An Empirical Study of the Discreteness Prior in Low-Rank Matrix Completion

Rodrigo Alves[1]* Antoine Ledent[1]* Renato Assunção[2] Marius Kloft[1]

[1] Department of Computer Science, TU Kaiserslautern, Germany
[2] Department of Computer Science, UFMG, Brazil
{alves,ledent,kloft}@cs.uni-kl.de, assuncao@dcc.ufmg.br
*The first two authors contributed equally

Abstract

A reasonable assumption in recommender systems is that the rows (users) and columns (items) of the rating matrix can be split into groups (communities) with the following property: each entry of the matrix is the sum of components corresponding to community behavior and a purely low-rank component corresponding to individual behavior. We propose to investigate (1) whether such a structure is present in real-world datasets, (2) whether the knowledge of the existence of such structure alone can improve performance, without explicit information about the community memberships. To these ends, we formulate a *joint* optimization problem over all (completed matrix, set of communities) pairs based on a nuclear-norm regularizer which jointly encourages *both* low-rank solutions *and* the recovery of relevant communities. Since our optimization problem is non-convex and of combinatorial complexity, we propose a heuristic algorithm to solve it. Our algorithm alternatingly refines the user and item communities through a clustering step jointly supervised by nuclear-norm regularization. The algorithm is guaranteed to converge. We propose some synthetic data experiments to confirm or disprove our hypothesis and evaluate the efficacy of our method at recovering the relevant communities.

1 Introduction

In recommender systems (RSs) we aim to recommend items (e.g. movies, products, books) to users. Oftentimes information about users (resp. items) is available in the form of categorical attributes (commonly referred to as communities) such as gender, nationality, or occupation (resp. genres, brands, or authors) [1, 2, 3, 4]. Such information is frequently used, for instance, to improve RSs' performance in terms of accuracy enhancement [4], interpretability [5, 6], and scalability [7].

However, user and item communities are often not explicitly available. A typical solution to this problem is to apply a clustering method on other forms of (non-categorical) side information. For instance, users are often clusterized considering user-user interactions [8, 9, 10, 11, 12]. Another research direction is concerned with providing partitions of the users and items into clusters *based on a rating matrix alone*. A simple solution is to apply a clustering method to the user and/or item profiles obtained as a natural byproduct of the rating prediction process of most collaborative filtering models [13, 14, 15]. However, such methods are ad hoc post-processing steps and do not exploit the cluster structure in the predictions themselves.

Previous attempts at detecting user and item clusters based purely on a low-rank partially observed matrix assume noisily observed pure community behaviour [9, 8]. On the other hand, our hypothesis is that community behaviour and continuous low-rank structure can *coexist* in the same matrix. To

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

confirm or disprove this hypothesis, we aim to perform community discovery and low-rank matrix completion *jointly*, by constructing a model which efficiently exploits the "discreteness prior" on the existence of underlying user and item communities which play a role in the generation of the ratings.

We assume the rows (users) and columns (items) of the matrix can be split into groups (communities) with the property that each entry of the matrix is *a sum of components* corresponding to community behaviour and a purely low-rank component corresponding to individual behaviour. Such a decomposition was first introduced in [6], where an algorithm is provided to perform matrix completion based on this assumption, *assuming complete knowledge of the communities* of users and items. In contrast, we formulate an optimization problem *over* all (completed matrix, set of communities) *pairs* based on a nuclear-norm regularizer which jointly encourages *both* low-rank solutions *and* the recovery of 'relevant' communities. Since our optimization problem is non-convex and of combinatorial complexity, we propose a heuristic algorithm to solve it.

Our experiments will address the following questions:

- Is it conceivable for the prior knowledge of the existence of communities, as opposed to a more general low-rank prior, to improve the performance of matrix completion, *without any explicit knowledge of the community membership function*? Specifically, on synthetic data, how does our method perform in comparison with baselines that can be found in literature?
- Do real datasets (e.g. MovieLens) exhibit a non-trivial combination of discrete (community) behaviour and continuous (generic low-rank) behaviour?
- In real RSs datasets, are the groups recovered by our methods meaningful and interpretable?

