Cluster Developing 1-Bit Matrix Completion

1st Chengkun Zhang
Discipline of Business Analytics
The University of Sydney
Camperdown, NSW 2006, Australia
chengkun.zhang@sydney.edu.au

2nd Junbin Gao
Discipline of Business Analytics
The University of Sydney
Camperdown, NSW 2006, Australia
junbin.gao@sydney.edu.au

3rd Steven Lu
Discipline of Marketing
The University of Sydney
Camperdown, NSW 2006, Australia
steven.lu@sydney.edu.au

***Abstract*—1-bit matrix completion plays a significant role in recommender systems. However, conventional 1-bit methods do not consider the interactive clustering nature of customers and products. To tackle the gap, we introduce Group-Specific 1-bit Matrix Completion (GS1MC) by first-time consolidating cluster effects in 1-bit matrix completion tasks. To empower GS1MC even when grouping information is unobtainable, GS1MC is extended as Cluster Developing Matrix Completion (CDMC) by integrating subspace clustering techniques. CDMC can group entities and utilize their cluster effects simultaneously. Experiments show that GS1MC outperforms current 1-bit matrix completion methods and CDMC successfully captures items' genre features only with sparse binary rating data. Notably, GS1MC provides a new insight to incorporate and evaluate the efficacy of different clustering methods while CDMC can be served as a new tool to explore unrevealed social behavior or market phenomena.**

***Index Terms*—Matrix Completion, Subspace Clustering**

I. Introduction

Recommender systems aim at improving customers' experience by maximizing the use of available information, including (i) collaborative filtering (Deshpande and Karypis, 2004; Linden et al., 2003; Resnick et al., 1994; Sarwar et al., 2001), which utilizes user-item interactive data, such as ratings or clicking behavior; and (ii) content-based methods (Billsus and Pazzani, 2000; Pazzani and Billsus, 1997; Shoham, 1997), which utilizes attribute information, such as category or textual profiles.

Collaborative filtering can be considered as a special case of matrix completion task. It has become a cornerstone of most powerful recommender systems while it is mainly founded on two main streams of methods: neighbourhood-based methods (Deshpande and Karypis, 2004; Linden et al., 2003; Resnick et al., 1994; Sarwar et al., 2001) and model-based methods (Bell et al., 2007; Breese et al., 1998; Koren, 2008; Paterek, 2007; Takács et al., 2008, 2009). Though neighbourhood-based methods are easy to interpret and implement, they suffer from low prediction accuracy when observed data is sparse. Alternatively, model-based methods define a parameterized model to capture the key features of training data. Numerous model-based approaches were tested in previous research, including Support Vector Machines (Grčar et al., 2006), Maximum Entropy (Zitnick and Kanade, 2004), Boltzmann Machines (Salakhutdinov et al., 2007) and Singular Value Decomposition (SVD) (Koren, 2008; Paterek, 2007; Takács et al., 2008, 2009).

Under the assumption that the continuation of data points is convincing and compelling, standard collaborative filtering methods take observed entries of a rating matrix as real numbers. However, the adequacy of this measurement is undoubtedly questionable when intervals between data points are different. For instance, personal judgments from different customers vary as a result of personality. Generous customers tend to give fairly higher ratings than curmudgeon customers. Thus, instead of taking data as continuous numbers, it is more feasible considering them as categories, especially the polarity case. Bhaskar and Javanmard (2015); Cai and Zhou (2013); Davenport et al. (2014) use binary ratings generated by the real-valued entries, and experiments show their approach performs significantly better than continuous methods.

Although 1-bit matrix completion proves its success in recommender systems, same as most other matrix completion methods, it suffers from a fundamental limitation: every user/item is treated merely as standalone individuals, which arrogantly ignores the homogeneity of products and the clustering characteristic of social behaviors. For instance, some fundamental management theory points out that people have a propensity of conformity nature based on demographic, psychographic and behavioral variables (Kotler, 2009). Recent research is noticed focusing on integrating preliminary clusters into continuous matrix completion task (Bi et al., 2017) and experiments demonstrate that their approach outclasses traditional SVD methods. However, so far there is not any 1-bit matrix completion methods taking cluster information into consideration. Moreover, state-of-the-art recommender systems either take clustering as an independent task or treat cluster identities as preliminaries, there is not any existing method for summarizing clusters along with matrix completion. Since the clustering nature of individuals plays a vital role in social behavior research, it is consequently significant to introduce a new method that learns the clusters and utilizes the clustering effects for matrix completion at the same time.

In this work, we propose two methods: (i) *group-specific 1-bit matrix completion* (GS1MC), which integrates known group identities into conventional methods ; and (ii) *cluster developing matrix completion* (CDMC) which learns cluster identities automatically. Experimentally, we show that the proposed GS1MC outperforms its baselines and CDMC successfully captures targets' generic features and achieves convergence of both user/item clusters.

II. Preliminaries

In this section, we discuss some preliminary knowledge of our work, including traditional SVD-based matrix completion, the framework of probabilistic 1-bit matrix completion and sparse subspace clustering technique.

A. Matrix Completion

Traditional continuous matrix completion methods consider an incomplete user-item interaction matrix $\hat{\mathbf{R}} = (\hat{r}_{ui}) \in \mathbb{R}^{m_1 \times m_2}$ for m_1 users and m_2 items, where each existing entry $\hat{r}_{ui}$ stands for an explicit rating given by user-u towards item-i. The typical SVD-based method can be formulated as:

$$\begin{aligned} &\arg\min_{\mathbf{U},\mathbf{V}} J = \frac{1}{2}\|\hat{\mathbf{R}} - \mathbf{U}\mathbf{V}^T\|^2 \\ &\text{s.t. Columns of } \mathbf{U}, \mathbf{V} \text{ are mutually orthogonal.} \end{aligned} \tag{1}$$

The above formulation is extended in different directions. For instance, a variety of regularization terms are applied for specific considerations (Zhu et al., 2016), and biased version of SVD methods (Koren and Bell, 2015; Paterek, 2007) are also introduced to consider the general preference of each user and discrimination of each item. Moreover, users/items can also be allocated into clusters and aggregated with group effects. For instance, taking preliminary cluster identities as inputs, a set of latent variables representing the group bias (Bi et al., 2017) can be learned.

