Scalable Probabilistic Matrix Factorization with Graph-Based Priors

Jonathan Strahl,[1] Jaakko Peltonen,[2] Hiroshi Mamitsuka,[1,3] Samuel Kaski[1]

[1]Department of Computer Science, Aalto University, Finland
[2]Tampere University, Faculty of Information Technology and Communication Science, Finland
[3]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan

jonathan.strahl@aalto.fi, jaakko.peltonen@tuni.fi, mami@kuicr.kyoto-u.ac.jp, samuel.kaski@aalto.fi

Abstract

In matrix factorization, available graph side-information may not be well suited for the matrix completion problem, having edges that disagree with the latent-feature relations learnt from the incomplete data matrix. We show that removing these *contested* edges improves prediction accuracy and scalability. We identify the contested edges through a highly-efficient graphical lasso approximation. The identification and removal of contested edges adds no computational complexity to state-of-the-art graph-regularized matrix factorization, remaining linear with respect to the number of non-zeros. Computational load even decreases proportional to the number of edges removed. Formulating a probabilistic generative model and using expectation maximization to extend graph-regularised alternating least squares (GRALS) guarantees convergence. Rich simulated experiments illustrate the desired properties of the resulting algorithm. On real data experiments we demonstrate improved prediction accuracy with fewer graph edges (empirical evidence that graph side-information is often inaccurate). A 300 thousand dimensional graph with three million edges (Yahoo music side-information) can be analyzed in under ten minutes on a standard laptop computer demonstrating the efficiency of our graph update.

1 Introduction

Matrix factorization (MF) is popular in a number of domains including recommender systems (Koren, Bell, and Volinsky 2009; Mehta and Rana 2017), bioinformatics (Brunet et al. 2004; Jacoby and Brown 2018; Stein-OBrien et al. 2018; Zakeri et al. 2018; Zheng et al. 2013), image restoration (Xue, Zhang, and Cai 2017) and many more (Davenport and Romberg 2016). Much of the data is of a very large scale and sparse, and additional (side-)information is usually available. Therefore, many methods focus on scalability (Davenport and Romberg 2016; Mnih and Salakhutdinov 2008; Sardianos, Papadatos, and Varlamis 2019) and the addition of side information (SI) (Chiang, Hsieh, and Dhillon 2015; Chiang, Dhillon, and Hsieh 2018; Gönen, Khan, and Kaski 2013; Ma et al. 2011; Zakeri et al. 2018; Zhou et al. 2012; Zhao et al. 2015), and more recently scalable methods with SI (Monti, Bronstein, and Bresson 2017; Rao et al. 2015; Yao and Li 2018).

Empirical evidence shows that prediction accuracy is significantly improved by graph SI, where edges in the graph represent similarity between connected nodes (Cai et al. 2011; Ma et al. 2011; Monti, Bronstein, and Bresson 2017; Rao et al. 2015; Yao and Li 2018; Zhou et al. 2012; Zhao et al. 2015). MF (or low-rank matrix completion) has theoretical guarantees for exact completion without and with noise (Candes and Plan 2010; Candès and Recht 2009). Introducting noisy SI is shown to reduce sample-complexity, and is reduced even further handling the noise (Chiang, Hsieh, and Dhillon 2015). Reduction in sample complexity through the introduction of graph SI has also been shown (Ahn et al. 2018; Rao et al. 2015), as a function of graph quality. However, to the best of our knowledge there is no work on scalable methods to handle the noise in the graph SI.

Mnih and Salakhutdinov (Mnih and Salakhutdinov 2008) introduced probabilistic matrix factorisation (PMF), which is equivalent to ℓ_2-regularised (alternating least squares) MF. Probabilistic interpretations for MF with graph SI are kernelized PMF (KPMF (Zhou et al. 2012)) and kernelized Bayesian MF (KBMF (Gönen, Khan, and Kaski 2013)): placing priors over the columns of the latent feature matrices. This type of prior models the pairwise relation between rows, where these rows correspond to rows or columns of the incomplete data matrix. KPMF and KBMF showed good results on moderate-sized data but failed to scale to large data.

To address scalability, graph-regularised alternating least squares (GRALS (Rao et al. 2015)) was proposed, with conjugate gradient descent exploiting the sparsity in the data matrix and the graphs, resulting in linear computational complexity and fast convergence. Recently there has been progress on applying deep learning to matrix completion, with and without side information, with good accuracy and showing potential for scalability (Berg, Kipf, and Welling 2017; Hartford et al. 2018; Monti, Bronstein, and Bresson 2017; Yao and Li 2018).

All of the non-Bayesian or scalable methods incorporating graph SI (Cai et al. 2011; Ma et al. 2011; Monti, Bronstein, and Bresson 2017; Rao et al. 2015; Zhou et al. 2012)

fix the edges in the graph, considering them as true. However, these graphs are known to be uncertain (Adar and Re 2007; Asthana et al. 2004), and furthermore, the similarities they represent (e.g. homophily (McPherson, Smith-Lovin, and Cook 2001)) are rarely specific to the matrix factorization task leaving no guarantee that correlations correspond (Ma et al. 2011; Singla and Richardson 2008); graphs are often formed for other purposes, and hence their usefulness for MF is uncertain. This leaves room for improving the quality of the graph, leading to a significant reduction in sample complexity (Ahn et al. 2018). In this work we will introduce a solution based on contested edges, defined later in the paper.

Example of Graph Side-Information and Contested Edges To better understand how graph similarities are not task-specific (are non-specific) to MF, take a common example of a movie-recommendation problem with social network (SN) SI (Ma et al. (2011) and in our experiments on Douban data). Connected users in the SN do not connect based on their similar preference of movies, instead they connect on the basis of a broader social context. Similarly, the demographic information in MovieLens[1], used to form a user-similarity graph, is only very indirectly related to the movie preferences (McPherson, Smith-Lovin, and Cook 2001). Nevertheless, more general similarity has been shown to often work well in practice, but some parts of it may turn out to be detrimental as we illustrate below.