The presence of the predicted behaviour can be confirmed or disproved by evaluating whether our methods (which model both phenomena) outperform baselines which do not allow for such phenomena. Categorical information is easier to interpret than generic low-rank features: it can be compared with known groups (e.g. genres), or more finely investigated (one can, e.g., search for common plot themes). If confirmed, our hypothesis could shed light on the underlying phenomena driving recommender systems predictions, greatly improving explainability.

2 Methodology and experimental design

Notation: Let $R \in \mathbb{R}^{m\times n}$ be a partially observed matrix. We denote by $\Omega \subset \{1,2,\ldots,m\} \times \{1,2,\ldots,n\}$ the set of observed entries and R_Ω the matrix of observed entries with zeros imputed in the missing entries. For all $i \le m$ (resp. $j \le n$), write $f(i)$ (resp. $g(j)$) for the community to which i (resp. j) belongs. Denote by d_1 (resp. d_2) the number of user (resp. item) communities. Thus, f (resp. g) are functions from $\{1,2,\ldots,m\}$ (resp. $\{1,2,\ldots,n\}$) to $\{1.2.\ldots,d_1\}$ (resp. $\{1.2.\ldots,d_2\}$). By abuse of notation, we will identify each element of u (resp. v) in $\{1.2.\ldots,d_1\}$ (resp. $\{1.2.\ldots,d_2\}$) with the community $f^{-1}(u) \subset \{1,2,\ldots,m\}$ (resp. $g^{-1}(v) \subset \{1,2,\ldots,n\}$) it represents.

Optimization problem: We propose the following optimization problem:

$$\min_{f,g}\min_{C,M,U,Z}\mathcal{L} \quad \text{with} \quad \mathcal{L} = \sum_{(i,j)\in\Omega} |C_{f(i),g(j)} + M_{i,g(j)} + U_{f(i),j} + Z_{i,j} - R_{i,j}|^2 + \lambda_C\|C\|_* + \lambda_{MU}\left[\|M\|_* + \|U\|_*\right] + \lambda_Z\|Z\|_*, \tag{1}$$

subject to

$$\sum_{i\in f^{-1}(u)} M_{i,v} = 0 \quad \forall u \le d_1, v \le d_2, \qquad \sum_{j\in g^{-1}(v)} U_{u,j} = 0 \quad \forall u \le d_1, v \le d_2,$$
$$\sum_{i\in f^{-1}(u)} Z_{i,j} = 0 \quad \forall j \le n, \quad \text{and} \quad \sum_{j\in g^{-1}(v)} Z_{i,j} = 0 \quad \forall i \le m. \tag{2}$$

Here, λ_C, λ_{MU} and λ_Z are regularization parameters. The conditions (2) imply that the matrix Z is free of any community-wide behaviour component for either users and items, and the matrices $M \in \mathbb{R}^{m\times d_2}$ and $U \in \mathbb{R}^{d_1\times n}$ are free of any community-wide behaviour components for the users and items respectively.

Note that the optimization is over not only the matrices C, M, U and Z, but also over the choice of communities f, g. In the case where the community side information is fixed in advance, an equivalent problem has been formulated in [6], where an iterative imputation algorithm is proposed together with a proof of convergence.

Algorithm: Since (1) involves optimization over a combinatorial number of possible functions f, g we propose a heuristic algorithm to reach a solution. Like (1), our algorithm takes as input the partially observed matrix R and the hyperparameters d_1, d_2 and $\Lambda = \{\lambda_Z, \lambda_C, \lambda_{MU}\}$. Our strategy, further represented in Algorithm 1, is as follows.

First, we solve the optimization problem (1) for $f = g = \mathrm{null}$ (which is equivalent to $d_1 = d_2 = 0$). Secondly, we cluster both the rows and the columns of the recovered matrix, with the numbers of clusters set to d_1 and d_2, yielding the partitions f_0 and g_0 respectively. Our next aim is now to iteratively refine the partitions f and g. To this end, we solve problem (1) with $f = f_0, g = g_0$ fixed, obtaining the matrices $\hat{R}_0 = \{C_0, M_0, U_0, Z_0\}$, and consider, for each set of non-negative parameters $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$ in some predetermined set Θ, the following cluster profile:

$$S^\theta = \theta_1 \tilde{C}_0 + \theta_2 \tilde{M}_0 + \theta_3 \tilde{U}_0 + \theta_4 Z_0, \tag{3}$$