B. 1-Bit Matrix Completion

Though matrix completion methods have been used for recommender systems for long, 1-bit matrix completion (Davenport et al., 2014) has been officially introduced lately. Varied from the continuous model which applies numerical computation on discrete rating data improperly, original observation is converted into an incomplete binary matrix $\hat{\mathbf{Y}}$ first. The objective of the task is formalized as learning an $n_1 \times n_2$ latent variable matrix $\mathbf{M} = (\hat{M}_{ui}) \in \mathbb{R}^{n_1 \times n_2}$ to estimate the binary data via:

$$Y_{ui} = \begin{cases} +1, & \text{with probability } f(M_{ui}) \\ -1, & \text{with probability } 1 - f(M_{ui}) \end{cases} \quad (u,i) \in \Omega, \tag{2}$$

where Ω is the set of all the observed entries and f can be a sigmoid function defined as:

$$f(z) = \frac{1}{1 + \exp\{-z\}}. \tag{3}$$

Similar to other low-rank matrix completion methods, a wide variety of approaches have been applied to constrain the latent variable matrix (Bhaskar and Javanmard, 2015; Cai and Zhou, 2013; Davenport et al., 2014). However, these methods treat every instance as autonomous individuals, ignoring the ground truth that users/items tend to have any baseline or belong to certain clusters. Furthermore, a severe gap is that there is not any methodology that can both learn the cluster identities and leverage their group effects for matrix completion at the same time.

C. Subspace Clustering

Subspace clustering (Brbić and Kopriva, 2018; Elhamifar and Vidal, 2013; Liu et al., 2010) aims at grouping data points in their low-dimensional subspace via the self-expressive matrix, which represents each instance by an affine combination of other points within the same subspace. Assume the noise-free dataset $\mathbf{X} \in \mathbb{R}^{D \times N_l}$ can be separated into n subspaces $\{S_l\}_{l=1}^n$ of dimensions $D = \{d_l\}_{l=1}^n$. Each data point $\mathbf{x}_i \in \cup_{l=1}^n S_l$ can be reconstructed by a combination of other points within the same subspace as:

$$\mathbf{x}_i = \mathbf{X}\mathbf{c}_i, c_{ii} = 0. \tag{4}$$

Since representations for each data point by the other should be as sparse as possible, which results in an NP-hard problem, different norm functions are applied to get around the NP difficulty. Then, (4) can be reformulated in the fowllowing form:

$$\min \|\mathbf{C}\|_{norm} \text{ s.t. } \mathbf{X} = \mathbf{X}\mathbf{C}, \ \text{diag}(\mathbf{C}) = 0, \tag{5}$$

where $\mathbf{C} \triangleq [\mathbf{c}_1\mathbf{c}_2...\mathbf{c}_N] \in \mathbb{R}^{N \times N}$ corresponds to the non-trivial subspace-sparse representation for all the data points $\mathbf{x}_i$s and $\|\cdot\|_{norm}$ can be selected in favor of different application focuses.

Since user-item interaction data is exceedingly sparse and high-dimensional, many dimensions are irrelevant and covered by noise. Meanwhile, latent variables, representing the profiles of individuals, are not strictly around any centroids. Thus, conventional clustering methods that utilizing the spatial proximity is not applicable in this case while subspace clustering has several advantages. Firstly, subspace clustering methods aim at grouping the points that are not necessarily close but lie in the same subspace, which does not depend on the spatial characteristic of the data anymore. Then, as sparse subspace clustering deploys a convex approach to pick out the sparse representation of each point, the optimization process automatically eliminates some common issues of clustering methods, such as sensitivity to the ideal cluster size and bordering matter of the overlapped subspace.

III. Group-specific 1-Bit Matrix Completion (GS1MC)

In this section, we utilize known cluster identities with 1-bit matrix completion task, such that the group effects can be learned along with latent variable training process.

A. Model Framework

Suppose $\hat{\mathbf{Y}}$ is the observed $n_1 \times n_2$ binary rating matrix with entries equal to '$+1$' or '-1', corresponding to "interested" or "not interested", where n_1/n_2 is the number of users/items, the "not observed" entries are represented by '0'. Ω stands for the observed user-item pairs, i.e. entries with same indexes as '$+1$' and '-1' in $\hat{\mathbf{Y}}$. We construct the latent variable matrix as $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$. To make predictions for missing entries by (2), our main objective is to find the estimation of $\mathbf{M}$ that best approximates the observed data.

Since it has been proved that the exact low-rank method results in a high convergence rate (Bhaskar and Javanmard, 2015), especially when the fraction of revealed entries is small (*cold-start problem*), we choose to apply an exact low-rank constraint on $\mathbf{M}$. We assume that every user/item is classified into one single user/item group, respectively. We formulate the latent variable matrix $\mathbf{M}$ by integrating group bias into matrix factorization. Then each entry in $\mathbf{M}$ can be written as:

$$M_{ui} = (\mathbf{p}_u + \mathbf{s}_{v_u})'(\mathbf{q}_i + \mathbf{t}_{j_i}). \quad (6)$$

Here $\mathbf{p}_u \in \mathbb{R}^K$ and $\mathbf{q}_i \in \mathbb{R}^K$ are K-dimensional latent factors standing for user u's preference and item i's character, while $\mathbf{s}_{v_u} \in \mathbb{R}^K$ and $\mathbf{t}_{j_i} \in \mathbb{R}^K$ represent biases of clusters that individuals belong to. For instance, $\mathbf{s}_{v_u}$ means the cluster effect of user cluster v_u, i.e. the cluster user u belongs to. Here we have assumed that there are m_1 users clusters and m_2 item clusters, such that $v_u \in \{1, 2, ..., m_1\}$ and $j_i \in \{1, 2, ..., m_2\}$. Then, the group effects of the user and item clusters can be formalized as:

$$\mathbf{S}_U = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{m_1}]^T \in \mathbb{R}^{m_1 \times K} \quad \text{, and}$$
$$\mathbf{T}_J = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_{m_2}]^T \in \mathbb{R}^{m_2 \times K}.$$