Figure 1 (top) shows a small movie-recommendation data matrix with SN SI (bottom-left). Without SI, if row/column observations in the data matrix are similar, latent features will be similar. This can be inaccurate, e.g. users 2 and 3 would be considered similar based on the observations, and thus predictions for user 2 would be similar to ratings of user 3, whereas actually user 2 is similar to user 1. Graph information can help by encouraging latent features of connected users, like user 1 and user 2 here, to be similar, even when there is no observed data in the matrix to indicate they should be. However, for other users such as 4 and 5 the graph may mismatch with the data, indicating similarity whereas 4 and 5 are actually negatively correlated (as seen in their ratings of movies 5 and 6), and using the graph would thus worsen their predictions. We propose using this discrepancy to *contest* the graph edge between users 4 and 5; removing this edge as in Figure 1 (bottom-right) would improve predictions for users 4 and 5 to be consistent with their observed negative correlation, while the beneficial edge between users 1 and 2 will still remain. In real cases, mismatch between the data matrix and the SI would be detected based on much more data than in this illustration.

We do not propose to identify contested edges directly from the observed data but from correlations between the latent features. We introduce a probabilistic generative model that we call graph-based prior PMF (GPMF). Using the expectation-maximization (EM, (Bishop 2006)) algorithm we find a maximum a posteriori (MAP) estimate for the latent features and a maximum likelihood estimate (MLE)

[1]https://grouplens.org/datasets/movielens/

	Movie						
User	m_1	m_2	m_3	m_4	m_5	m_6	m_7
u_1	5		1				
u_2	5	4	1				
u_3	1	4	5				
u_4				5	4	2	1
u_5				1	2	4	5

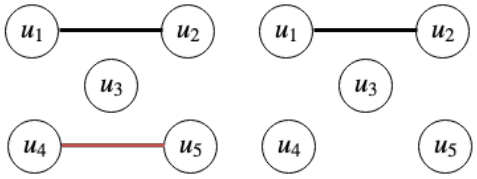


Figure 1: An illustrative movie recommendation problem. *Top:* data matrix where entries are user-ratings for movies: observations in black, unseen entries are blank and unseen entries to be predicted are in grey. *Bottom-left:* Social Network SI; connected users assumed to have similar ratings. The edge shown in red is contested due to negative correlation of u_4 and u_5 in the data matrix. *Bottom-right:* a graph update with removal of the contested edge to improve prediction accuracy.

for the correlations of the latent features. We show in Section 3 how using GLASSO approximation we can remove contested edges by simply thresholding a constrained sample covariance matrix (SCM).

There exist a number of approaches to reduce the edges in a labelled graph, graph summarization, Liu et al. (2018) for example. Most of these approaches do not use node attributes (labels) and to the best of our knowledge none use latent features for edge pruning. There are link prediction models that are probabilistic and use node attributes (Haghani and Keyvanpour 2017) but none of them can (yet) scale to large data (Li et al. 2014; Nguyen and Mamitsuka 2012; Zhao, Du, and Buntine 2017).

This paper introduces GPMF: the generative model in Section 2, the scalable constrained EM algorithm in Section 3, experiments in Section 4 and a conclusion in Section 5.

2 GPMF Generative Model and Relations to the Graph Side-Information

We are provided with a partially observed data matrix $\boldsymbol{R}$ with N rows and M columns. $\boldsymbol{R}$ is approximated as the product of two low-rank matrices, $\boldsymbol{U}$ and $\boldsymbol{V}$. The number of latent features D is fixed; $\boldsymbol{U}$ and $\boldsymbol{V}$ have D columns, each row is a latent feature vector for each row / column of $\boldsymbol{R}$ respectively. We use an indicator matrix $\boldsymbol{\Omega}$ where $[\boldsymbol{\Omega}]_{ij}$ is one if the element in row i and column j of $\boldsymbol{R}$ is observed, and zero otherwise. The goal is to learn latent-feature matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ that most accurately represent the full matrix $\boldsymbol{R}$.

ℓ_2-regularized MF has a scalable probabilistic interpretation: PMF. Each observed entry $\boldsymbol{R}_{ij} : (i,j) \in \{(s,t) : [\boldsymbol{\Omega}]_{s,t} = 1\}$ is assumed to have Gaussian noise σ^2; each row of $\boldsymbol{U}$ and $\boldsymbol{V}$ has a zero-mean spherical Gaussian prior. Similar to KPMF (Zhou et al. 2012), our model replaces the spherical Gaussian prior with a full-covariance Gaussian over the columns of the latent features (introducing row-wise dependencies):

$$p(\boldsymbol{R} \mid \boldsymbol{U}, \boldsymbol{V}, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \mathcal{N}(\boldsymbol{R}_{ij} \mid \boldsymbol{U}_{i:} \boldsymbol{V}_{j:}^{\top}, \sigma^2)^{\boldsymbol{\Omega}_{ij}} \quad (1)$$

$$p(\boldsymbol{U} \mid \boldsymbol{\Lambda}_U) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{U}_{:d} \mid \boldsymbol{0}, \boldsymbol{\Lambda}_U^{-1}) \quad (2)$$

$$p(\boldsymbol{V} \mid \boldsymbol{\Lambda}_V) = \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{V}_{:d} \mid \boldsymbol{0}, \boldsymbol{\Lambda}_V^{-1}) \; . \quad (3)$$

Graph SI constrains the structure of the precision matrices ($\boldsymbol{\Lambda}_U$ or $\boldsymbol{\Lambda}_V$) of (2) and (3), discussed next.

Gaussian Markov Random Field (GMRF) Relation to Precision Matrix

An undirected graph $\mathcal{G}_Z = (\mathcal{V}_Z, \mathcal{E}_Z)$ with a set of nodes $\mathcal{V}_Z$, representing a set of random variables $\{Z_i\}_{i=1}^{P}$, and a set of edges $\mathcal{E}_Z \subseteq \{(i,j) \mid i,j \in \mathcal{V}_Z\}$, defines the conditional independence of the random variables, where the absence of an edge $(i,j) \notin \mathcal{E}_Z$ implies that the two random variables are conditionally independent $[\boldsymbol{\Lambda}_Z]_{ij} = 0$ given the remaining random variables (Bishop 2006; Hastie, Tibshirani, and Friedman 2009; Lauritzen 1996; Rue and Held 2005): $Z_i \perp Z_j \mid \{Z_k : k \in \{1, ..., N\} \setminus (i,j)\}$. In the remainder of the paper we refer to the adjacency matrix of $\mathcal{G}_Z$: a symmetric matrix where $[\boldsymbol{A}_Z]_{ij}$ is one if an edge exists between nodes i and j and zero otherwise. We can summarize the GMRF relation as $[\boldsymbol{A}_Z]_{ij} = 0 \iff [\boldsymbol{\Lambda}_Z]_{ij} = 0 \mid i \neq j$.

