Proceedings of Machine Learning Research 148:111–125, 2021 NeurIPS 2020 Preregistration Workshop

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

Rodrigo Alves ALVES@CS.UNI-KL.DE
Antoine Ledent[*] LEDENT@CS.UNI-KL.DE
Department of Computer Science - TU Kaiserslautern / Germany

Renato Assunção ASSUNCAO@DCC.UFMG.BR
Department of Computer Science - UFMG / Brazil

Marius Kloft KLOFT@CS.UNI-KL.DE
Department of Computer Science - TU Kaiserslautern / Germany

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 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 performed synthetic and real data experiments to confirm our hypothesis and evaluate the efficacy of our method at recovering the relevant communities. The results shows that our method is capable of retrieving such an underlying (community behaviour + continuous low-rank) structure with high accuracy if it is present.

Keywords: Cluster detection, Inductive Matrix Completion, Orthogonal constraints

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) Chen et al. (2013); Christensen and Schiaffino (2011); Seko et al. (2011); Frolov and Oseledets (2019). Such information is frequently used, for instance, to improve RSs' performance in terms of accuracy enhancement Frolov and Oseledets (2019), interpretability Dara et al. (2019); Ledent et al. (2020), and scalability Jiang et al. (2020).

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

[*] The first two authors contributed equally

instance, users are often clusterized considering user-user interactions Ahn et al. (2018); Qiaosheng et al. (2019); Abbe (2018); Abbe et al. (2016); Yang et al. (2013). 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 Shi et al. (2015); Boratto et al. (2010, 2016). 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 Qiaosheng et al. (2019); Ahn et al. (2018). On the other hand, our hypothesis is that community behaviour and continuous low-rank structure can *coexist* in the same matrix. To 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 Ledent et al. (2020), 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 $i \leq m$ (resp. $j \leq n$), 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.

2.1. Optimization problem

We propose the following optimization problem:

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

subject to

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

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 Ledent et al. (2020), where an iterative imputation algorithm is proposed together with a proof of convergence.

2.2. 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 $f = g = \text{null}$ (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:

$$S^\theta = \theta_1 \tilde{C_0} + \theta_2 \tilde{M_0} + \theta_3 \tilde{U_0} + \theta_4 Z_0, \qquad (3)$$

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

[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.

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.

2.3. 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. 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 $\times$ (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 $\{x \in \mathbb{R}^m :$

[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, $\Lambda = \{\lambda_Z, \lambda_C, \lambda_{MU}\}$

```
$f = \text{null}, g = \text{null}$
$Z = \arg\min_Z \mathcal{L}(R_\Omega, \Lambda, f, g)$
$f_0 =$ clusterRows$(Z, d_1)$, $g_0 =$ 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
        $S^\theta = \theta_1 \tilde{C} + \theta_2 \tilde{M} + \theta_3 \tilde{U} + \theta_4 Z$
        $f_\theta =$ clusterRows$(S^\theta, d_1)$, $g_\theta =$ 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 $f$,$g$
```

$\sum_{i \in f^{-1}(c)} x_i = 0 \forall c \in \{1, 2, \ldots, k\}\}$ such that for each $c \in \{1, 2, \ldots, k\}$, 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 $c_1, c_2 \in \{1, 2, \ldots, k\}$, (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.

2.4. 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 Mazumder et al. (2010) 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.

2.5. 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

[3] such as architecture search for neural networks

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.

2.6. Evaluation 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:

$$\mathrm{rand}(f_1, f_2) = \#(\mathcal{S}_{f_1,f_2})/\begin{pmatrix} n \\ 2 \end{pmatrix}, \quad \text{where}$$

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

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

2.7. 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
- 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 Ahn et al. (2018), 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 Abbe (2018); Abbe et al. (2016); Abbe and Sandon (2015); Holland et al. (1983); Qiaosheng et al. (2019). The main difference between these works and ours is that they do not allow for non-random user-specific behaviour within each cluster (except Qiaosheng et al. (2019)), 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 Kalofolias et al. (2014); Ma et al. (2011); Jamali and Ester (2010), but our method is different. The paper Qiaosheng et al. (2019) 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 Defferrard et al. (2016); Wu et al. (2019); Henaff et al. (2015).