For the sake of convenience, in terms of matrix notations, we assume the user-item interaction data $\hat{\mathbf{Y}} = (\hat{Y}_{ui})$ and its corresponding latent variable $\mathbf{M} = (M_{ui})$ have been permuted such that the first U_1 rows corresponds user cluster 1, followed by U_2 rows corresponding to user cluster 2, ..., and the last U_{m_1} rows corresponding to user cluster m_1. Similarly, the columns have been rearranged accordingly. After this alteration, the decomposition (6) can be written as the following matrix format:

$$\mathbf{M} = (\mathbf{P} + \mathbf{S})(\mathbf{Q} + \mathbf{T})^T \quad (7)$$

where

$$\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{n_1}]^T \in \mathbb{R}^{n_1 \times K};$$
$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_{n_2}]^T \in \mathbb{R}^{n_2 \times K};$$
$$\mathbf{S} = [\mathbf{s}_1 \mathbf{1}_{U_1}^T, \mathbf{s}_2 \mathbf{1}_{U_2}^T, ..., \mathbf{s}_{m_1} \mathbf{1}_{U_{m_1}}^T]^T \in \mathbb{R}^{n_1 \times K};$$
$$\mathbf{T} = [\mathbf{t}_1 \mathbf{1}_{J_1}^T, \mathbf{t}_2 \mathbf{1}_{J_2}^T, ..., \mathbf{t}_{m_2} \mathbf{1}_{J_{m_2}}^T]^T \in \mathbb{R}^{n_2 \times K}.$$

Here $\mathbf{1}_m$ stands for m-dimensional (column) vector of all '1's. In other words, instances of group effects matrix $\mathbf{S}_U$ and $\mathbf{T}_J$ have been duplicated in order to match the dimension of matrix $\mathbf{P}$ and $\mathbf{T}$. For the convenience of the transformation between $\mathbf{S}$, $\mathbf{T}$ and $\mathbf{S}_U$, $\mathbf{T}_J$, we define the following two matrices:

$$\mathbf{I}_U^{m_1 \times n_1} = \begin{bmatrix} \mathbf{1}_{U_1}^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{U_2}^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{U_{m_1}}^T \end{bmatrix}, \text{ and}$$

$$\mathbf{I}_J^{m_2 \times n_2} = \begin{bmatrix} \mathbf{1}_{J_1}^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{J_2}^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{J_{m_2}}^T \end{bmatrix}.$$

Thus, it is clear that $\mathbf{S}$, $\mathbf{T}$ and $\mathbf{S}_U$, $\mathbf{T}_J$ can be transformed to each other by:

$$\mathbf{S} = \mathbf{I}_U^T \mathbf{S}_U \text{ and } \mathbf{T} = \mathbf{I}_J^T \mathbf{T}_J. \quad (8)$$

Then, (7) can be rewritten as:

$$\mathbf{M} = (\mathbf{P} + \mathbf{I}_U^T \mathbf{S}_U)(\mathbf{Q} + \mathbf{I}_J^T \mathbf{T}_J). \quad (9)$$

B. Objective and Optimization

Following the objective function of basic 1-bit matrix completion method (Bhaskar and Javanmard, 2015), the fundamental loss function is defined as:

$$F_{\Omega, \hat{\mathbf{Y}}}(\mathbf{M}) = - \sum_{(u,i) \in \Omega} \{ \mathbb{I}_{(\hat{Y}_{ui}=1)} \log(f(M_{ui})) + \mathbb{I}_{(\hat{Y}_{ui}=-1)} \log(\mathbf{1} - f(M_{ui})) \},$$

where $f(\mathbf{M})$ is the matrix operation of applying f over $\mathbf{M}$ element-wise, and $\mathbf{1}$ is the all 1's matrix. Here $\mathbb{I}_\mu$ is the indicator function, i.e. $\mathbb{I}_\mu = 1$ when μ is true, else $\mathbb{I}_\mu = 0$. $\mathbb{I}_\mu$ can be implemented as two mask matrices $\mathbf{Y}_1^{n1 \times n2} = (Y_1(u,i))$ and $\mathbf{Y}_{-1}^{n1 \times n2} = (Y_{-1}(u,i))$ of the same size as $\mathbf{M}$, where $Y_1(u,i) = 1$ if $\hat{Y}_{ui} = 1$, otherwise $Y_1(u,i) = 0$. Similarly, $Y_{-1}(u,i) = 1$ if $\hat{Y}_{ui} = -1$, otherwise $Y_{-1}(u,i) = 0$. Then, the fundamental loss function can be transformed into:

$$F_{\Omega, \hat{\mathbf{Y}}}(\mathbf{M}) = - \sum_{\Omega} (\mathbf{Y}_1 \circ \log f(\mathbf{M}) + \mathbf{Y}_{-1} \circ \log(\mathbf{1} - f(\mathbf{M})), \quad (10)$$

where $\circ$ means the element-wise product of two matrices. We notate $\Gamma = (\mathbf{P}, \mathbf{Q}, \mathbf{S_U}, \mathbf{T_J})$ and $R^0 = \{\hat{Y}_{ij} : (i,j) \in \Omega\}$. After adding the regularization term, the new loss function can be formulated as:

$$L(\Gamma | R^0) = F_{\Omega, \hat{\mathbf{Y}}}(\mathbf{M}) + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{S}_U\|_F^2 + \|\mathbf{Q}\|_F^2 + \|\mathbf{T}_J\|_F^2). \quad (11)$$

Our goal is to predict the missing entries of the rating matrix, which can be computed by:

$$\hat{\Gamma} = \arg\min_{\Gamma} L(\Gamma | R^0). \quad (12)$$

We solve the optimization problem (12) via the Alternating direction method of multipliers (ADMM). Firstly, to update the latent factors of users and user clusters, we fix $\mathbf{Q}$ and $\mathbf{T}_J$, and minimize (12) by estimating $\hat{\mathbf{P}}$ and $\hat{\mathbf{S_U}}$:

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} \{ F_{\Omega, \hat{\mathbf{Y}}}(\mathbf{M}) + \lambda \|\mathbf{P}\|_F^2 \}, \quad (13)$$

$$\hat{\mathbf{S}}_U = \arg\min_{\mathbf{S}_U} \{ F_{\Omega, \hat{\mathbf{Y}}}(\mathbf{M}) + \lambda \|\mathbf{S}_U\|_F^2 \}. \quad (14)$$