where $\tilde{C}$, $\tilde{M}$ and $\tilde{U}$ are $m \times n$ matrices such that $\tilde{C}_{i,j} = C_{f(i),g(j)} \forall i \leq m, j \leq n$, $\tilde{M}_{i,j} = M_{i,g(j)} \forall i \leq m, j \leq n$, and $\tilde{U}_{i,j} = U_{f(i),j} \forall i \leq m, j \leq n$[1]. For each $\theta \in \Theta$ we now obtain partitions f_θ (resp. g_θ) of the users (resp. items) by clustering the rows (resp. columns) of S^θ. Next we solve (1) fixing $f = f_\theta, g = g_\theta$, obtaining the matrices $\hat{R}_\theta = \{C_\theta, M_\theta, U_\theta, Z_\theta\}$ and calculate $\ell_\theta = \mathcal{L}(R_\Omega, \hat{R}_\theta, \Lambda, f_\theta, g_\theta)$. Finally, we compute the minimum $\ell_{\theta_{\min}}$ of ℓ_θ over all values of θ and retain the partitions $f_{\theta_{\min}}, g_{\theta_{\min}}$ and the associated matrices $\hat{R}_{\theta_{\min}} = \{C_{\theta_{\min}}, M_{\theta_{\min}}, U_{\theta_{\min}}, Z_{\theta_{\min}}\}$. Next we can feed this data to the next iteration of the algorithm: we use $\hat{R}_{\theta_{\min}}$ to build the matrices S^θs again and continue the process until convergence.

Regarding the choice of the searched set Θ, since we use the k-means algorithm as the clustering procedure we can restrict ourselves to θs such that $\theta_1 + \theta_2 + \theta_3 + \theta_4 = 1$, and for computational reasons, we set Θ to be the intersection of that set with a given discrete grid. Note that the value $\theta = (1, 0, 0, 0) \in \Theta$ will always return the same clustering and the same loss as the previous iteration. Thus, the loss is guaranteed to decrease monotonically at each iteration and the algorithm converges.

Remark 1: The motivation for using k-means as the background clustering procedure is that it can be interpreted as a well-principled approximation to the optimization of the loss $\mathcal{L}$ over the user (item) cluster assignments: assume for simplicity that there are no clusters over items so that the matrix is only composed of the terms C and Z, with $\Lambda_C = 0$. For any clustering assignment over the users, the rows of the matrix Z are the distances to the cluster centers. Minimizing the nuclear norm of Z over the choice of assignments is very difficult due to the implicit (cross-cluster) low-rank condition. However, if we instead consider the Frobenius norm (at small cost to the intuition), the solution is given exactly by k-means.

Remark 2: The intuition behind the introduction of the heuristic search parameter θ and the construction (3) of S^θ is as follows. If λ_Z and λ_{MU} are both very large[2], and the item partition g is correct, it is clearly best to cluster the rows of $\tilde{M} + \tilde{C}$. Indeed, the items only exhibit community behaviour in those components. On the other hand, if the ground truth contains a large $\tilde{U}$ component (i.e. if there is significant interaction between user communities and specific items), or if the current item partition g is significantly wrong, then the component $Z + \tilde{U}$ will be more relevant to the clustering problem. We further split all components so we can look for solutions across a spectrum of confidence in the current partition (a very large θ_4 will reset the optimization procedure to a distant solution, whilst a large θ_1 will keep the current solution unchanged). Thus our algorithm includes a mix of incremental steps and explorative search.

Synthetic data generation: To examine our proposed method in different regimes we aim to generate square matrices in $\mathbb{R}^{m \times m}$ where the users and items are naturally divided into k clusters of size m/k.

[1] Note that since the matrices $\tilde{C}, \tilde{M}, \tilde{U}$ and Z live in mutually orthogonal subspaces with respect to the Frobenius inner product, the matrices C, M, U, Z (and in particular the loss $\mathcal{L}$) are well-defined for any full matrix $R = \tilde{C} + \tilde{M} + \tilde{U} + Z$ for any given set of hyperparameters and partitions f, g.

[2] This implies, assuming suitably cross-validated parameters, that the Z, M, U components of the ground truth matrix are very small.

