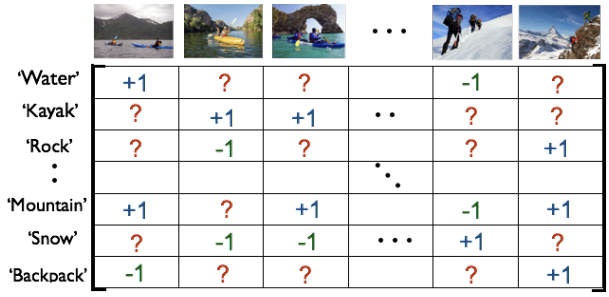


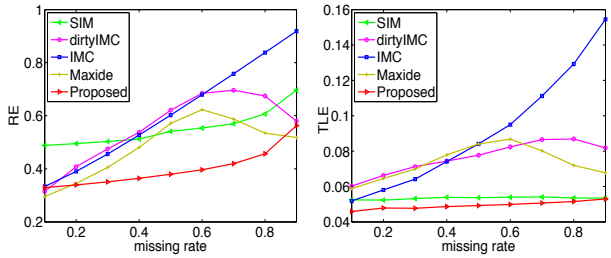
Figure 2: In multi-label learning, we are given an incomplete matrix of label-samples, whose entries indicate the presence (+1) or absence (-1) of each label in each sample.

datasets, where points are ordered according to their memberships to subspaces. Notice that in both cases, our method successfully recovers a representation of each point as a sparse combination of points from its own subspace.

Real Experiments: Multi-Label Learning

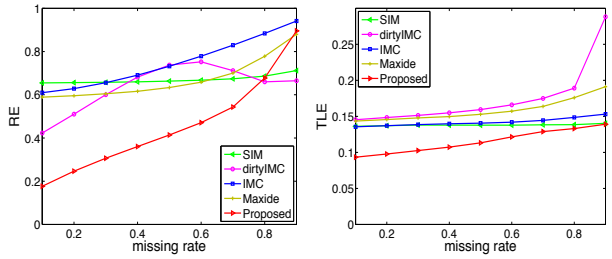
In this section, we evaluate our method on real data for the problem of multi-label learning, where the goal is to predict the unobserved labels in an instance-label matrix. More specifically, in multi-label learning, each sample/instance could belong to multiple classes, in which case the labels of the classes are assigned +1, while the labels of other classes to which the sample does not belong will be -1. Given an incomplete matrix of samples and labels, whose entries are ± 1, our goal is to complete the matrix using additional information about the feature representation of samples, hence, predict the values of missing labels, see Figure 2. Notice that while each sample is often correlated with a few other samples, hence a sparse combination is valid, it is unrealistic to assume that all samples are highly correlated and lie on the same low-rank model. To evaluate our method, we consider three datasets, consisting of audio, music and image data.

Multi-label learning for audio classification. The Birds dataset consists of 645 ten-second audio files recording the sounds of 19 different species of birds as well as the sounds of environments, such as wind or rain (Briggs et al. 2013). Each recording is labeled by a 19-dimensional vector corresponding to 19 species of birds with the entries being 1 if the recording contains a particular class of birds and -1 otherwise. Each recording is associated with a 258-dimensional feature vector extracted from its spectrogram, corresponding to the side information. In our experiments, the ground-truth label matrix is $\boldsymbol{Y} \in \mathbb{R}^{19 \times 645}$, corresponding to 19 labels of 645 samples. We set the feature matrix $\boldsymbol{F}_c \in \mathbb{R}^{645 \times 258}$ using the spectrogram information of each of the 645 recordings. We also set $\boldsymbol{F}_r$ as the 19×19 identity matrix, since we only have access to the feature matrix for the audio files, not samples. Analysis of the singular values of $\boldsymbol{Y}$ shows the data matrix being high-rank, where the smallest singular value is



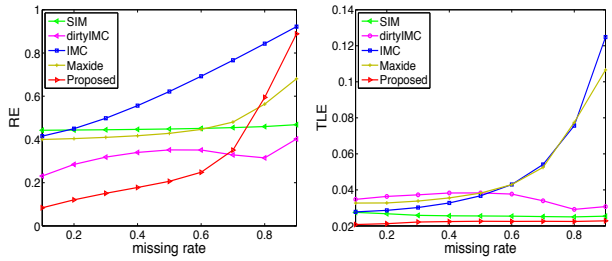
(a) (b)

Figure 3: (a) Relative matrix completion error, (b) Transductive label error, for the Birds dataset.



(a) (b)

Figure 4: (a) Relative matrix completion error, (b) Transductive label error, for the CAL500 dataset.