Laplacian Matrix Relation to Precision Matrix

The Laplacian matrix of a graph is $\boldsymbol{L}_Z = \boldsymbol{D} - \boldsymbol{A}_Z$, where $\boldsymbol{D}_{i,i} = \sum_{j=1}^{N} [\boldsymbol{A}_Z]_{ij}$ is a diagonal degree matrix, and is positive-semi-definite by definition. The regularised Laplacian $\boldsymbol{L}_Z^{+} = \boldsymbol{L}_Z + \gamma \boldsymbol{I}$, $\gamma > 0$ is a positive-definite matrix; a valid precision matrix retains the GMRF property (Dong et al. 2016; Egilmez, Pavez, and Ortega 2016; 2017; Hastie, Tibshirani, and Friedman 2009; Liu et al. 2014): $[\boldsymbol{L}_Z^{+}]_{ij} = 0 \iff [\boldsymbol{\Lambda}_Z]_{ij} = 0 \mid i \neq j$.

Lemma 1. *If the precision matrix in* (2) *and* (3) *is the regularised Laplacian matrix $\boldsymbol{L}_U^{+}, \boldsymbol{L}_V^{+}$, then the MAP estimator of our model has the same objective function as GRALS (Rao et al. 2015). Our GPMF model therefore gives a generalization of the GRALS objective function.*

Proof of Lemma 1.. Our generative model is biconvex, and hence it suffices to prove for $\boldsymbol{U}$ that the posterior is equivalent to the GRALS objective. Holding $\boldsymbol{V}$ fixed and finding the log posterior of $\boldsymbol{U}$:

$$\ln p(\boldsymbol{U} | \boldsymbol{R}, \sigma^2, \boldsymbol{V}, \boldsymbol{\Lambda}_U) \propto \ln p(\boldsymbol{R} \mid \boldsymbol{U}, \boldsymbol{V}, \sigma^2) p(\boldsymbol{U} \mid \boldsymbol{\Lambda}_U)$$

$$\propto -\frac{1}{2} \sum_{(i,j):\Omega_{i,j}=1} \left(\boldsymbol{R}_{ij} - \boldsymbol{U}_{i:} \boldsymbol{V}_{j:}^{\top} \right)^2 - \frac{\sigma^2}{2} \sum_{d=1}^{D} \boldsymbol{U}_{:d}^{\top} \boldsymbol{\Lambda}_U \boldsymbol{U}_{:d}$$

$$= -\frac{1}{2} \| \mathcal{P}_{\Omega}(\boldsymbol{R} - \boldsymbol{U}\boldsymbol{V}^{\top}) \|_{\mathrm{F}}^2 - \frac{\sigma^2}{2} \mathrm{tr}(\boldsymbol{U}^{\top} \boldsymbol{L}_U^{+} \boldsymbol{U}) \; , \quad (4)$$

where $\boldsymbol{U}_{i:}$ is row i of matrix $\boldsymbol{U}$ and $\boldsymbol{U}_{:d}$ is column d, $\mathcal{P}_{\Omega}$ is a projection operator retaining entries of the matrix in the set $\{(i,j) : \Omega_{i,j} = 1\}$, setting $\boldsymbol{\Lambda}_U = \boldsymbol{L}_U^{+}$ and noting that $\sum_{i,j} \boldsymbol{U}_{ij}^2 = \mathrm{tr}(\boldsymbol{U}^{\top}\boldsymbol{U}) = \|\boldsymbol{U}\|_{\mathrm{F}}^2$ is the Frobenius norm squared. Equation (4) is the GRALS objective function (Rao et al. 2015). Derivations are provided in the supplementary material. □

3 GRAEM: Scalable EM for GPMF

We naturally extend each least-squares sub-problem of GRALS (Rao et al. 2015) with graph-regularised alternating EM (GRAEM), having the same global convergence guarantees as GRALS: (Xu and Yin 2013). We work through optimising $\boldsymbol{U}$ with $\boldsymbol{V}$ fixed. Solving for $\boldsymbol{V}$ has the same form.

The EM Formulation

We have an incomplete data matrix $\boldsymbol{R}$, fixed matrix $\boldsymbol{V}$, latent variable matrix $\boldsymbol{U}$, and graph SI. From the graph we derive $\boldsymbol{L}_U^{+}$ (see Section 2), then set the precision matrix $\boldsymbol{\Lambda}_U = \boldsymbol{L}_U^{+}$. We want to maximize the expectation of the joint density of the data and the latent variables, with $\boldsymbol{U}$ as our unknowns and $\boldsymbol{\Lambda}_U$ as our input parameters:

$$\mathcal{Q}(\boldsymbol{\Lambda}_U, \boldsymbol{\Lambda}_U^{\text{old}}) = \int_{\boldsymbol{U}} p(\boldsymbol{U}|\boldsymbol{R}, \boldsymbol{\Lambda}_U^{\text{old}}) \ln p(\boldsymbol{R}, \boldsymbol{U} \mid \boldsymbol{\Lambda}_U) \, \mathrm{d}\boldsymbol{U}$$
$$= \mathbb{E}_{p(\boldsymbol{U}|\boldsymbol{R}, \boldsymbol{\Lambda}_U^{\text{old}})} \left[\ln p(\boldsymbol{R}, \boldsymbol{U} \mid \boldsymbol{\Lambda}_U) \right] \; . \quad (5)$$

E-step: Expected Value of the Latent Variables

The expected value of our latent variables has a Gaussian posterior distribution (see supplementary material), we can therefore use the MAP, which is equivalent to the GRALS objective function as shown in Lemma 1: $\mathbb{E}_{p(\boldsymbol{U}|\boldsymbol{R}, \boldsymbol{\Lambda}_U^{\text{old}})}[\, \boldsymbol{U} \,] = \boldsymbol{\mu}_U^{\text{post.}} \approx \hat{\boldsymbol{\mu}}_U^{MAP}$.

M-step: Removing Contested Edges

We can remove edges in the graph that correspond to negative correlations between the latent features by simply removing negative covariances from an SCM; this relationship holds for large scale and sparse problems; details follow.