Algorithm 1 Collaborative Clustering

INPUT: Partially observed matrix R_Ω and hyperparameters d_1,d_2, $\Lambda = \{\lambda_Z, \lambda_C, \lambda_{MU}\}$

```
$f = \text{null}, g = \text{null}$
$Z = \arg\min_Z \mathcal{L}(R_\Omega, \Lambda, f, g)$
$f_0 = \text{clusterRows}(Z, d_1)$, $g_0 = \text{clusterColumns}(Z, d_2)$
$\hat{R}_0 = \{C_0, M_0, U_0, Z_0\} = \arg\min_{C,M,U,Z} \mathcal{L}(R_\Omega, \Lambda, f_0, g_0)$
repeat
    MAKE $\tilde{C}_0, \tilde{M}_0, \tilde{U}_0$ FROM $\hat{R}_0, f_0, g_0$
    $f = f_0, g = g_0, \ell_0 = \mathcal{L}(R_\Omega, \hat{R}_0, \Lambda, f_0, g_0)$
    for $\theta \in \Theta$ do
        $S^\theta = \theta_1\tilde{C} + \theta_2\tilde{M} + \theta_3\tilde{U} + \theta_4 Z$
        $f_\theta = \text{clusterRows}(S^\theta, d_1)$, $g_\theta = \text{clusterColumns}(S^\theta, d_2)$
        $\hat{R}_\theta = \{C_\theta, M_\theta, U_\theta, Z_\theta\} = \arg\min_{C,M,U,Z} \mathcal{L}(R_\Omega, \Lambda, f_\theta, g_\theta)$
        $\ell_\theta = \mathcal{L}(R_\Omega, \hat{R}_\theta, \Lambda, f_\theta, g_\theta)$
    end for
    $\theta_{\min} = \arg\min_\theta(\ell_\theta)$
    $\hat{R}_0 = \hat{R}_{\theta_{\min}}, f_0 = f_{\theta_{\min}}, g_0 = g_{\theta_{\min}}$
until $f_0 == f$ and $g_0 == g$
return $f$,$g$
```

Without loss of generality, the first cluster consists of the first m/k entries, etc. and we assume f, g are defined according to this clustering arrangement. In our first strand of experiments, a wide range of ground truth matrices $\mathbb{R}^{m\times m}$ will be built from the following three basis matrices:

- **[Pure community component]** (A): First construct a random orthogonal $k \times k$ matrix $\bar{A}$ to represent the cross-community affinities, then set $U_{i,j} = \bar{A}_{f(i),g(j)}$ and set A to be a normalized version of U of Frobenius norm m.
- **[User × (Item community) and vice versa]** (B): Construct two matrices $\tilde{B}^1 \in \mathbb{R}^{m\times k}$ (and $\tilde{B}^2 \in \mathbb{R}^{k\times m}$) whose columns (resp. rows) are k random orthonormal vectors in $\{x \in \mathbb{R}^m : \sum_{i\in f^{-1}(c)} x_i = 0 \forall c \in \{1, 2, \dots, k\}\}$ such that for each $c \in \{1, 2, \dots, k\}$, the columns vectors $\tilde{B}^1_{f^{-1}(c),j}$ for $j \leq k$ are orthonormal (similarly for $\tilde{B}^2$). Set $U_{i,j} = \bar{B}^1_{i,g(j)} + \bar{B}^2_{f(i),j}$ and let B be a normalised version of U with Frobenius norm m.
- **[Community-free behaviour]** (C): For each $c_1, c_2 \in \{1, 2, \dots, k\}$, (independently) generate a random matrix $U^{c_1,c_2} \in \mathbb{R}^{m/k\times(m/k-1)}$ whose columns form an orthonormal basis of the space $\{x \in \mathbb{R}^{m/k} : \sum_i x_i = 1\}$. Then construct the matrix $\mathbb{R}^{m/k\times m/k} \ni C^{c_1,c_2} = U^{c_1,c_2}(U^{c_1,c_2})^\top$ (for each c_1, c_2). Define $\bar{C} \in \mathbb{R}^{m\times m}$ as a block $k \times k$ matrix whose blocks are the matrices C^{c_1,c_2}. Finally, C is a normalized version of $\bar{C}$ with Froebenius norm m.