Another systematic work which studies collaborative clustering is Ok et al. (2017), 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 Shi et al. (2015); Boratto et al. (2010), 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 Baltrunas et al. (2010), the authors use the Pearson correlation to define the similarity among the users while in Kim and El Saddik (2015), 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 Ding et al. (2006); Liu et al. (2013); Li et al. (2010). 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 Ledent et al. (2020), 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.

4. Experimental results

We have strictly followed the methodology and the experimental design proposed in Section 2. To describe our experiments' outcomes, we first introduce and discuss the results of the synthetic experiments (Section 2.3). Then we then show how our method performed on a real-world data sets (Section 2.7). Finally, we discuss the results in light of the proposed hypotheses and questions that guided this research (Section 1).

4.1. Synthetic experiments

In Section 2.3 we proposed two generation procedures strands. In both of them, the parameters α and β in (4) regulate the importance of the community components and the community-free component. We varied α and β in $\{0, 0.25, 0.50, 0.75, 1\}$. We also varied the percentage of observed entries $P_\Omega \in \{0.15, 0.30\}$. For both experiments strand and for each each possible combination of α, β and P_Ω, we generated 40 samples of $R(\alpha, \beta) \in \mathbb{R}^{m\times m}$ with users (resp. items) divided into clusters of size m/k, where $m = 100$ and $k = 4$.

All the hyperparameter tuning was done through cross-validation. We split the set of observed entries Ω into two randomly sampled subsets: 95% for training and 5% for validation. The range of the parameters was adjusted according to each model's needs. In this section we refer to our method as **Collaborative Clustering ([CC])**.

Figures 1 and 2 summarize the results of the performed simulations with respect to the accuracy of the detected cluster by [CC] and the baselines [MF] and [NN] for strands 1 and 2, respectively. Since the baselines do not provide a clear way to select the number of clusters, we assume that the value is known, that is, $d_1 = d_2 = 4$. We observe that our method consistently outperforms [MF] and [NN], specially when the importance of the community-free component is low (β is low).

We also investigate whether d_1 and d_2 can be determined through *cross validation* as suggested in Section 2.5. For strand 1 (resp. strand 2), Figure 3 (resp. Figure 4) shows the distribution of the detected number of clusters obtained by our method through *cross validation*. Note that, for both strands, [CC] can often accurately predict the number of clusters, especially when the number of observed entries is sufficient and the community-free component is small.

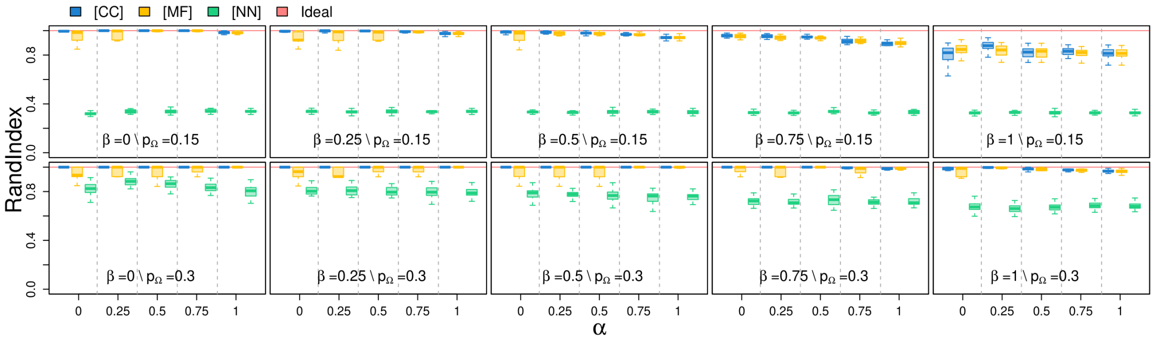

Figure 1: **Strand 1** - Rand index distribution grouped by the parameters α, β and $\mathbb{P}_\Omega$