The MLE of the Parameters and GLASSO To find the MLE we maximise the $\mathcal{Q}$ function in Equation (5) with respect to $\boldsymbol{\Lambda}_U$. The maximum can be found in closed form by taking the derivative with respect to the parameter $\boldsymbol{\Lambda}_U$ and setting to zero:

$$\arg\max_{\boldsymbol{\Lambda}_U} \mathcal{Q} = \left(\mathbb{E}_{p(\boldsymbol{U}|\boldsymbol{R}, \boldsymbol{\Lambda}_U^{\text{old}})} \left[\frac{1}{D} \sum_{d=1}^{D} \boldsymbol{U}_{:d} \boldsymbol{U}_{:d}^{\top} \right] \right)^{-1}$$
$$= \left(\mathbb{E} \left[\boldsymbol{S}_U^{D} \right] \right)^{-1} = \boldsymbol{\Lambda}_U^{*} \; . \quad (6)$$

Equation (6) is the inverse of an SCM, where each sample is one of the columns of $\boldsymbol{U}$. Values for $\boldsymbol{U}$ are unknown, so we use the MAP given the previous estimate of the parameters ($\boldsymbol{\Lambda}_U^{\text{old}}$). The solution (if any) is almost surely not sparse. Graphical lasso (GLASSO (Mazumder and Hastie 2012)) finds a sparse solution for the MLE of the precision matrix, where samples are assumed to be normally distributed, in line with our model assumptions in Section 2. We therefore propose solving (6) with GLASSO.

Constrained GLASSO and Highly Efficient Approximation GLASSO finds the MLE of the precision matrix under an ℓ_1 penalty, given an SCM $\boldsymbol{S}$. Grechkin et al. (2015) showed that the problem space can be reduced with prior knowledge on which pairwise relationships do not exist, forcing them to be zero in the solution:

$$\begin{aligned} \min_{\boldsymbol{\Lambda}_U \succeq 0} \quad & tr(\boldsymbol{S}\boldsymbol{\Lambda}_U) - \log|\boldsymbol{\Lambda}_U| + \tau \left\|\boldsymbol{\Lambda}_U\right\|_1 , \\ \text{subject to} \quad & \left[\boldsymbol{\Lambda}_U\right]_{ij} = 0 \; \forall \left\{(i,j) : \left[\boldsymbol{A}_U^0\right]_{ij} = 0\right\} . \end{aligned} \tag{7}$$

Zhang, Fattahi, and Sojoudi (2018) uses a relation between the sparsity structure of the τ-thresholded SCM and the GLASSO solution; for large-scale problems, when the solution is very sparse, the connected components are equivalent (Mazumder and Hastie 2012), given further assumptions the complete sparsity structure is equivalent (Fattahi and Sojoudi 2019; Sojoudi 2016a; 2016b). However, this solution will locate correlations, positive and negative, with a strong magnitude, greater than τ. Next we detail how to identify edges that correspond to only negative correlations.

Removing a Contested Edge The sparsity structure of the SCM and the (GLASSO) solution are equivalent under mild assumptions that are found to be true for sufficiently large τ, that result in $\approx 10N$ non-zeros in the solution (Fattahi and Sojoudi 2017; 2019). One of these assumptions is sign-consistency where each non-zero element of the solution has the opposite sign in the SCM. Assuming sign-consistency we can identify all graph edges that correspond to negative correlations in the latent features, with $\mathbb{E}[\boldsymbol{S}_U^D]$ from Equation (6) as our SCM:

$$[\boldsymbol{A}_U^{\text{new}}]_{ij} = \begin{cases} 1, & \left[\boldsymbol{A}_U^0\right]_{ij} = 1 \; , \; \mathbb{E}\left[\boldsymbol{S}_U^D\right]_{ij} \geq \tau \\ 0, & \left[\boldsymbol{A}_U^0\right]_{ij} = 1 \; , \; \mathbb{E}\left[\boldsymbol{S}_U^D\right]_{ij} < \tau \; , \; \textbf{CE} \\ 0, & \text{otherwise}, \qquad\qquad\qquad\qquad \textbf{cons-E}, \end{cases} \tag{8}$$

where $\boldsymbol{A}_U^{\text{new}}$ is the updated adjacency matrix, the threshold parameter τ is set to zero (or can be increased for a sparser solution) and $\boldsymbol{A}_U^0$ is the adjacency matrix of the graph SI; CE is a contested edge and cons-E is a constrained edge. To solve Equation (8) we need to compute $\mathbb{E}[\boldsymbol{S}_U^D]$, which can be decomposed as:

$$\begin{aligned} \mathbb{E}[\boldsymbol{S}_U^D] &= \frac{1}{D}\sum_{d=1}^{D} \mathbb{E}\left[\boldsymbol{U}_{:d}\boldsymbol{U}_{:d}^\top\right] \\ \mathbb{E}\left[\boldsymbol{U}_{:d}\boldsymbol{U}_{:d}^\top\right] &= \text{Cov}[\boldsymbol{U}_{:d}] + \mathbb{E}[\boldsymbol{U}_{:d}]\mathbb{E}[\boldsymbol{U}_{:d}^\top] \\ &= \boldsymbol{\Sigma}_{U_{:d}}^{\text{post.}} + \left[\boldsymbol{\mu}_{U_{:d}}^{\text{post.}}\right]\left[\boldsymbol{\mu}_{U_{:d}}^{\text{post.}}\right]^\top . \end{aligned}$$

The remaining task is to efficiently approximate the posterior covariance $\boldsymbol{\Sigma}_{U_{:d}}^{\text{post.}}$ for each column, d, of $\boldsymbol{U}$, which we discuss next.