Note that for a given f, g, the matrices A, B, C belong to the (independent) subspaces corresponding to $\tilde{C}$, ($\tilde{M} + \tilde{U}$) and Z respectively. Using these basis matrices, we can construct matrices of the form:

$$R(\alpha, \beta) := A + \alpha B + \beta C, \tag{4}$$

where the parameters α and β regulate the importance of ground truth behaviours associated to A, B and C. We plan to run experiments varying α and β as well as the proportion of observed entries of $R(\alpha, \beta)$ and observe how our method performs in different difficulty regimes. Note that the orthogonality conditions we imposed in the specific construction above make the problem especially well-behaved: in the ground truth solution, all clusters of both users and items have equidistant centers, and all of the vectors in any given cluster are equidistant to each other and each is at the same distance from the center. This means no cluster is easier to detect than any other.

In our second strand of synthetic experiments, we will verify that the proposed method performs well in a slightly less contrived setting without the orthogonality constraints presented above. Specifically:

- The pure community component $\tilde{A}$ will be constructed as a $k \times k$ matrix with i.i.d. $N(0, 1)$ entries. The (normalised) matrix A will be constructed from $\tilde{A}$ as before.

- The columns of the user × community raw matrix $\tilde{B}^1 \in \mathbb{R}^{m\times k}$ are projections of independent isotropic Gaussian vectors in $\mathbb{R}^m$ onto the space $\{x \in \mathbb{R}^m : \sum_{i\in f^{-1}(c)} x_i = 0 \forall c \in \{1, 2, \ldots, k\}\}$. B^2 is constructed similarly. Further normalisation steps are unchanged.
- The matrix C corresponding to pure low-rank effects, is simply constructed with i.i.d. $N(0, 1)$ entries, then projected to the space $\{X \in \mathbb{R}^{m\times m} : \sum_{i\in f^{-1}(c)} x_{i,j} = 0 \forall c \in \{1, 2, \ldots, k\}, j \leq m \wedge \sum_{j\in g^{-1}(c)} x_{i,j} = 0 \forall c \in \{1, 2, \ldots, k\}, i \leq m\}$ and normalised to have unit Frobenius norm.

In the above situation, it is no longer true that each cluster is equally hard to detect.

Baselines: In the scenario where no explicit side information is provided for users or items, two branches of clustering frameworks are widely used in collaborative filtering-based recommendation systems: (1) matrix factorization (MF) methods and (2) nearest neighbor (NN) methods. We select as baselines a state-of-the-art representative example of each branch as follows:

- **[MF]**: Apply standard nuclear-norm matrix factorization [16] and then cluster the rows (resp. columns) of the recovered matrix to detect communities of users (resp. items).
- **[NN]**: Nearest neighbor methods typically calculate a statistical distance between users (resp. items) using only the known entries, and then group the users (resp. items) hierarchically. As a representative example, we propose to use the Pearson correlation.

Hyperparameter selection and scalability: The relevant hyperparameters in our model are $d_1, d_2, \lambda_Z, \lambda_C, \lambda_{MU}$. In practice, they can later be determined through *cross validation*. Note that the CV procedure can be executed in parallel: different sets of (Λ, d_1, d_2) can be fitted separately. In the case of d_1 and d_2, it is not necessary to run the full algorithm for each combination. Indeed, note that the choice of d_1 and d_2 is likely to have a large effect on the optimal loss for typical values of Λ. Thus, a promising strategy is to run a rudimentary version of our algorithm (e.g. with a single clustering step) for several d_1's and d_2's, and select the best performing values.

Regarding the for loop in Algorithm 1 (lines 8-13) observe that the iteration $i + 1$ does not depend on iteration i. In this case, small adjustments also allow these steps to be executed in parallel, significantly reducing the computing burden of the search for the parameters Θ.

Note that line 11, which requires performing an iterative imputation procedure to solve the version of problem (1) for known f, g, can be greatly accelerated with warm starts: the full recovered matrix from the previous iteration (of the repeat loop) is used as a warm start for each value of Θ, so that only a small number of imputations is required. Similarly to other involved optimization algorithms[3], further improvements can be performed if necessary: for instance, one could initially select the optimal value of Θ based on an even smaller number of imputations, and perform a more thorough imputation procedure on the chosen Θ before moving to the next iteration of the repeat loop.