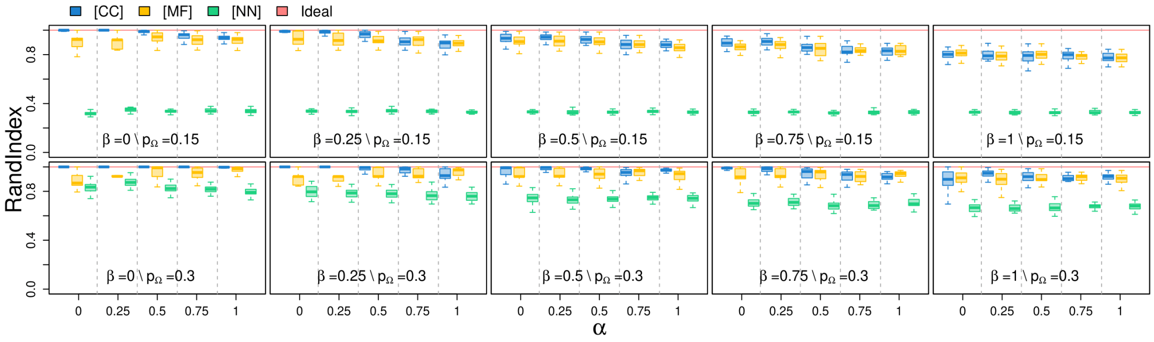

Figure 2: **Strand 2** - Rand index distribution grouped by the parameters α, β and $\mathbb{P}_\Omega$

Finally, we compared the accuracy of our method with [MF][4] that can be seen as single optimization of Problem (1) with $f = g = \text{null}$. Let R_{CC} and R_{MF} be the matrices R recovered through [CC] and [MF] respectively. We proposed quantity $\phi = (||R_{MF} - R||_F - ||R_{CC} - R||_F)/||R||_F$ to make the comparison between the methods. Note that ϕ is normalized by the Frobenious norm of R which forces the matrices to be of similar scale. If ϕ is greater than 0 our method outperforms [MF]. Figures 5 and 6 show, for strands 1 and 2 (respectively), the distribution of ϕ grouped by α, β and P_Ω. We observe that [CC] consistently outperforms [MF] in all scenarios.

4.2. Real-data experiments

To evaluate the behaviour and performance of our methods on broadly used and stable benchmarks we considered the following datasets: Douban, LastFM and MovieLens.

- **Douban**[5] ($R \in \mathbb{R}^{4999 \times 4577}$): Douban is a social network where users can produce content related to movies, music, and events. Douban users are members of the social network and Douban items are a subset of popular movies. The rating range is $[1, 5] \in \mathbb{N}$ and the entry (i, j) corresponds the rating of user i to movie j. The authors collected the movies' genres from the Douban website.

[4] Note that [NN] does not aim to predict the unknown entries but only recover clusters.

[5] Rating matrix available in https://doi.org/10.7910/DVN/JGH1HA

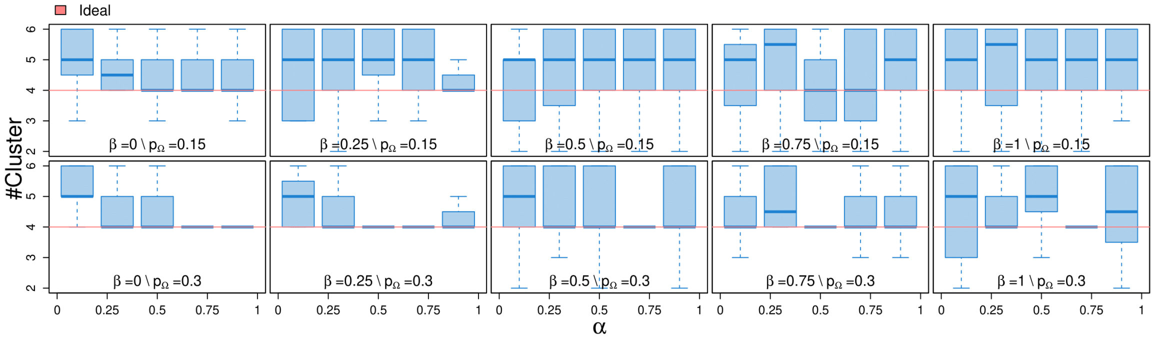

Figure 3: **Strand 1** - The distribution of the number of our model detected clusters according to the parameters α, β and $\mathbb{P}_\Omega$