(a) (b)

Figure 5: (a) Relative matrix completion error, (b) Transductive label error, for the NUS-WIDE dataset.

4.9. We randomly select δ fraction of labels of $\boldsymbol{Y}$, as missing entries, and set them to zeros to generate $\bar{\boldsymbol{Y}}$. We then recover the completed data matrix using different methods and record both the relative matrix completion error (RE) and the transductive label error (TLE). To compute TLE, we take the sign of the entries of the recovered data matrices and then compute the percentage of unobserved labels that are misclassified (Goldberg et al. 2010). We report the average of the two errors over 10 random trials. As the results in Figure 3 illustrates, our proposed method outperforms Maxide, IMC, dirtyIMC and SIM on both types of errors. Notice that the relative completion error (RE) of all algorithms is higher than the transductive label error (TLE). This comes from the fact that while the completed entries might be far from the true values, their sign is compatible with the true label.

Multi-label learning for music classification. The CAL500 dataset is a collection of 502 "Western popular" songs from 502 unique artists, where each song is has multiple labels from 174 "musically relevant" semantic keywords, corresponding to different categories, such as genre, emotional content and instrumentation (Turnbull et al. 2008). In addition, each song is represented by a 68-dimensional acoustic feature vector extracted from the audio files. In our experiments, the label and feature matrices are $\boldsymbol{Y} \in \mathbb{R}^{174\times 502}$ and $\boldsymbol{F}_c \in \mathbb{R}^{502\times 68}$, respectively. Similar to before, we set $\boldsymbol{F}_r$ to be the identity matrix. Figure 4 shows the average multi-label learning and relative completion errors of different methods over 10 random trials. Notice that our proposed method achieves the smallest transductive label errors, in particular, when the fraction of missing entries is smaller than 0.7. Moreover, our method significantly performs better than other algorithms for matrix completion with side information when the missing entries fraction is less than 0.7, thanks to our more general setting of dealing with high-rank data matrices.

Multi-label learning for image classification. The NUS-WIDE dataset contains more than 269,648 images, where each images is represented by a 128-dimensional feature vector (Spyromitros-Xioufis et al. 2014). The images are collected from Flickr, where each image has 81 unique tags (labels) which will be used for our evaluation. In our experiments, we randomly 1,079 images from the dataset (sampling images in the list with steps of 250). Thus the ground truth matrix is $\boldsymbol{Y} \in \mathbb{R}^{81\times 1079}$ and the feature matrix is $\boldsymbol{F}_c \in \mathbb{R}^{1079\times 128}$. We set $\boldsymbol{F}_r$ to be the identity matrix. In this case, the data matrix $\boldsymbol{Y}$ is high-rank, since the top 75 singular values out of 81 are larger than 1. Figure 5 shows the average results of different algorithms as a function of the percentage of missing entries of $\boldsymbol{Y}$, over 10 random trials. Notice that our method achieves the smallest transductive label errors as well as relative completion error across all percentages of missing entries. In particular, with less than 70% missing entries, our method achieves the smallest completion error. Similar to previous cases, the label errors of different methods are smaller than their completion error, as they correctly predict the sign in most cases while the recovered entries can be far from the ground-truth.

Conclusions

We studied the problem of high-rank matrix completion with side information. We cast the problem as finding a factorization of the data into the product of side information matrices with an unknown interaction matrix, under which each column of the data matrix can be reconstructed using a sparse combination of its other columns. We proposed a lifting framework, where we coupled sparse coefficients and missing values and defined an equivalent optimization, amenable to convex relaxation. We proposed an efficient implementation of our convex framework using a Linearized Alternating Direction Method. Experiments on synthetic and real data demonstrated that our method outperforms existing techniques in dealing with both low and high rank matrices.

Acknowledgements

This work is partially supported by NSF IIS-1657197 award and startup funds from the Northeastern University, College of Computer and Information Science.

References

Adams, R. P.; Dahl, G. E.; and Murray, I. 2010. Incorporating side information in probabilistic matrix factorization with gaussian processes. *Conference on Uncertainty in Artificial Intelligence*.

Agarwal, D., and Chen, B. C. 2009. Regression-based latent factor models. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2008. Convex multi-task feature learning. *Machine Learning* 73(3).

Balzano, L.; Szlam, A.; Recht, B.; and Nowak, R. 2012. K-subspaces with missing data. In *Statistical Signal Processing Workshop*.

Biswas, P.; Lian, T.-C.; Wang, T.-C.; and Ye, Y. 2006. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks* 2(2).

Briggs, F.; Huang, Y.; Raich, R.; Eftaxias, K.; Lei, Z.; Cukierski, W.; Hadley, S. F.; Hadley, A.; Betts, M.; Fern, X. Z.; and et. al. 2013. New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. In *IEEE International Workshop on Machine Learning for Signal Processing*.