Evalutation procedure: In the synthetic data, we propose to assess the agreement of our clustering method with the ground truth using the *Rand Index*. Let f_1, f_2 be two partitions of a set $\{1, 2, \ldots, m\}$, the Random index $\mathrm{Rand}(f_1, f_2)$ between f_1 and f_2 is defined as the proportion of pairs of elements in $\{1, 2, \ldots, m\}$ which are either placed in the same cluster in both partitions f_1, f_2 or placed in a different cluster in both partitions f_1, f_2:

$$\mathrm{Rand}(f_1, f_2) = \#(\mathcal{S}_{f_1,f_2}) / \binom{n}{2}, \quad \text{where}$$

$$\mathcal{S}_{f_1,f_2} = (\{i_1, i_2\} : [f_1(i_1) = f_1(i_2) \wedge f_2(i_1) = f_2(i_2)] \vee [f_1(i_1) \neq f_1(i_2) \wedge f_2(i_1) \neq f_2(i_2)]) \,.$$

Note that the random index is well defined even if f_1 and f_2 return a different number of clusters.

Real data experiments: We intend to evaluate the behaviour and performance of our methods on broadly used and stable benchmark datasets such as MovieLens, Douban and LastFM. In the real data experiments, since we do not have access to the "correct" clusters, we can only rely on the following two ad hoc solutions:

- comparing the accuracy (for instance the RMSE) of our method with that of other methods such as a single optimization of Problem (1) with $f = g = \mathrm{null}$; and

[3] such as architecture search for neural networks

- manually observing correlations between our recovered clusters and explicitly or implicitly available categorical side information (such as movie genres or common plot themes).

3 Related work

Community discovery is a widely researched task in recommender systems. In [8], the authors propose a probabilistic model to solve binary matrix completion with graph side information based on the assumption that the users form communities. The clusters are recovered from the graph information via the stochastic block model (SBM), and the cluster preferences are then recovered from the observed data. Similar approaches can be observed in [10, 11, 17, 18, 9]. The main difference between these works and ours is that they do not allow for non-random user-specific behaviour within each cluster (except [9]), that is, there is no difference between predicting the matrix and predicting the clusters. In that respect, our setting is more similar to the regularization based techniques [19, 20, 21], but our method is different. The paper [9] is to our knowledge the only work that incorporates item-specific behaviour in a community detection context. They do so in a discrete fashion with the concept of "atypical" movies and users, whilst our approach is a continuous one, which includes the possibility of representing any matrix (at a regularization cost). A deep learning approach to extracting community information from graphs is offered by graph neural networks [22, 23, 24].

Another systematic work which studies collaborative clustering is [25], which provides a deep theoretical analysis of a model where items must be clusterized based on discrete ratings given by users, themselves belonging to certain communities (here the ratings are iid for any fixed pair of communities and no specific algorithm is presented). In [13, 14], the authors detect user groups applying k-means on the user-latent factor matrix (imputing the unknown entries via collaborative filtering). Nearest neighbor techniques are also employed in aggregation methods: in [26], the authors use the Pearson correlation to define the similarity among the users while in [27], the cosine similarity is applied. One distinguishing characteristic of our model is that it is able to learn both ratings and communities *jointly*. It is important to point out some authors explore orthogonality and factorization to implement clustering in matrices [28, 29, 30]. However, these works differ from ours since they start from a fully-known matrix, and use different methods. The most related work to ours is [6], which studied the case where there the community memberships are known. In contrast to this work, here we study how to recover both communities and the matrix starting only from the incompletely observed matrix.

Acknowledgments

The authors acknowledge support by the Carl-Zeiss Foundation, by the German Research Foundation (DFG) award KL 2698/2-1, and by the Federal Ministry of Science and Education (BMBF) awards 01IS18051A and 031B0770E.

References

[1] Y.-Y. Chen, A.-J. Cheng, and W. H. Hsu, “Travel recommendation by mining people attributes and travel group types from community-contributed photos,” *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1283–1295, 2013.

[2] I. A. Christensen and S. Schiaffino, “Entertainment recommender systems for group of users,” *Expert Systems with Applications*, vol. 38, no. 11, pp. 14 127–14 135, 2011.

[3] S. Seko, M. Motegi, T. Yagi, and S. Muto, “Video content recommendation for group based on viewing history and viewer preference,” in *ITE Technical Report 35.7*. The Institute of Image Information and Television Engineers, 2011, pp. 25–26.