Then for items and item clusters, we fix $\mathbf{P}$ and $\mathbf{S}_U$, conducting following computations:

$$\hat{\mathbf{Q}} = \arg\min_{\mathbf{Q}} \{ F_{\Omega, \hat{\mathbf{Y}}}(\mathbf{M}) + \lambda \|\mathbf{Q}\|_F^2 \}, \quad (15)$$

$$\hat{\mathbf{T}}_J = \arg\min_{\mathbf{T}_J} \{ F_{\Omega, \hat{\mathbf{Y}}}(\mathbf{M}) + \lambda \|\mathbf{T}_J\|_F^2 \}. \quad (16)$$

Each of sub-problems (13) - (16) can be solved by the gradient descent algorithm. We can work out the gradient in the following way. First we take f as the sigmoid function defined in (3), then it is easy to check that:

$$\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{M}} = \mathbf{Y}_1 \circ (f(\mathbf{M}) - \mathbf{1}) + \mathbf{Y}_{-1} \circ f(\mathbf{M}).$$

Considering (7), with the matrix differentiation chain rule, it can be proved that:

$$\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{P}} = [\mathbf{Y}_{-1} + (\mathbf{Y}_1 + \mathbf{Y}_{-1}) \circ (f(\mathbf{M}) - \mathbf{1})](\mathbf{Q} + \mathbf{T}) \tag{17}$$

$$\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{Q}} = [\mathbf{Y}_{-1}^T + (\mathbf{Y}_1^T + \mathbf{Y}_{-1}^T) \circ (f(\mathbf{M}^T) - \mathbf{1})]^T(\mathbf{P} + \mathbf{S}). \tag{18}$$

On the one hand, we have

$$\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}} = \frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{P}} \quad \text{and} \quad \frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{T}} = \frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{Q}}.$$

On the other hand, according to (8), it is clear to state that:

$$\frac{\partial \mathbf{S}}{\partial \mathbf{S}_U} = \mathbf{I}_K \otimes \mathbf{I}_U^T \quad \text{and} \quad \frac{\partial \mathbf{T}}{\partial \mathbf{T}_J} = \mathbf{I}_K \otimes \mathbf{I}_J^T,$$

where $\mathbf{I}_K$ is the identity matrix.

According to the chain rules, we finally get:

$$\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}_U} = \mathbf{I}_U \frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}} \quad \text{, and}$$
$$\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{T}_J} = \mathbf{I}_J \frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{T}}.$$

In other words, the sum of the first U_1 rows of $\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}}$ is the first row of $\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}_U}$, the sum of the next U_2 rows of $\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}}$ is the second row of $\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}_U}$, ..., and the sum of the last U_{m_1} rows of $\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}}$ becomes the m_1-th row (the last row) of $\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{S}_U}$. The similar way can be used to construct $\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{T}_J}$ from $\frac{\partial F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M})}{\partial \mathbf{T}}$.

GS1MC has been thoroughly tested on synthetic and real-world dataset with promising results. The experiment details has been discussed in section V.

IV. Cluster Developing Matrix Completion (CDMC)

In this section, based on the compelling results of GS1MC, we take one step further. In terms of the case that cluster information is not available, we aggregate GS1MC and subspace clustering to learn the cluster identities of users/items automatically. This section will be divided into three parts. Firstly, we provide a description for the problem setting. Secondly, we validate the feasibility of using self-expressive matrix in 1-bit matrix completion task via experiments. Afterwards, the CDMC algorithm is introduced in detail.

A. Problem Setting

We decide to introduce subspace clustering into GS1MC according to three main reasons. Firstly, the model (GS1MC) proposed in section III takes cluster identities as preliminary information. However, in most practical scenarios, it might be inaccessible to such details, especially for the *cold-start problem*. Secondly, since the original binary user-item interaction data is extremely sparse, it is controversial to apply standard clustering techniques directly. Thirdly, common clustering methods may take advantage of distance between points to divide the space into different partitions. Nevertheless, regarding a latent variable model, market segments may not necessarily congregate based on spatial proximity but lie in a subspace.

A common dilemma for most clustering techniques is the drawback that they might be decidedly sensitive to improper initialization, such as cluster size and centroids. As long as the size of user/item clusters is unrevealed and each data points can have an infinite number of expressions in terms of the other, it is advisable to incorporate subspace clustering technique to optimize a sparse representation among these expressions through a convex realization approach.

B. Feasibility of Subspace Clustering Technique in 1-Bit Matrix Completion

Since subspace clustering is mainly founded on the self-expressive matrix (5), which has not been discussed in the 1-bit matrix completion latent variable framework before, we test the feasibility of such technique via the following formulation:

$$F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M}) + \lambda_1\|\mathbf{Z}\|_* + \frac{\lambda_2}{2}\|\mathbf{M} - \mathbf{M}\mathbf{Z}\|_F^2, \quad \text{s.t. } \mathbf{Z} = \mathbf{Z}^T. \tag{19}$$

Instead of applying the low-rank constraint directly on $\mathbf{M}$, we extract its intrinsic structure as $\mathbf{Z}$ via the Frobenius norm $\|\mathbf{M} - \mathbf{M}\mathbf{Z}\|_F$ and transfer the low-rank constraint on it through $\|\mathbf{Z}\|_*$. In other words, this approach is the relaxed form of the original formulation and more concerned about users/items "hidden similarities ($\mathbf{Z}$)" other than their "hidden profiles ($\mathbf{M}$)".

We conducted the Augmented Lagrange Multiplier (ALM) with Alternating Direction Minimizing (ADM) method to solve (19). Every optimization iteration can be divided into two steps.