Posterior Covariance Approximation The posterior of our GPMF model, in Section 2, is a joint Gaussian distribution, where the likelihood in Equation (1) introduces relations between the columns of the latent features and the prior in Equation (2) introduces relations between the rows. This results in a posterior covariance matrix with an inverse Kronecker sum structure (Kalaitzis et al. 2013; Schacke 2004): $\boldsymbol{\Sigma}_U^{\text{post.}} = (I_D \otimes \boldsymbol{\Lambda}_U + \alpha\ \boldsymbol{C})^{-1}$ where $\otimes$ is the Kronecker product operator and

$$\begin{aligned} \boldsymbol{C} &= [\boldsymbol{c}(d,d')]_{d,d'=1}^{D} \; , \\ \boldsymbol{c}(d,d') &= \text{diag}\left(\left\{\sum_{j=1}^{M} \boldsymbol{\Omega}_{ij}\boldsymbol{V}_{jd}\boldsymbol{V}_{jd'}\right\}_{i=1}^{N}\right) . \end{aligned}$$

Column-Wise Independence Assumption. We simplify the Kronecker sum with a column-wise independence assumption, setting all off-diagonals of $\boldsymbol{C}$ to zero:

$$\begin{aligned} \boldsymbol{\Lambda}_U^{\text{post.}} &\approx I_D \otimes \boldsymbol{\Lambda}_U + \alpha\ \text{diag}\left(\boldsymbol{C}\right) \\ &= \text{blkdiag}\left(\left\{\hat{\boldsymbol{\Lambda}}_{U_{:d}}^{\text{post.}}\right\}_{d=1}^{D}\right) , \\ \hat{\boldsymbol{\Lambda}}_{U_{:d}}^{\text{post.}} &= \boldsymbol{\Lambda}_U + \alpha\ \text{diag}\left(\boldsymbol{C}_d\right) , \\ \text{diag}\left(\boldsymbol{C}_d\right) &= \text{diag}\left(\left\{\sum_{j=1}^{M} \boldsymbol{\Omega}_{i,j}\boldsymbol{V}_{j,d}^2\right\}_{i=1}^{N}\right) , \end{aligned} \tag{9}$$

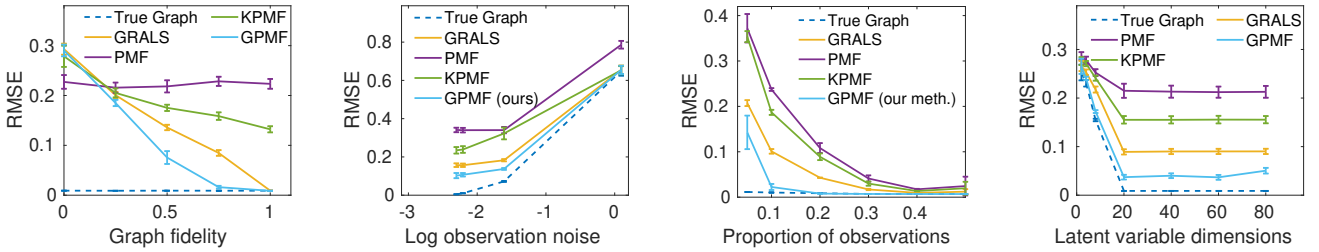
where $\alpha = [\sigma^2]^{-1}$ is the inverse of the observation noise in (1), diag takes a vector to create a diagonal matrix and blkdiag takes a sequence of matrices to construct a block-diagonal matrix.

Sparse Cholesky Factorisation: Each $\hat{\boldsymbol{\Lambda}}_{U_{:d}}^{\text{post.}}$ is still too large to invert. Assuming the high-dimensional matrix is sparse, as in Zhang, Fattahi, and Sojoudi (2018), its Cholesky factorisation is computable in $\mathcal{O}(N)$ time (Davis et al. 2004). We compute K samples as an unbiased estimate for the approximate posterior covariance:

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_{U_{:d}}^{\text{post.}} &= \left[\hat{\boldsymbol{\Lambda}}_{U_{:d}}^{\text{post.}}\right]^{-1} \approx \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{x}_k\boldsymbol{x}_k^\top \\ \boldsymbol{x}_k &\sim \mathcal{N}\left(\boldsymbol{0}, \left[\hat{\boldsymbol{\Lambda}}_{U_{:d}}^{\text{post.}}\right]^{-1}\right) . \end{aligned}$$

The Algorithm

The EM algorithm iterates between E-step and M-step until convergence. We initialize the latent feature matrices $(\boldsymbol{U}, \boldsymbol{V})$ by finding the MAP with no graph SI using PMF, to learn latent features that reflect the observed entries of the data matrix. In practise any method to learn the latent features with



(a) Decreasing contested edges (b) Increasing observation noise (c) Increasing observed entries (d) Increasing dimensionality

Figure 2: Synthetic data experiments

no SI can be used. The M step uses the relations between the latent features to identify negative correlations and remove them from the graph SI. The E-step then finds the MAP of the latent features given the updated graph. In theory the E and M step could be continued until some convergence criterion was met, but this would be less efficient and we get good results with just one step. So the three steps of our algorithm are lines 1,3 and 4:

Algorithm 1 Graph-regularised alternating EM (GRAEM)

Input: $\boldsymbol{A}_U^0, \boldsymbol{A}_V^0$
Output: $\hat{\boldsymbol{U}}, \hat{\boldsymbol{V}}, \boldsymbol{A}_U^+, \boldsymbol{A}_V^+$

1: $\boldsymbol{U}^0, \boldsymbol{V}^0 \leftarrow$ Initialise with PMF (no graphs)
2: **while** not converged **do**
3: $\quad \boldsymbol{A}_U^t, \boldsymbol{A}_V^t \leftarrow$ Run M-step Equation (8) with $\boldsymbol{U}^{t-1}, \boldsymbol{V}^{t-1}$ and $\boldsymbol{A}_U^0, \boldsymbol{A}_V^0$ as structural constraints
4: $\quad \boldsymbol{U}^t, \boldsymbol{V}^t \leftarrow$ Run E-step with regularized Laplacians given $\boldsymbol{A}_U^t, \boldsymbol{A}_V^t$
5: **end while**

Scalability: Computational Complexity

The algorithm has three steps: lines 1,3,4 in Algorithm 1. Line 1 is linear in the number of non-zeros $nz()$ in the data matrix $\mathcal{O}(nz(\boldsymbol{\Omega}))$ per conjugate gradient (CG) iteration. Line 3 comprises sparse Cholesky factorisation, linear in time with respect to the dimension size $\mathcal{O}(N+M)$, constrained SCM computation and thresholding, $\mathcal{O}(nz(\boldsymbol{A}_U) + nz(\boldsymbol{A}_V))$ both converge in one time step. Line 4 uses GRALS with the sparsified graphs: $\mathcal{O}(nz(\boldsymbol{\Omega}) + nz(\boldsymbol{A}_U^+) + nz(\boldsymbol{A}_V^+))$ per CG iteration. Line 4 is initialised with $\boldsymbol{U}, \boldsymbol{V}$ values from the PMF run, largely reducing the number of iterations required. Our algorithm remains linear with respect to the number of non-zeros. The additional M-step is a trivial additional cost, and if $\boldsymbol{A}_U^+, \boldsymbol{A}_V^+$ are much sparser, reducing iteration costs in Line 4, the overall computational load can be less than GRALS using the original graphs.