[4] E. Frolov and I. Oseledets, “Hybridsvd: when collaborative information is not enough,” in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 331–339.

[5] S. Dara, C. R. Chowdary, and C. Kumar, “A survey on group recommender systems,” *Journal of Intelligent Information Systems*, pp. 1–25, 2019.

[6] A. Ledent, R. Alves, and M. Kloft, “Orthogonal inductive matrix completion,” *arXiv preprint arXiv:2004.01653*, 2020.

[7] J.-Y. Jiang, P. H. Chen, C.-J. Hsieh, and W. Wang, "Clustering and constructing user coresets to accelerate large-scale top-k recommender systems," in *Proceedings of The Web Conference 2020*, 2020, pp. 2177–2187.

[8] K. Ahn, K. Lee, H. Cha, and C. Suh, "Binary rating estimation with graph side information," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 4272–4283. [Online]. Available: http://papers.nips.cc/paper/7681-binary-rating-estimation-with-graph-side-information.pdf

[9] Qiaosheng, Zhang, V. Y. F. Tan, and C. Suh, "Community Detection and Matrix Completion with Two-Sided Graph Side-Information," *arXiv e-prints*, p. arXiv:1912.04099, Dec. 2019.

[10] E. Abbe, "Community detection and stochastic block models: Recent developments," *Journal of Machine Learning Research*, vol. 18, no. 177, pp. 1–86, 2018. [Online]. Available: http://jmlr.org/papers/v18/16-480.html

[11] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *IEEE Transactions on Information Theory*, vol. 62, no. 1, pp. 471–487, Jan 2016.

[12] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *2013 IEEE 13th International Conference on Data Mining*, Dec 2013, pp. 1151–1156.

[13] J. Shi, B. Wu, and X. Lin, "A latent group model for group recommendation," in *2015 IEEE International conference on mobile services*. IEEE, 2015, pp. 233–238.

[14] L. Boratto, S. Carta, and M. Satta, "Groups identification and individual recommendations in group recommendation algorithms." in *PRSAT@ recsys*, 2010, pp. 27–34.

[15] L. Boratto, S. Carta, and G. Fenu, "Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering," *Future Generation Computer Systems*, vol. 64, pp. 165–174, 2016.

[16] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, p. 2287–2322, Aug. 2010.

[17] E. Abbe and C. Sandon, "Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, Oct 2015, pp. 670–688.

[18] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109 – 137, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0378873383900217

[19] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix Completion on Graphs," *arXiv e-prints*, p. arXiv:1408.1717, Aug. 2014.

[20] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King, "Recommender systems with social regularization," 01 2011, pp. 287–296.

[21] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, ser. RecSys '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 135–142. [Online]. Available: https://doi.org/10.1145/1864708.1864736

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 3844–3852.

[23] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *CoRR*, vol. abs/1901.00596, 2019. [Online]. Available: http://arxiv.org/abs/1901.00596

[24] M. Henaff, J. Bruna, and Y. Lecun, "Deep convolutional networks on graph-structured data," 06 2015.

[25] J. Ok, S.-Y. Yun, A. Proutiere, and R. Mochaourab, "Collaborative clustering: Sample complexity and efficient algorithms," ser. Proceedings of Machine Learning Research, S. Hanneke and L. Reyzin, Eds., vol. 76. Kyoto University, Kyoto, Japan: PMLR, 15–17 Oct 2017, pp. 288–329. [Online]. Available: http://proceedings.mlr.press/v76/ok17a.html

[26] L. Baltrunas, T. Makcinskas, and F. Ricci, “Group recommendations with rank aggregation and collaborative filtering,” in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 119–126.

[27] H.-N. Kim and A. El Saddik, “A stochastic approach to group recommendations in social media systems,” *Information Systems*, vol. 50, pp. 76–93, 2015.

[28] C. Ding, T. Li, W. Peng, and H. Park, “Orthogonal nonnegative matrix t-factorizations for clustering,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 126–135.

[29] J. Liu, C. Wang, J. Gao, and J. Han, “Multi-view clustering via joint nonnegative matrix factorization,” in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 252–260.

[30] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, “Bridging domains with words: Opinion analysis with matrix tri-factorizations,” in *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 2010, pp. 293–302.