1. Updating $\mathbf{M}$ when fixing $\mathbf{Z}$ to its current value $\mathbf{Z}^{(t)}$:

While $\mathbf{Z}$ has been fixed, it can be treated as the constant matrix, so the problem (19) can be simplified as:

$$L_1(\mathbf{M}) = F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M}) + \frac{\lambda_2}{2}\|\mathbf{M} - \mathbf{M}\mathbf{Z}^{(t)}\|_F^2. \tag{20}$$

It is not difficult to notice that minimizing $L_1(\mathbf{M})$ with respect to $\mathbf{M}$ can be treated as a regularized logistic regression problem when f is defined as a sigmoid function. Then (20) can be solved by using the standard gradient descent algorithm. Denote $\mathbf{Z}_I = (\mathbf{I} - \mathbf{Z}^{(t)})(\mathbf{I} - \mathbf{Z}^{(t)})^T$, we have:

$$\frac{\partial L_1(\mathbf{M})}{\partial \mathbf{M}} = \mathbf{Y}_1 \circ (f(\mathbf{M}) - \mathbf{1}) + \mathbf{Y}_{-1} \circ f(\mathbf{M}) + \lambda_2 \mathbf{M}\mathbf{Z}_I. \tag{21}$$

TABLE I: %Prediction Accuracy: Introducing subspace clustering in 1-bit matrix completion.

$\lambda_1 = 800; \lambda_2 = 0.006$	jester-1	jester-2	jester-3	Movielens-100k
Subspace Clustering Assisted - 1BMC	71.8	72.6	72.6	71.4
1BMC	71.6	72.8	72.7	72.3

2. Updating Z when fixing $\mathbf{M}$ to its current value $\mathbf{M}^{(t)}$:
In this case, (19) becomes the following problem

$$L_2(\mathbf{Z}) = \|\mathbf{Z}\|_* + \frac{\lambda_2/\lambda_1}{2}\|\mathbf{M}^{(t)} - \mathbf{M}^{(t)}\mathbf{Z}\|_F^2. \tag{22}$$

(22) can be solved by using the following theorem:

Theorem 1 (Vidal and Favaro (2014)): Let $\mathbf{M}^{(t)} = \mathbf{U\Sigma V}$ be the singular vector decomposition (SVD) of $\mathbf{M}^{(t)}$, where the diagonal entries of $\mathbf{\Sigma} = \text{diag}(\{\sigma_i\})$ are the singular values of $\mathbf{M}^{(t)}$ in descending order. The optimal solution to (22) is given by:

$$\mathbf{Z}^* = \mathbf{V}\mathcal{P}_{\lambda_2/\lambda_1}(\mathbf{\Sigma})\mathbf{V}^T \tag{23}$$

where the operator $\mathcal{P}_{\lambda_2/\lambda_1}$ acts on the diagonal entries of $\mathbf{\Sigma}$ as

$$\mathcal{P}_{\lambda_2/\lambda_1}(x) = \begin{cases} 1 - \frac{\lambda_1}{\lambda_2 x^2}; & x > \sqrt{\frac{\lambda_1}{\lambda_2}} \\ 0; & x \leq \sqrt{\frac{\lambda_1}{\lambda_2}} \end{cases}. \tag{24}$$

With the above formulation, we conduct decisive experiments to prove the feasibility of introducing subspace clustering techniques into 1-bit matrix completion method. We compare the subspace clustering-assisted 1-bit matrix completion with 1BMC (Davenport et al., 2014) on benchmark datasets: Jester (Goldberg et al., 2001) and Movielens $100k$ (Harper and Konstan, 2016)[1]. Following the problem setting of previous literature (Bhaskar and Javanmard, 2015; Cai and Zhou, 2013; Davenport et al., 2014), the original observations, have been quantized as '+1' and '-1' according to whether they are above or below the average score. The comparison results are listed in Table I.

As shown in Table I, the relaxed subspace clustering-assisted method has achieved comparable performance even the low-rank constraint has been only applied on the intrinsic structure. Thus, we deduce the feasibility of self-expressive matrix in 1-bit matrix completion task. In the following section, we will introduce this technique into the proposed GS1MC algorithm.

C. CDMC Algorithm

Based on GS1MC, we extend the scope of the method to developing clusters during the latent variable training process. Since subspace clustering encourages data points lie on a low-rank subspace, the original formulation of GS1MC (25) is modified as:

$$\begin{aligned} L(\Gamma|R^0) =& F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M}) + \lambda_1(\|\mathbf{P}\|_F^2 + \|\mathbf{S}_U\|_F^2 + \|\mathbf{Q}\|_F^2 + \|\mathbf{T}_J\|_F^2) \\ &+ \lambda_2(||\mathbf{P}+\mathbf{S}||_* + ||\mathbf{Q}+\mathbf{T}||_*), \end{aligned} \tag{25}$$

[1]The proposed method is implemented with Intel Xeon CPU (2.8GHz) and 128GB memory.

where the nuclear norm encourages the blocklike pattern of user/item latent variables, i.e. points tend to gather around low-rank subspace. With similar ADMM process to GS1MC, we have:

$$\hat{\mathbf{P}} = \min_{\mathbf{P}}\{F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M}) + \lambda_1\|\mathbf{P}\|_F^2 + \lambda_2||\mathbf{P}+\mathbf{S}||_*\} \tag{26}$$

$$\hat{\mathbf{S}}_U = \min_{\mathbf{S}_U}\{F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M}) + \lambda_1\|\mathbf{S}_U\|_F^2 + \lambda_2||\mathbf{P}+\mathbf{S}||_*\} \tag{27}$$

$$\hat{\mathbf{Q}} = \min_{\mathbf{Q}}\{F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M}) + \lambda_1\|\mathbf{Q}\|_F^2 + \lambda_2||\mathbf{Q}+\mathbf{T}||_*\} \tag{28}$$

$$\hat{\mathbf{T}}_J = \min_{\mathbf{T}_J}\{F_{\Omega,\hat{\mathbf{Y}}}(\mathbf{M}) + \lambda_1\|\mathbf{T}_J\|_F^2 + \lambda_2||\mathbf{Q}+\mathbf{T}||_*\} \tag{29}$$

The other difference is that instead of having pre-defined cluster identities, we insert a clustering procedure based on the gradually recovering matrix and re-utilize this group identity in the next iteration. After each iteration of updating latent variables $\mathbf{P}, \mathbf{S}_U, \mathbf{Q}$ and $\mathbf{T}_J$, we construct $\mathbf{M}$ and $\mathbf{M}^T$ via (9). We consider $\mathbf{M}$ lie in m_2 disjoint subspaces $\{S_i\}_{i=1}^{m_2}$ while $\mathbf{M}^T$ lie in $\{S_i\}_{i=1}^{m_1}$. To get the low-rank blocklike pattern (Liu et al., 2010), we deploy nuclear-norm relaxation of the self-expressive matrix to obtain the sparse representation $\mathbf{C}_1$ and $\mathbf{C}_2$ for users and items respectively, namely:

$$\min \|\mathbf{C}_l\|_* \text{ s.t. } \begin{cases} \mathbf{M} = \mathbf{M}\mathbf{C}_1 \text{ or} \\ \mathbf{M} = \mathbf{C}_2^T\mathbf{M}, \end{cases} \quad l = \{1, 2\}. \tag{30}$$