4 Experiments[2]

We compare our algorithm to a baseline with no graph SI (PMF, (Mnih and Salakhutdinov 2008)), the current most scalable method, GRALS (Rao et al. 2015), and to evaluate accuracy less scalable methods KPMF (Zhou et al. 2012) and sRMGCNN (Monti, Bronstein, and Bresson 2017). For sRMGCNN we used their published code, ran it on a (NVIDIA Tesla P100) GPU and used cross validation to tune the T value; this model took several orders of magnitude more time to converge: on Flixster data GPMF and GRALS converged in 20 seconds, PMF in 0.2 seconds, sRMGCNN took 30 minutes. We also ran KBMF (Gönen, Khan, and Kaski 2013) and non-convex IMC (Zhang, Du, and Gu 2018), with adjacency matrix rows as feature vectors, but with long computational time on the smaller datasets, we failed to achieve reasonable results. KPMF exploits rich side information and IMC experiments have more densely observed data, so they don't seem suited to this problem.

Experiments on Synthetic Data

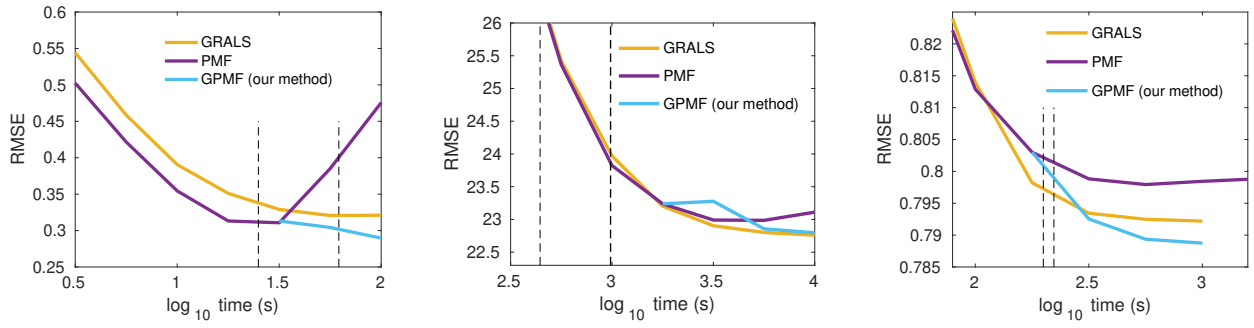
To analyze the behaviour of our algorithm we generate a data matrix with a known underlying graph. Therefore we can replace real edges in the graph with *corrupted edges* (CEs) that contest the real underlying structure, controlling the accuracy of the graph SI. We use a block-diagonal regularised-Laplacian precision matrix. We generate a 400×400 data matrix by Equations (1)-(3), with proportion of corrupted edges 0.3, observation noise 0.01, 7% observed values, and 40 latent dimensions; we vary these settings in the experiments below. See supplementary material for further details.

Graph Fidelity. In Figure 2 (a) we vary the number of CEs. A graph with no CEs has fidelity one ($F = 1$), with all CEs $F = 0$. GPMF consistently improves prediction accuracy over methods with graph SI for $F > 0$, and performance is equal for $F = 0$. PMF with no graph performs better below $F = 0.3$, showing that a graph of low quality can make prediction accuracy worse.

Observation Noise. Figure 2 (b) shows the benefit of GPMF diminishes as noise increases; learning correlations requires learning from the observations. However, at worst GPMF is only as bad as using the original corrupted graph.

Proportion of Observations. In Figure 2 (c) with just 10% of observed entries our algorithm can almost attain the same prediction accuracy as using the true graph. GRALS requires 30% to achieve a similar accuracy. At 40% of observed entries the graph is no longer beneficial. Note that most large scale matrix completion problems have fewer

[2]Code: https://github.com/strahl2e/GPMF-GBP-AAAI-20



(a) Epinions (45% of edges) (b) Yahoo Music (80% edges) (c) MovieLens20M (65% edges)

Figure 3: Convergence time on large data; vertical lines show start and end of M-step. c) 40NN graph.

than 10% observed entries.

Model Capacity. Figure 2 (d) shows that with too few latent features all models are negatively effected, but overall GPMF attains the best prediction accuracy.

GLASSO Accuracy We see clearly that observation noise strongly effects the ability to identify contested edges, as shown in Figure 2 (b). Accuracy improves with more observations, but even with low levels of noise and a reasonable amount of observations successful removal of CEs is moderate. Regardless of this moderate (best-case) accuracy, experiments show it is enough to attain significant improvements in accuracy of the latent features. We analyse the accuracy of removing CEs over several simulations. With 7% of observed entries, 31.7% of CEs are correctly removed and 19% of true edges (TEs) are wrongly removed; increasing observed entries to 40%, 44.3% of CEs are removed and 0.3% of TEs. Fixing observed entries at 20%, with noise $\sigma^2 = 0.01$, 39% of CEs and 2.7% of TEs are removed, and with $\sigma^2 = 1$, 34.3% CEs and 42.7% TEs are removed.

Experiments on Real Data

In Table 1 GPMF (our method) gives improved accuracy over GRALS on all small datasets: 3000 (3k) by 3k subsets of Flixster and Douban (Monti, Bronstein, and Bresson 2017), full datasets not attainable, and MovieLens100k (Harper and Konstan 2015)); the bottom rows of the table show the size and number of observations for each data matrix and the number of edges in each SI graph. In Figure 3 our method is shown to add no computational cost on large data: Epinions (Tang, Gao, and Liu 2012), Yahoo Music (Rao et al. 2015; Dror et al. 2011) and MovieLens 20 million (Harper and Konstan 2015)), note that proportion of edges used by GPMF is reported in figure title. Figure 3 (a) is an example of poor quality graph SI, we see this as PMF outperforms GRALS with the SI; our method (GPMF) estimates over half the edges as contested, removing them improves the accuracy. We believe that there were no gains in Figure 3 (b) as the graph is extremely sparse and removing edges has little effect. We test this hypothesis with MovieLens 20M in Table 1 by increasing the number of nearest neighbours from 10 to 40, we see that GRALS with the original graph decreases in performance while our algorithm continues to improve, we plot $k = 40$ in Figure 3 (c); the computational time of our M-step (between the two vertical lines) is a fraction of the running time of the algorithm with $k = 40$.