Cai, J.-F.; Candès, E.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20(4).

Candes, E., and Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9(6).

Chen, Y.; Xu, H.; Caramanis, C.; and Sanghavi, S. 2011. Robust matrix completion with corrupted columns. *IEEE Transactions on Information Theory* 62(1).

Chiang, K.-Y.; Hsieh, C.-J.; Natarajan, N.; Dhillon, I.; and Tewari, A. 2014. Prediction and clustering in signed networks: a local to global perspective. *Journal of Machine Learning Research* 15(1).

Chiang, K.-Y.; Hsieh, C.-J.; and Dhillon, I. 2015. Matrix completion with noisy side information. In *Advances in Neural Information Processing Systems*.

Chiang, K. Y.; Hsieh, C. J.; and Dhillon, I. 2016. Robust principal component analysis with side information. In *International Conference on Machine Learning*.

Elhamifar, E., and Vidal, R. 2009. Sparse subspace clustering. *IEEE Conference on Computer Vision and Patter Recognition*.

Elhamifar, E., and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(11).

Elhamifar, E. 2016. High-rank matrix completion and clustering under self-expressive models. In *Advances in Neural Information Processing Systems*, 73–81.

Eriksson, B.; Balzano, L.; and Nowak, R. 2012. High-rank matrix completion. In *International Conference on Artificial Intelligence and Statistics*.

Ghahramani, Z.; Hinton, G. E.; et al. 1996. The em algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto.

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

Gruber, A., and Weiss, Y. 2004. Multibody factorization with uncertainty and missing data using the em algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Jain, P., and Dhillon, I. 2013. Provable inductive matrix completion. *arXiv preprint, arXiv:1306.0626*.

Keshavan, R.; Montanari, A.; and Oh, S. 2010. Matrix completion from a few entries. *IEEE Transactions on Information Theory* 56(6).

Knott, M., and Bartholomew, D. J. 1999. Latent variable models and factor analysis. *Journal of Educational Statistics* 7(4).

Liu, G., and Li, P. 2016. Low-rank matrix completion in the presence of high coherence. *IEEE Transactions on Signal Processing* 64(21).

Lu, J.; Liang, G.; Sun, J.; and Bi, J. 2016. A sparse interactive model for matrix completion with side information. In *Advances in Neural Information Processing Systems*.

Menon, A. K.; Chitrapura, K. P.; Garg, S.; Agarwal, D.; and Kota, N. 2011a. Response prediction using collaborative filtering with hierarchies and side-information. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Menon, A. K.; Chitrapura, K.-P.; Garg, S.; Agarwal, D.; and Kota, N. 2011b. Response prediction using collaborative filtering with hierarchies and side-information. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Natarajan, N., and Dhillon, I. 2014. Inductive matrix completion for predicting gene–disease associations. *Bioinformatics* 30(12).

Porteous, I.; Asuncion, A.; and Welling, M. 2010. Bayesian matrix factorization with side information and dirichlet process mixtures. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Rennie, J., and Srebro, N. 2005. Fast maximum margin matrix factorization for collaborative prediction. In *International conference on Machine learning*.

Sindhwani, V.; Bucak, S.; Hu, J.; and Mojsilovic, A. 2010. One-class matrix completion with low-density factorizations. In *International Conference on Data Mining*.

Singer, A., and Cucuringu, M. 2010. Uniqueness of low-rank matrix completion by rigidity theory. *SIAM Journal on Matrix Analysis and Applications* 31(4).

Singer, A. 2008. A remark on global positioning from local distances. *Proceedings of the National Academy of Sciences* 105(28).

Soltanolkotabi, M.; Elhamifar, E.; and Candes, E. 2014. Robust subspace clustering. *Annals of Statistics* 42(2).

Spyromitros-Xioufis, E.; Papadopoulos, S.; Kompatsiaris, I. Y.; Tsoumakas, G.; and Vlahavas, I. 2014. A comprehensive study over vlad and product quantization in large-scale image retrieval. *IEEE Transactions on Multimedia* 16(6).

Tipping, M. E., and Bishop, C. M. 1999a. Mixtures of probabilistic principal component analyzers. In *International Conference on Artificial Neural Networks*.

Tipping, M. E., and Bishop, C. M. 1999b. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61(3).

Turnbull, D.; Barrington, L.; Torres, D.; and Lanckriet, G. 2008. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing* 16(2).

Xu, M.; Jin, R.; and Zhou, Z.-H. 2013. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in Neural Information Processing Systems*.

Yang, J., and Yuan, X. 2013. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation* 82(281).