Next, a non-directional weighted graph of $\mathbf{C}_1$ is built as $\mathbb{G}_{\not{k}} = (\mathbf{N_1}, \mathbf{W_1})$, where $\mathbf{N_1}$ is the nodes regarding all sparse representations in $\mathbf{C}_1$, and $\mathbf{W_1}$ is the weighted edges between each pair of $\mathbf{N_1}$. A natural choice of the weighted matrix is that the nodes within the same subspace will share non-zero weighted edges while the other edges are zero-weighted. In other words, an adjacency matrix can be constructed by $\mathbf{W}_1 = |\mathbf{C}_1| + |\mathbf{C}_1^T|$ (Elhamifar and Vidal, 2013), where the non-zero entries represents latent variable pairs that actually lie in the same subspace. Then, we apply spectral clustering (Ng et al., 2002) on $\mathbf{W}_1$ to procure item clusters. Similar method is conducted to build $\mathbf{W_2}$ for user cluster developing.

After estimating new clusters in each iteration, we update the identities of users/items and learn their new group effects. Alternatively speaking, CDMC conducts sparse subspace clustering and GS1MC iteratively. The complete algorithm is shown in Algorithm 1.

V. Experiments

In this section, we evaluate the proposed GS1MC and its extension CDMC respectively. The experiments are based on simulation analysis as well as benchmark comparison with the benchmark methods.

A. Experiments on GS1MC

1) Simulation Analysis: To start with, to verify the effectiveness of GS1MC, a synthetic dataset with group information was designed in the following way. Firstly, we set $n_1 = 200$, $n_2 = 800$, $m_1 = 10$ and $m_2 = 10$. Then we generate $\hat{\mathbf{P}} \sim N(0, \mathbf{I}_{n_1 \times K})$ and $\hat{\mathbf{Q}} \sim N(0, \mathbf{I}_{n_2 \times K})$, where $\mathbf{I}_{m \times K}$ is an $(m \times K) \times (m \times K)$-order identity matrix. To include

Algorithm 1 Cluster Developing 1-bit Matrix Completion

1 **procedure** CDMC
2 Randomly initialize user/item groups.
3 Update latent variables $\mathbf{P}, \mathbf{S}_U, \mathbf{Q}, \mathbf{T}_J$ by (13) to (16).
4 *loop*:
5 Construct $\mathbf{M}$ Matrix by (9).
6 Build adjacency matrices $\mathbf{W}_1$, $\mathbf{W}_2$ by $\mathbf{W}_l = |\mathbf{C}_l| + |\mathbf{C}_l^T|, l = \{1, 2\}$ (Elhamifar and Vidal, 2013).
7 Apply spectral clustering (Ng et al., 2002) on $\mathbf{W}_1$ and $\mathbf{W}_2$.
8 Update cluster identities.
9 Update latent variables $\mathbf{P}, \mathbf{S}_U, \mathbf{Q}, \mathbf{T}_J$ by (26) to (29) in a smaller inner loop.
10 If not converged, **goto** Step 4: *loop*.

TABLE II: Synthetic Dataset: The relative error is computed by: $\frac{\|\mathbf{M}-\hat{\mathbf{M}}\|_F^2}{\|\hat{\mathbf{M}}\|_F^2}$.

Dimension	Observ. Rate: π	10%	15%	20%	25%
K = 3	GS1MC	**1.00**	**0.85**	**0.78**	**0.73**
	Trace-Norm	1.89	1.74	1.67	1.59
K = 6	GS1MC	**1.00**	**0.92**	**0.81**	**0.74**
	Trace-Norm	2.53	2.27	2.15	2.02
K = 10	GS1MC	**0.95**	**0.91**	**0.90**	**0.89**
	Trace-Norm	2.16	1.69	1.46	1.31
K = 20	GS1MC	**0.95**	**0.94**	**0.93**	**0.93**
	Trace-Norm	3.20	2.62	2.34	2.19
K = 50	GS1MC	**0.98**	**0.97**	**0.97**	**0.97**
	Trace-Norm	5.65	5.10	4.67	4.32

the group information, we taks $\hat{\mathbf{S}}_U = (\hat{\mathbf{s}}_v)$ and $\hat{\mathbf{T}}_J = (\hat{\mathbf{t}}_j)$, where $\hat{\mathbf{s}}_v \sim N(-2 + 0.4v, \mathbf{1}_K)$, $\hat{\mathbf{t}}_j \sim N(-3 + 0.6j, \mathbf{1}_K)$, $v \in \{1, ..., m_1\}$ and $j \in \{1, ..., m_2\}$. Then, we construct the latent variable matrix by $\hat{\mathbf{M}} = (\hat{\mathbf{P}} + \mathbf{I}_U^T \hat{\mathbf{S}}_U)(\hat{\mathbf{Q}} + \mathbf{I}_J^T \hat{\mathbf{T}}_J)$ and scale it so that $\|\hat{\mathbf{M}}\|_\infty = 1$. Now, we take the 1-bit transformation and add the noise by $f(\hat{\mathbf{M}}) + N(0, \mathbf{I}_{n_1 \times n_2})$. We keep a certain percentage π of entries as observations.

We set K from small to large and randomly split the data in terms of different training size, namely $\pi = \{25\%, 20\%, 15\%, 10\%\}$. we assume the right group identities are preliminary and compare our method with the baseline (Davenport et al., 2014). The results for both methods, shown in Table II, are taken as the average among 10 replications of cross-validation. It is indicated that GS1MC has a much more robust performance compared to traditional 1-bit matrix completion, especially when the observed data is sparse (*cold-start problem*) or when the latent variable have higher dimensions.