We also tested general usefulness of the updated graph: We get a small improvement for Douban with KPMF using 77 % of the edges, we also get the same accuracy for Flixster with almost half the edges.

5 Conclusion

We present a highly efficient method to improve the quality of graph side-information for matrix factorisation. Of the three steps in the algorithm, the initialisation of the latent features and the estimation of the latent features with the updated graph (the E-step) can be performed with any method for matrix completion without SI and with graph SI respectively. With such a small computational cost a graph update (the M-step) to improve quality seems like a valuable step when including graph SI into matrix factorisation. Furthermore, we demonstrated the robustness using our algorithm on real graph side-information. By increasing the number of nearest neighbours for generating graphs from feature side-information our algorithm, GRAEM, improved while GRALS worsened. Our graph update step allows for more noisy graphs to improve the matrix completion accuracy.

Future work could improve the graph update accuracy; we showed with simulated data the GLASSO approximation is only moderately successful.

6 Acknowledgments

The research was partly funded by the Academy of Finland grant 313748 and Business Finland grant 211548. H.M. has also been supported by JST ACCEL grant JPMJAC1503, MEXT Kakenhi grant 16H02868 and 19H04169. Computational resources provided by the Aalto Science-IT project.

References

Adar, E., and Re, C. 2007. Managing uncertainty in social networks. *IEEE Data Eng. Bull.* 30(2):15–22.

Ahn, K.; Lee, K.; Cha, H.; and Suh, C. 2018. Binary rating estimation with graph side information. In *Adv Neur In*, 4272–4283.

Table 1: Result summary on real datasets (RMSE), $\boldsymbol{A}^+$ is the graph updated with GRAEM (our method) where contested edges have been removed, we report the proportion of remaining edges in the bottom row. Bold = best result.

ALGO.	FLIXSTER (3K)	DOUBAN (3K)	MOVIELENS 100K	EPINIONS	YAHOO MUSIC	MOVIELENS 20M (10-/20-/40-NN)
PMF	0.9809	0.7492	0.9728	0.31	22.991	0.7980 / 0.7980 / 0.7980
GRALS	0.9152	0.7504	0.9178	0.32	**22.760**	0.7898 / 0.7925 / 0.7922
GPMF (OURS)	**0.8857**	0.7497	**0.9174**	**0.28**	22.795	0.7894 / 0.7895 / **0.7887**
KPMF	0.9212	0.7324	0.9336	-	-	-
KPMF ($\boldsymbol{A}^+$)	0.9212	**0.7323**	0.9374	-	-	-
SRMGCNN	0.9108	0.7915	0.9263	-	-	-
DATA DIMS.	3K X 3K	3K X 3K	1K X 1.5K	22K X 296K	250K X 300K	138K X 27K
NUM. OF OBS.	2.6K	137K	100K	824K	6M	20M
EDGES ($\boldsymbol{A}_U/\boldsymbol{A}_V$)	59K / 51K	2.7K / 0	12.6K / 29K	574K / 0	0 / 3M	0 / 493K - 0 / 963K - 0 / 1.9M
PROP. OF EDGES IN $\boldsymbol{A}^+$	0.57 / 0.63	0.77 / 0	0.63 / 0.61	0.45 / 0	0 / 0.8	0 / 0.88 - 0 / 0.71 - 0 / 0.65

Asthana, S.; King, O. D.; Gibbons, F. D.; and Roth, F. P. 2004. Predicting protein complex membership using probabilistic network reliability. *Genome research* 14(6):1170–1175.

Berg, R. v. d.; Kipf, T. N.; and Welling, M. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Brunet, J.-P.; Tamayo, P.; Golub, T. R.; and Mesirov, J. P. 2004. Metagenes and molecular pattern discovery using matrix factorization. *P Natl Acad Sci USA* 101(12):4164–4169.

Cai, D.; He, X.; Han, J.; and Huang, T. S. 2011. Graph regularized nonnegative matrix factorization for data representation. *IEEE T Pattern Anal* 33(8):1548–1560.

Candes, E. J., and Plan, Y. 2010. Matrix completion with noise. *Proceedings of the IEEE* 98(6):925–936.

Candès, E. J., and Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9(6):717.

Chiang, K.-Y.; Dhillon, I. S.; and Hsieh, C.-J. 2018. Using side information to reliably learn low-rank matrices from missing and corrupted observations. *Journal of Machine Learning Research* 19(76):1–35.

Chiang, K.-Y.; Hsieh, C.-J.; and Dhillon, I. S. 2015. Matrix completion with noisy side information. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Adv Neur In*, 3447–3455. Curran Associates, Inc.

Davenport, M. A., and Romberg, J. 2016. An overview of low-rank matrix recovery from incomplete observations. *IEEE J Sel Top Signa* 10(4):608–622.

Davis, T. A.; Gilbert, J. R.; Larimore, S. I.; and Ng, E. G. 2004. A column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software (TOMS)* 30(3):353–376.

Dong, X.; Thanou, D.; Frossard, P.; and Vandergheynst, P. 2016. Learning laplacian matrix in smooth graph signal representations. *IEEE T Signal Proces* 64(23):6160–6173.

Dror, G.; Koenigstein, N.; Koren, Y.; and Weimer, M. 2011. The yahoo! music dataset and kdd-cup'11. In *Proceedings of the 2011 International Conference on KDD Cup 2011-Volume 18*, 3–18. JMLR. org.

Egilmez, H. E.; Pavez, E.; and Ortega, A. 2016. Graph learning with laplacian constraints: Modeling attractive gaussian markov random fields. In *Signals, Systems and Computers, 2016 50th Asilomar Conference on*, 1470–1474. IEEE.

Egilmez, H. E.; Pavez, E.; and Ortega, A. 2017. Graph learning from data under laplacian and structural constraints. *IEEE J Sel Top Signa* 11(6):825–841.

Fattahi, S., and Sojoudi, S. 2017. Graphical lasso and thresholding: Equivalence and closed-form solutions. *arXiv preprint arXiv:1708.09479*.