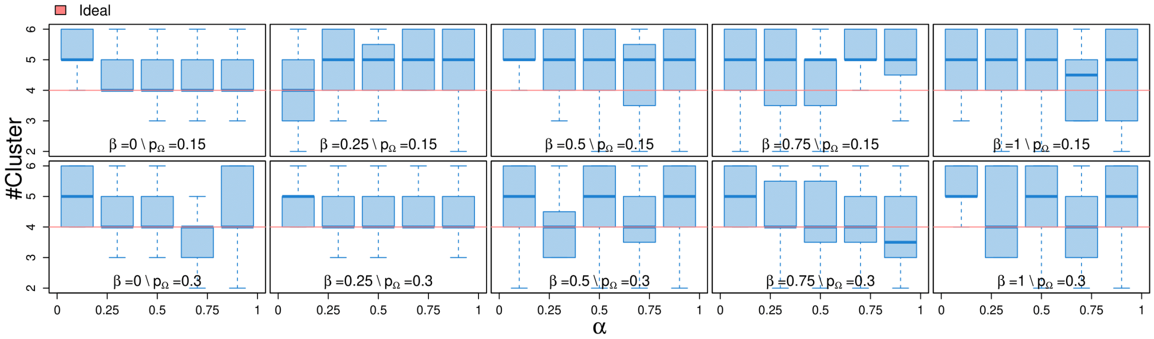

Figure 4: **Strand 2** - The distribution of the number of our model detected clusters according to the parameters α, β and $\mathbb{P}_\Omega$

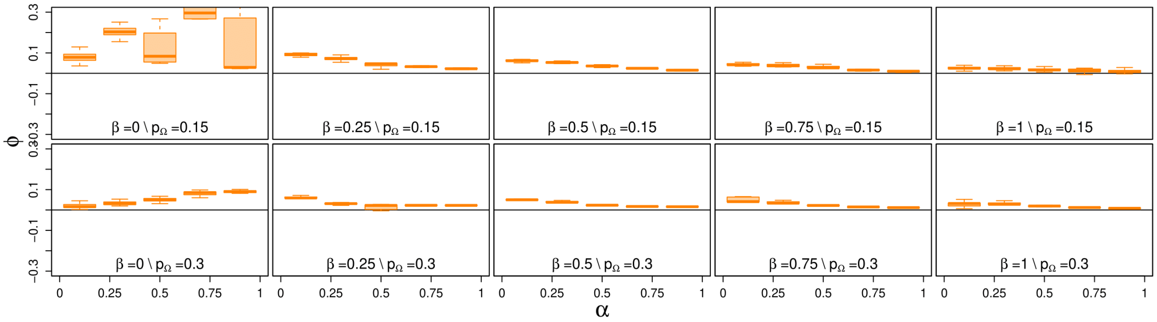

Figure 5: **Strand 1** - ϕ distribution grouped by the parameters α, β and $\mathbb{P}_\Omega$

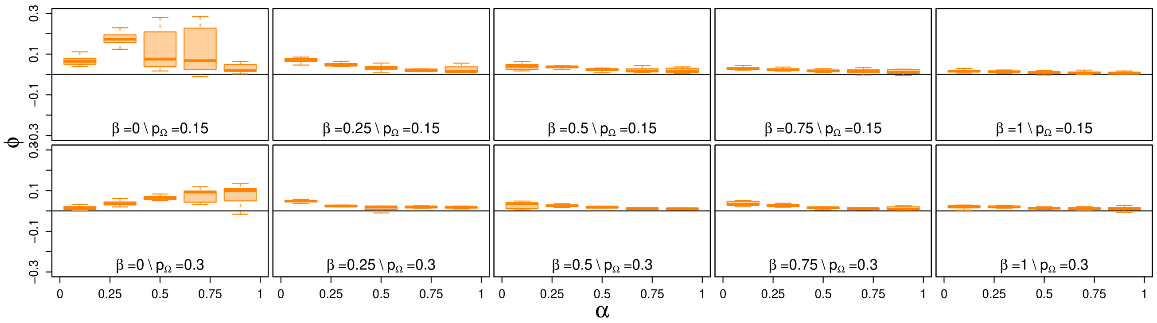

Figure 6: **Strand 2** - ϕ distribution grouped by the parameters α, β and $\mathbb{P}_\Omega$