2) Benchmark Comparison: We compare GS1MC on Jester (Goldberg et al., 2001) and Movielens $100k$ (Harper and Konstan, 2016) with (a) the trace-norm frequentist logistic model (Trace-norm) (Davenport et al., 2014); (b) the exact low-rank model (Exact-rank) (Bhaskar and Javanmard, 2015); and (c) a max-norm constrained minimization approach (Max-norm) (Cai and Zhou, 2013). We show the best results appeared in previous literature without augmentation. Movielens $100k$ dataset is split into different training-test size.

TABLE III: Benchmark comparison on Movielens-100k and Jester dataset.

	% Prediction accuracy			
	Max-norm	Trace-norm	Exact-rank	GS1MC
95% Mov.	71.5 ± 0.7	72.4 ± 0.6	72.3 ± 0.7	**73.8 ± 0.2**
10% Mov.	58.4 ± 0.6	58.5 ± 0.5	60.4 ± 0.6	**65.7 ± 0.2**
5% Mov.	50.3 ± 0.2	49.2 ± 0.7	53.7 ± 0.8	**62.6 ± 0.3**
jester-1	-	71.3 ± 0.5	71.0 ± 0.2	**71.7 ± 0.4**
jester-2	-	72.4 ± 0.2	71.5 ± 0.4	**72.5 ± 0.3**
jester-3	-	**72.5 ± 0.6**	70.8 ± 0.4	71.7 ± 0.6

Since the fact that the cluster information of most user-item interaction data is not available, to provide GS1MC cluster information, we group the original dataset according to their implicit feedback. Implicit feedback refers to the frequency of items receiving comments or users giving feedback. It only concerns about the identity of ratings irrespective of actual rating values (Devooght et al., 2015). It is expected that people giving more ratings tend to be more curmudgeon while items with more feedback tend to have higher average ratings (Bi et al., 2017). Thus, we group users and items according to the number of ratings they have given or received.

The converged accuracy results are displayed in Table III. The proposed method outperforms most of the baselines. Conspicuously, regarding the scenario when the training size is extremely small (5%), our method has greatly boosted traditional binary matrix completion method by utilizing the group information.

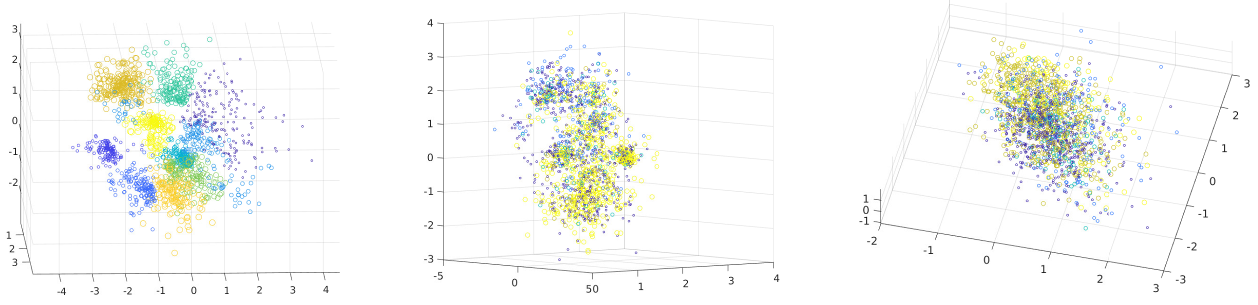
B. Experiments on CDMC

1) Quantitative Comparison: To evaluate the performance of CDMC, we compare it with various settings of GS1BC and use exact low-rank 1-Bit matrix completion (Bhaskar and Javanmard, 2015) as the baseline on Movielens-100K dataset. Approaches include:

- CDMC: Randomly initialize group identities and learn clusters along with matrix completion.
- GS1MC+IF: Use implicit feedback as group identities.
- GS1MC+KM: Fill missing entries with zero and use result from k-means as group identities.
- GS1MC+SSC: Fill missing entries with zero and use result from sparse subspace clustering (Elhamifar and Vidal, 2013) as group identities.
- Exact-rank: Exact-rank 1-Bit matrix completion (Bhaskar and Javanmard, 2015).

We split Movielens-100K dataset with different training sizes. Then, we randomly initialize group identities of users/items for CDMC to start. The method is validated with criteria: (1) *clustering error*; and (2) *completion error* (Elhamifar, 2016). We compute:

$$\text{Clustering Error} = \frac{\#\text{Missclassified entries}}{\#\text{test entries}}, \qquad (31)$$

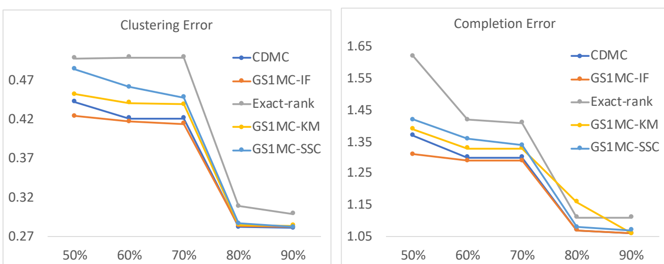


(a) CDMC: Item clusters. (b) Genre identities on CDMC embeddings. (c) Genre identities on Exact-rank embeddings.

Fig. 1: (a) Item clustering results of CDMC. (b) Using $(\mathbf{Q} + \mathbf{I}_J^T \mathbf{T}_J)$ from CDMC as coordinates, products' genre identities are visualized in different colors. (c) Using embeddings from Exact-rank matrix completion (Bhaskar and Javanmard, 2015) as coordinates, products' genre identities are visualized in different colors.

and

$$\text{Completion Error} = \frac{\|\hat{\mathbf{Y}} - \mathbf{Y}\|}{\mathbf{Y}}. \tag{32}$$



(a) Clustering Error (b) Completion Error

Fig. 2: Performance comparison between CDMC and different GS1MC settings with exact-rank 1-bit matrix completion (Bhaskar and Javanmard, 2015) as the baseline.

The result is shown in Figure 2. As shown, GS1MC+IF with implicit feedback identities achieves the best performance while CDMC without knowing group identities ranks the second. Meanwhile, it is noteworthy that all methods considering group effects, including CDMC, GS1MC+IF, GS1MC+KM and GS1MC+SSC, have much more robust performance compared to conventional method such as exact-rank 1-bit matrix completion.