Fattahi, S., and Sojoudi, S. 2019. Graphical lasso and thresholding: equivalence and closed-form solutions. *The Journal of Machine Learning Research* 20(1):364–407.

Gönen, M.; Khan, S.; and Kaski, S. 2013. Kernelized bayesian matrix factorization. In *International Conference on Machine Learning*, 864–872.

Grechkin, M.; Fazel, M.; Witten, D.; and Lee, S.-I. 2015. Pathway graphical lasso. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Haghani, S., and Keyvanpour, M. R. 2017. A systemic analysis of link prediction in social network. *Artificial Intelligence Review* 1–35.

Harper, F. M., and Konstan, J. A. 2015. The movielens datasets: History and context. *Transactions on Interactive Intelligent Systems* 5(4).

Hartford, J.; Graham, D. R.; Leyton-Brown, K.; and Ravanbakhsh, S. 2018. Deep models of interactions across sets. *arXiv preprint arXiv:1803.02879*.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York.

Jacoby, E., and Brown, J. 2018. The future of computa-

tional chemogenomics. In *Computational Chemogenomics*. Springer. 425–450.

Kalaitzis, A.; Lafferty, J.; Lawrence, N.; and Zhou, S. 2013. The bigraphical lasso. In *International Conference on Machine Learning*, 1229–1237.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*.

Lauritzen, S. 1996. *Graphical Models*. Oxford science publications. Clarendon Press.

Li, K.; Gao, J.; Guo, S.; Du, N.; Li, X.; and Zhang, A. 2014. Lrbm: A restricted boltzmann machine based approach for representation learning on linked data. In *2014 IEEE International Conference on Data Mining*, 300–309. IEEE.

Liu, F.; Chakraborty, S.; Li, F.; Liu, Y.; Lozano, A. C.; et al. 2014. Bayesian regularization via graph laplacian. *Bayesian Analysis* 9(2):449–474.

Liu, Y.; Safavi, T.; Dighe, A.; and Koutra, D. 2018. Graph summarization methods and applications: A survey. *ACM Computing Surveys (CSUR)* 51(3):62.

Ma, H.; Zhou, D.; Liu, C.; Lyu, M. R.; and King, I. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 287–296. ACM.

Mazumder, R., and Hastie, T. 2012. The graphical lasso: New insights and alternatives. *Electron J Stat* 6:2125.

McPherson, M.; Smith-Lovin, L.; and Cook, J. M. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27(1):415–444.

Mehta, R., and Rana, K. 2017. A review on matrix factorization techniques in recommender systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, 269–274. IEEE.

Mnih, A., and Salakhutdinov, R. R. 2008. Probabilistic matrix factorization. In *Adv Neur In*, 1257–1264.

Monti, F.; Bronstein, M.; and Bresson, X. 2017. Geometric matrix completion with recurrent multi-graph neural networks. In *Adv Neur In*, 3697–3707.

Nguyen, C. H., and Mamitsuka, H. 2012. Latent feature kernels for link prediction on sparse graphs. *IEEE T Neur Net Lear* 23(11):1793–1804.

Rao, N.; Yu, H.-F.; Ravikumar, P. K.; and Dhillon, I. S. 2015. Collaborative filtering with graph information: Consistency and scalable methods. In *Adv Neur In*, 2107–2115.

Rue, H., and Held, L. 2005. *Gaussian Markov Random Fields: Theory and Applications*. CRC Press.

Sardianos, C.; Papadatos, G. B.; and Varlamis, I. 2019. Optimizing parallel collaborative filtering approaches for improving recommendation systems performance. *Information* 10(5):155.

Schacke, K. 2004. On the kronecker product. *Master's thesis, University of Waterloo*.

Singla, P., and Richardson, M. 2008. Yes, there is a correlation:-from social networks to personal behavior on the web. In *Proceedings of the 17th international conference on World Wide Web*, 655–664. ACM.

Sojoudi, S. 2016a. Equivalence of graphical lasso and thresholding for sparse graphs. *The Journal of Machine Learning Research* 17(1):3943–3963.

Sojoudi, S. 2016b. Graphical lasso and thresholding: Conditions for equivalence. In *Decision and Control (CDC)*, 7042–7048. IEEE.

Stein-OBrien, G. L.; Arora, R.; Culhane, A. C.; Favorov, A. V.; Garmire, L. X.; Greene, C. S.; Goff, L. A.; Li, Y.; Ngom, A.; Ochs, M. F.; et al. 2018. Enter the matrix: factorization uncovers knowledge from omics. *Trends in Genetics*.

Tang, J.; Gao, H.; and Liu, H. 2012. mtrust: discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*, 93–102. ACM.

Xu, Y., and Yin, W. 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences* 6(3):1758–1789.

Xue, H.; Zhang, S.; and Cai, D. 2017. Depth image inpainting: Improving low rank matrix completion with low gradient regularization. *IEEE T Image Process* 26(9):4311–4320.

Yao, K.-L., and Li, W.-J. 2018. Convolutional geometric matrix completion. *arXiv preprint arXiv:1803.00754*.

Zakeri, P.; Simm, J.; Arany, A.; ElShal, S.; and Moreau, Y. 2018. Gene prioritization using bayesian matrix factorization with genomic and phenotypic side information. *Bioinformatics* 34(13):i447–i456.

Zhang, X.; Du, S.; and Gu, Q. 2018. Fast and sample efficient inductive matrix completion via multi-phase procrustes flow. In *International Conference on Machine Learning*, 5751–5760.

Zhang, R.; Fattahi, S.; and Sojoudi, S. 2018. Large-scale sparse inverse covariance estimation via thresholding and max-det matrix completion. In *Proceedings of the 35th International Conference on Machine Learning*, 5766–5775. PMLR.

Zhao, Z.; Zhang, L.; He, X.; and Ng, W. 2015. Expert finding for question answering via graph regularized matrix completion. *IEEE T Knowl Data En* 27(4):993–1004.

Zhao, H.; Du, L.; and Buntine, W. 2017. Leveraging node attributes for incomplete relational data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 4072–4081. JMLR. org.

Zheng, X.; Ding, H.; Mamitsuka, H.; and Zhu, S. 2013. Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1025–1033. ACM.

Zhou, T.; Shan, H.; Banerjee, A.; and Sapiro, G. 2012. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, 403–414. SIAM.