Table 1: Summary of the results of the real-world data experiments

	Matrix		RMSE		Pearson Correlation		RandIndex[CC]	
	$m \times n$	P_Ω	[CC]	[MF]	[CC]	[MF]	Users	Items
Douban	4999×4577	0.0305	**0.8796**	0.9582	**0.5443**	0.5090	-	0.5548
LastFM	1875×4354	0.0051	**2.1801**	2.4109	**0.5362**	0.5040	-	0.7305
MovieLenz	6040×3382	0.0448	**0.8958**	0.9280	**0.6071**	0.5943	0.7036	0.6080

- **LastFM** ($R \in \mathbb{R}^{1875\times4354}$): Last.fm is a British music website that builds a detailed profile of each user's musical taste based on music recommendation. Differently from the other datasets an entry (i, j) represents the number of views of user i to band/artist j. We expressed the number of views in a log scale. The website allows users to tag artists, which provides us with the opportunity to group the items (artists) by their associated tags. The corresponding (tag-based) groups were then compared to the clusters obtained by our method.

- **MovieLens** ($R \in \mathbb{R}^{6040\times3382}$): We consider the MovieLens 1M dataset, which is a broadly used and stable benchmark dataset. MovieLens is a non-commercial website for movie recommendations. Just as in Douban, an entry (i, j) represents the rate of user i to movie j on a scale from 1 to 5. We selected this version of the dataset because in addition to the rating matrix, cluster information such as age-range for users and genres for movies are available.

As proposed in Section 2.7 we aim to observe correlations between our recovered clusters and explicitly or implicitly available categorical side information. We noticed that some of this categorical side information was highly unbalanced: some categories have fewer than ten items while others have thousands of items. In order to properly assess our method [CC] we keep only items that are associated with categories which have a significant number of instances. Table 1 summarizes the results of the real-world data experiments. Two important (and broadly used) metrics to measure the accuracy in recommender systems are the RMSE and the Spearman correlation (both evaluated between two vectors composed of the values on the test set of the ground-truth matrix R and the estimated matrix $\hat{R}$ respectively) . Observe that [CC] has the lowest RMSE and largest Spearman correlation on all datasets. Note correlations between our recovered clusters and available sets of categorical side information are high, especially the item groups on the LastFM dataset and the users' age-range on the MovieLenz dataset.

4.3. Hypothesis discussion

Our hypothesis is that community behaviour and continuous low-rank structure can *coexist* in the same matrix. The research questions and methodology proposed in Section 1 aim to confirm or disprove this hypothesis.

From the results on synthetic data (Section 4.1), we conclude our proposed method is capable of retrieving such an underlying (community behaviour + continuous low-rank) structure with high accuracy if such structure is present. This also results in a reduction in test error compared to baseline methods which do not exploit the discreteness prior.

In Section 4.2, we showed that on real-life datasets, the use of our model results in a decrease in the RMSE and an increase the Spearman correlation, compared to baselines which do not rely on our core hypothesis. It follows that real-world datasets do, in fact, exhibit a non-trivial combination of discrete (community) behaviour and continuous (generic low-rank) behaviour (although the extent to which that is the case is moderate). Furthermore, the groups recovered by our method are interpretable in the sense that they often correlate with available qualitative categorical side information (even though this information was not fed to the model). This further confirms the relevance of our method and hypotheses.

5. Conclusion

In this paper, we have presented a model which is based on a sum of terms corresponding to community behaviour and continuous low-rank behaviour. A key characteristic of our model is that the community membership functions and the low-rank matrix are optimized jointly, without access to any categorical side information.

On synthetic data experiments, our model was able to disentangle the continuous (generic low rank) and discrete (community-based) components and recover the underlying communities highly accurately.

On real data experiments, the superior performance of our model, compared to baselines which are agnostic to the existence of underlying categories, demonstrates that real data does exhibit a non-trivial combination of discrete and continuous behaviour.

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

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*, pages 670–688, Oct 2015. doi: 10.1109/FOCS.2015.47.

E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, Jan 2016. ISSN 1557-9654. doi: 10.1109/TIT.2015.2490670.

Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018. URL http://jmlr.org/papers/v18/16-480.html.

Kwangjun Ahn, Kangwook Lee, Hyunseung Cha, and Changho Suh. Binary rating estimation with graph side information. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4272–4283. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7681-binary-rating-estimation-with-graph-side-information.pdf.

Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126, 2010.

Ludovico Boratto, Salvatore Carta, and Michele Satta. Groups identification and individual recommendations in group recommendation algorithms. In *PRSAT@ recsys*, pages 27–34, 2010.

Ludovico Boratto, Salvatore Carta, and Gianni Fenu. Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering. *Future Generation Computer Systems*, 64:165–174, 2016.

Yan-Ying Chen, An-Jung Cheng, and Winston H Hsu. Travel recommendation by mining people attributes and travel group types from community-contributed photos. *IEEE Transactions on Multimedia*, 15(6):1283–1295, 2013.

Ingrid A Christensen and Silvia Schiaffino. Entertainment recommender systems for group of users. *Expert Systems with Applications*, 38(11):14127–14135, 2011.

Sriharsha Dara, C Ravindranath Chowdary, and Chintoo Kumar. A survey on group recommender systems. *Journal of Intelligent Information Systems*, pages 1–25, 2019.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3844–3852, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135, 2006.

Evgeny Frolov and Ivan Oseledets. Hybridsvd: when collaborative information is not enough. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 331–339, 2019.

Mikael Henaff, Joan Bruna, and Yann Lecun. Deep convolutional networks on graph-structured data. 06 2015.

Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137, 1983. ISSN 0378-8733. doi: https://doi.org/10.1016/0378-8733(83)90021-7. URL http://www.sciencedirect.com/science/article/pii/0378873383900217.

Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, page 135–142, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589060. doi: 10.1145/1864708.1864736. URL https://doi.org/10.1145/1864708.1864736.

Jyun-Yu Jiang, Patrick H Chen, Cho-Jui Hsieh, and Wei Wang. Clustering and constructing user coresets to accelerate large-scale top-k recommender systems. In *Proceedings of The Web Conference 2020*, pages 2177–2187, 2020.

Vassilis Kalofolias, Xavier Bresson, Michael Bronstein, and Pierre Vandergheynst. Matrix Completion on Graphs. *arXiv e-prints*, art. arXiv:1408.1717, August 2014.

Heung-Nam Kim and Abdulmotaleb El Saddik. A stochastic approach to group recommendations in social media systems. *Information Systems*, 50:76–93, 2015.

Antoine Ledent, Rodrigo Alves, and Marius Kloft. Orthogonal inductive matrix completion. *arXiv preprint arXiv:2004.01653*, 2020.

Tao Li, Vikas Sindhwani, Chris Ding, and Yi Zhang. Bridging domains with words: Opinion analysis with matrix tri-factorizations. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 293–302. SIAM, 2010.

Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 252–260. SIAM, 2013.

Hao Ma, Dengyong Zhou, Chao Liu, Michael Lyu, and Irwin King. Recommender systems with social regularization. pages 287–296, 01 2011. doi: 10.1145/1935826.1935877.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res.*, 11:2287–2322, August 2010. ISSN 1532-4435.

Jungseul Ok, Se-Young Yun, Alexandre Proutiere, and Rami Mochaourab. Collaborative clustering: Sample complexity and efficient algorithms. volume 76 of *Proceedings of Machine Learning Research*, pages 288–329, Kyoto University, Kyoto, Japan, 15–17 Oct 2017. PMLR. URL http://proceedings.mlr.press/v76/ok17a.html.

Qiaosheng, Zhang, Vincent Y. F. Tan, and Changho Suh. Community Detection and Matrix Completion with Two-Sided Graph Side-Information. *arXiv e-prints*, art. arXiv:1912.04099, December 2019.

Shunichi Seko, Manabu Motegi, Takashi Yagi, and Shinyo Muto. Video content recommendation for group based on viewing history and viewer preference. In *ITE Technical Report 35.7*, pages 25–26. The Institute of Image Information and Television Engineers, 2011.

Jing Shi, Bin Wu, and Xiuqin Lin. A latent group model for group recommendation. In *2015 IEEE International conference on mobile services*, pages 233–238. IEEE, 2015.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596, 2019. URL http://arxiv.org/abs/1901.00596.

J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156, Dec 2013. doi: 10.1109/ICDM.2013.167.