To notice, this experiment also demonstrates the second usage scenario of GS1BC. Since in realistic situation, most clustering problems may not have the ground-truth, one could use GS1MC to validate the clustering quality from a quantitative perspective, i.e. in respect of their completion errors.

2) Insight Generation: To validate the practical influence of CDMC, we project the actual profile features of Movieslens-100k dataset onto the latent variable CDMC learned and discovered some noteworthy findings.

In Movielens-100K dataset, movies are labeled with 19 categories, and each movie can be labeled as multiple genres. We construct a genre matrix $\mathbf{A} = (A_{ig}) \in \mathbb{R}^{n_2 \times 19}$, here $A_{ig} = 1$ means item-i can be classified in category-g. As items in $\mathbf{A}$ share the 1-to-1 exact same index with $(\mathbf{Q} + \mathbf{I}_J^T \mathbf{T}_J)$, we apply k-means clustering method on this generic information and visualize its results corresponding to the latent variable that CDMC learned.

We consider using three-dimensional embedding space for CDMC. Latent variable $(\mathbf{Q} + \mathbf{I}_J^T \mathbf{T}_J)$ are constructed as coordinates for each entity. The genre clusters learned from products profile are visualized by different colors in Figure 1b. To evaluate the performance, we take exact low-rank 1-bit matrix completion as the baseline. Similarly, we construct learned latent variables as coordinates and visualize each entity regarding to its genre identity. The results for exact low-rank approach are shown in Figure 1c.

As shown, one could notice that the genre identities have a more discernible pattern on learned latent variable $(\mathbf{Q} + \mathbf{I}_J^T \mathbf{T}_J)$ compare to the conventional method. In other words, even though the fact that our proposed CDMC method did not take any generic information, it has captured items' factual profile with the sparse binary matrix. Besides, as CDMC conducts subspace clustering and group-specific matrix completion in an iterative manner, along with gradually learning the hidden profiles, the model can integrate this information immediately into matrix completion task.

VI. Conclusion

In this paper, we introduce group-specific matrix factorization into 1-bit matrix completion and proposed GS1MC. Then we integrate subspace clustering with matrix completion task and proposed CDMC. Instead of receiving pre-known cluster information, CDMC learn cluster identities during matrix completion and utilize their group effects. Experiments demonstrate that GS1MC outperforms conventional 1-bit methods on both synthetic and real-world data, especially for the *cold-start problem*, and CDMC successfully captures items' hidden generic features from the sparse 1-bit rating matrix. Notably, GS1MC can serve as a quantitative protocol to compare the efficacy of different clustering methods while CDMC is an insightful tool to explore unrevealed social phenomena.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

References

Bell, R., Y. Koren, and C. Volinsky
2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. *Proceedings of the 13th ACM SIGKDD*.

Bhaskar, S. and A. Javanmard
2015. 1-bit matrix completion under exact low-rank constraint. *arXiv:1502.06689*.

Bi, X., A. Qu, J. Wang, and X. Shen
2017. A group-specific recommender system. *Journal of the American Statistical Association*.

Billsus, D. and M. J. Pazzani
2000. User modeling for adaptive news access. *UMUAI*.

Brbić, M. and I. Kopriva
2018. l0 motivated low-rank sparse subspace clustering. *IEEE Transactions on Cybernetics*.

Breese, J. S., D. Heckerman, and C. Kadie
1998. Empirical analysis of predictive algorithms for collaborative filtering. *UAI*.

Cai, T. and W.-X. Zhou
2013. A max-norm constrained minimization approach to 1-bit matrix completion. *JMLR*.

Davenport, M. A., Y. Plan, E. Van Den Berg, and M. Wootters
2014. 1-bit matrix completion. *Information and Inference: A Journal of the IMA*.

Deshpande, M. and G. Karypis
2004. Item-based top-n recommendation algorithms. *ACM TOIS*.

Devooght, R., N. Kourtellis, and A. Mantrach
2015. Dynamic matrix factorization with priors on unknown values. *Proceedings of the 21th ACM SIGKDD*.

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

Elhamifar, E. and R. Vidal
2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TPAMI*.

Goldberg, K., T. Roeder, D. Gupta, and C. Perkins
2001. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*.

Grčar, M., B. Fortuna, D. Mladenič, and M. Grobelnik
2006. KNN versus SVM in the collaborative filtering framework. Springer.

Harper, F. M. and J. A. Konstan
2016. The movielens datasets: History and context. *ACM TIIS*.

Koren, Y.
2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD*.

Koren, Y. and R. Bell
2015. Advances in collaborative filtering. Springer.

Kotler, P.
2009. *Marketing management: A south Asian perspective*. Pearson Education India.

Linden, G., B. Smith, and J. York
2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*.

Liu, G., Z. Lin, and Y. Yu
2010. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, Pp. 663–670.

Ng, A. Y., M. I. Jordan, and Y. Weiss
2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*.

Paterek, A.
2007. Improving regularized singular value decomposition for collaborative filtering. *Proceedings of KDD Cup and Workshop*.

Pazzani, M. and D. Billsus
1997. Learning and revising user profiles: The identification of interesting web sites. *ML*.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl
1994. Grouplens: an open architecture for collaborative filtering of netnews. *Proceedings of the 1994 ACM CSCW*.

Salakhutdinov, R., A. Mnih, and G. Hinton
2007. Restricted Boltzmann machines for collaborative filtering. *Proceedings of the 24th ICML*.

Sarwar, B. M. et al.
2001. Item-based collaborative filtering recommendation algorithms. *WWW*.

Shoham, Y.
1997. Combining content-based and collaborative recommendation. *Communications of the ACM*.

Takács, G., I. Pilászy, B. Németh, and D. Tikk
2008. Investigation of various matrix factorization methods for large recommender systems.

Takács, G., I. Pilászy, B. Németh, and D. Tikk
2009. Scalable collaborative filtering approaches for large recommender systems. *JMLR*.

Vidal, R. and P. Favaro
2014. Low rank subspace clustering. *Pattern Recognition Letters*.

Zhu, Y., X. Shen, and C. Ye
2016. Personalized prediction and sparsity pursuit in latent factor models. *Journal of the American Statistical Association*.

Zitnick, C. L. and T. Kanade
2004. Maximum entropy for collaborative filtering. *Proceedings of the 20th Conference on UAI*.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE