



HAL
open science

Deep Probabilistic Matrix Factorization on Graphs: Application to Drug Repositioning for Monkeypox

Stuti Jain, Emilie Chouzenoux, Angshul Majumdar

► **To cite this version:**

Stuti Jain, Emilie Chouzenoux, Angshul Majumdar. Deep Probabilistic Matrix Factorization on Graphs: Application to Drug Repositioning for Monkeypox. Inria Saclay. 2022. hal-03878514

HAL Id: hal-03878514

<https://hal.science/hal-03878514>

Submitted on 29 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Deep Probabilistic Matrix Factorization on Graphs: Application to Drug Repositioning for Monkeypox

Stuti Jain⁽¹⁾, Emilie Chouzenoux⁽¹⁾ and Angshul Majumdar⁽²⁾

(1) CVN, Inria Saclay, Univ. Paris Saclay, Gif-sur-Yvette, 91190, France

(2) Dept. of ECE, IIIT - Delhi, India, 110020.

November 29, 2022

Abstract

Motivation: Over the centuries, there have been occurrences of various viral pandemics, one after another. There were major outbreaks caused by the Ebola virus, influenza virus, Mers-COV virus, Sars-COV virus, and most recently, the COVID-19 pandemic caused by Sars-COV-2. These pandemics caused havoc in the world, killed millions of humans and severely effected the world economy. Statistics indicate that COVID-19 will probably become endemic very soon. However, as the cases of monkeypox caused by the monkeypox virus (MPV) are seen to be increasing rapidly, there are fears that we might enter into another pandemic. The COVID-19 pandemic researchers faced numerous challenges in creating a viable vaccine, including a scarcity of time, high costs, and the potential for adverse effects due to lack of proper testing. In order to avoid these problems now use of existing drugs seem to be the best way to treat the infected individuals. This is called drug repositioning. This research aims to repurpose existing drugs for treating monkeypox. We aim to solve the problem of drug-virus interaction (DVA) through deep probabilistic matrix factorization with graph regularization. Here, we recast the curated database of drug-virus associations as a binary matrix, indicating whether the specific drug (from row) has been utilized for treating the associated virus (from column). This matrix is sparse. Prospective treatment regimes planning (using existing drugs) require to infer the missing entries of this matrix. We also utilize metadata for drugs and viruses, obtained from chemical and mechanism of action based similarities between drugs and genomic and symptomatic similarities between the viruses. We demonstrate that our proposed approach significantly outperforms state-of-the-art MF methods. Moreover, when our algorithm is used to make in silico predictions of anti-virals for monkeypox, it returns the drugs that are already being used to treat infected patients. This shows the high efficacy of our algorithm.

1 Introduction

Monkeypox is a viral disease that usually heals on its own; symptoms usually last two to four weeks ¹. It is rarely life - threatening, but patients may experience excruciating blister-like sores in delicate areas. Cases in youngsters and persons with impaired immune systems are more likely to be severe [1].

Monkeypox has historically been present in rural regions of West and Central Africa. The infection was typically spread by animals to people, so there wasn't much dispersion among the populace. But now, as the virus is spreading widely in countries where monkeypox is not regularly found, the current outbreak is an extremely rare occurrence and a matter of great concern ².

Monkeypox is not just a threat to nations in west and central Africa, but is also a major health risk worldwide. In 2003, monkeypox was first detected outside of Africa in the United States, where contact with infected prairie dogs had led to the infection. More than 70 monkeypox cases were reported in the U.S. during this outbreak. It has also been reported that travelers from Nigeria have been responsible for monkeypox cases in Israel in September 2018, the United Kingdom in September 2018, December 2019, May 2021, and May 2022, Singapore in May 2019, and the United States of America in July and November 2021. As recently as May 2022, monkeypox outbreaks have been reported in a number of non-endemic nations ³. Case numbers quickly increased in the aftermath of this report. More than 1,500 cases had been documented in 43 countries by June 10, 2022, including Europe and North America [2]. Studies are now being conducted to better understand the epidemiology, sources of infection and patterns of transmission. The Centers for Disease Control and Prevention (CDC) confirmed that there are no particular treatments for the 2022 monkeypox virus infection; however, existing antivirals proved effective against smallpox and may reduce the spread of monkeypox ⁴.

Given the current global state, the World Health Organization (WHO) has warned that the world may face another daunting infectious disease crisis. Furthermore, the current increase in number of cases worldwide may continue and become an international hazard. With this in mind, ways to limit the spread of the monkeypox virus and finding viable treatments are necessary.

In order to avoid the drastic situation faced by the researchers during the COVID-19 pandemic, where they faced a scarcity of time and resources to develop a viable vaccine as new strains kept on emerging. The best method appears to be repositioning/repurposing current drugs to treat the disease. In this strategy, existing drugs (those that have already been approved for use) are investigated for use in treating novel diseases. Because the relevant drug's effects have been well researched, the recommendation and inquiry procedure is

¹<https://www.cnbc.com/2022/10/01/monkeypox-unlikely-to-be-eliminated-in-the-us-cdc-says.html>

²<https://www.cnbc.com/2022/10/01/monkeypox-unlikely-to-be-eliminated-in-the-us-cdc-says.html>

³<https://www.who.int/news-room/fact-sheets/detail/monkeypox>

⁴mileto2022new

less expensive, less dangerous, and faster than producing a new drug or vaccine.

In short, we use existing data about drugs, viruses, and previously verified drug-virus interactions to train a predictive model, which is then used to predict interactions between drugs and corresponding viruses.

To put it briefly, we train a predictive model with data already available on drugs, viruses, and previously confirmed drug-virus interactions. This model is then used to anticipate unknown interactions between drugs and the corresponding viruses.

2 Approach

Traditionally, neighborhood models, matrix completion models, network diffusion models, and feature-based classification models have been used to solve the drug-virus interaction prediction problem. Recent empirical research using well-known drug-target interaction databases [3] shows that matrix completion models perform best at making drug virus association predictions.

Therefore, in this paper, we propose to solve the problem of drug-virus association prediction using graph regularized deep probabilistic matrix factorization. We offer a novel, theoretically sound approach (DPMFG framework) for this task.

Our current work is algorithmic in nature. We are proposing a probabilistic formulation for deep matrix factorization on graphs. To the best of our knowledge, this is the first proposition of a formulation and sounded inference method, for probabilistic deep matrix factorization. Studies like [4, 5] are not probabilistic or Bayesian in their formulations as they rely on regular deep neural networks trained via stochastic gradient descent. Our work can be thought of as a deep extension of probabilistic graph regularized matrix factorization [6].

We gathered data on drugs and viruses, including details on the chemical makeup and mode of action of drugs as well as the genomic sequences and symptoms related to viruses. In order to make graph-regularized approaches applicable, we then expressed it as drug and virus similarity matrices. Our models are assessed using a variety of metrics, including AUC, AUPR, precision, and recall. Additionally, using computational methods, we forecast potential medications that could be successful against the relevant viruses, including the novel Monkeypox virus.

3 Background

This section provides an introduction to matrix factorization approaches for matrix completion.

3.1 Matrix Completion Problem

Let us consider that we have a partially known matrix $Y \in \mathbb{R}^{N \times M}$, and we have to recover a full matrix $R \in \mathbb{R}^{N \times M}$ from it. Let us define the set \mathcal{D} as the set

of indices belonging to the partially observed matrix:

$$\mathcal{D} = \{i \in \{1, \dots, N\}, j \in \{1, \dots, M\} \text{ s.t. } (i, j) \text{ is observed}\}.$$

The partially known matrix Y can be expressed as:

$$Y = B \odot R, \tag{1}$$

where \odot is the Hadamard product and matrix $B \in \{0, 1\}^{N \times M}$ masks the indexes outside the set \mathcal{D} , and is defined such that $B_{ij} = 1$ if $(i, j) \in \mathcal{D}$, and $B_{ij} = 0$ otherwise.

The task of matrix completion (MC) is to recover the entries of R (or equivalently, of Y) that do not belong to the set of observed indices \mathcal{D} .

3.2 Matrix Factorization (MF)

In MF [7], the missing entries in matrix R are recovered by minimizing a simple least-squares function under some structural prior constraints on R . Specifically, R is assumed to be the product of two matrices $U_1 \in \mathbb{R}^{N \times K}$ and $U_2 \in \mathbb{R}^{K \times M}$, where $K \geq 1$ defines a latent space dimension (typically low compared to (N, M)). The matrices U_1 and U_2 are inferred by solving:

$$\underset{U_1, U_2}{\text{minimize}} \|Y - B \odot (U_1 U_2)\|_F^2. \tag{2}$$

Then, the full matrix is recovered as $R = U_1 U_2$. The problem (2) is, however, still ill-posed, and additional priors are frequently added to produce better solutions.

The NMF (nonnegative MF) formulation, created by imposing a positivity constraint on the entries of the latent components (U_1, U_2) , is probably one of the most well-known formulations. Nuclear norm minimization [8] is another strategy imposing low rank on R . In such approaches, additional regularization procedures based on graph modelling have been considered in [9] and [10], respectively. The so-called deep MF method, which is studied, for example, in the context of MC in [11], relies on MF models with more than two components (i.e., factors). The recent study [12] has proposed a graph regularized variant for deep MC.

3.3 Probabilistic Matrix Factorization (PMF)

PMF [13] introduces probabilistic models on the latent factors U_1 and U_2 in the aforementioned MF formulation. Precisely:

- Each observed entry $(Y_{ij})_{(i,j) \in \mathcal{D}}$ is assumed to be the realization of a Gaussian distribution, with mean $[U_1 U_2]_{i,j}$ and variance σ^2 (positive scalar assumed to be known).
- Each observed entry $([U_1]_{ik})_{1 \leq i \leq N, 1 \leq k \leq K}$ is assumed to be the realization of a Gaussian distribution with zero mean and variance $\sigma_{U_1}^2$ (positive scalar assumed to be known).

- Each observed entry $([U_2]_{kj})_{1 \leq k \leq K, 1 \leq j \leq M}$ is assumed to be the realization of a Gaussian distribution with zero mean and variance $\sigma_{U_2}^2$ (positive scalar assumed to be known).

The maximum a posteriori (MAP) estimator of (U_1, U_2) given Y , associated with the above model is obtained by solving:

$$\underset{U_1, U_2}{\text{minimize}} \frac{1}{2\sigma^2} \|Y - B \odot (U_1 U_2)\|_F^2 + \frac{1}{2\sigma_{U_1}^2} \|U_1\|_F^2 + \frac{1}{2\sigma_{U_2}^2} \|U_2\|_F^2. \quad (3)$$

The minimization of the above cost function with respect to U_1 (resp. U_2) amounts to inverting a linear system, which can be performed using a conjugate gradient solver (for instance). The PMF formulation can be enhanced by incorporating correlated Gaussian distributions, which results in the PMFG formulation described hereafter.

3.4 Probabilistic Matrix Factorization with Graph regularization (PMFG)

A graph regularization approach, called PMFG, is introduced in [6], to the prior distributions of U_1 and U_2 . This strategy entails jointly inferring correlations along the rows (resp. columns) of U_1 (resp. U_2), in top of estimating the factors U_1 and U_2 . Two precision matrices, $\Gamma_{U_1} \in S_N^{++}$ and $\Gamma_{U_2} \in S_M^{++}$, are used to model these correlations. Here, we denote S_N (resp. S_N^{++}) as the set of symmetric (resp. symmetric positive definite matrices) of size $N \times N$. Hence, the PMFG prior can be explained as:

- The columns $([U_1]_{:,k})_{1 \leq k \leq K} \in \mathbb{R}^N$ of U_1 are independent realizations of a multivariate Gaussian distribution with zero mean and covariance $C_{U_1} = \Gamma_{U_1}^{-1}$;
- The rows $([U_2]_{k,:})_{1 \leq k \leq K} \in \mathbb{R}^M$ of U_2 , are independent realizations of a multivariate Gaussian distribution with zero mean and covariance $C_{U_2} = \Gamma_{U_2}^{-1}$.

The precision matrices Γ_{U_1} and Γ_{U_2} are related to Gaussian graphical models associated with the two underlying Gaussian distributions [14], which justifies the name for graph regularization. Expressly, the matrix Γ_{U_1} (resp. Γ_{U_2}) can be understood as the adjacency matrix of an undirected graph where each edge (i.e., non zero entry in the precision matrix) identifies with two entries of $[U_1]_{:,k}$ (resp. $[U_2]_{k,:}$) being correlated, given all the others.

The MAP estimate can now be obtained by solving:

$$\underset{U_1, U_2, \Gamma_{U_1}, \Gamma_{U_2}}{\text{minimize}} \frac{1}{2\sigma^2} \|Y - B \odot (U_1 U_2)\|_F^2 + \frac{1}{2} \text{tr}(U_1^\top \Gamma_{U_1} U_1) + \frac{1}{2} \mathcal{L}(\Gamma_{U_1}) + \frac{1}{2} \text{tr}(U_2 \Gamma_{U_2} U_2^\top) + \frac{1}{2} \mathcal{L}(\Gamma_{U_2}), \quad (4)$$

where $\text{tr}(\cdot)$ denotes the trace operation and

$$(\forall \Gamma_{U_1} \in S_N) \quad \mathcal{L}(\Gamma_{U_1}) = \begin{cases} -\ln \det(\Gamma_{U_1}) & \text{if } \Gamma_{U_1} \in S_N^{++} \\ +\infty & \text{otherwise} \end{cases} \quad (5)$$

with $\det(\cdot)$ the determinant operation.

The PMFG formulation does not include any physical information for the sought factors (U_1, U_2) . However, such expert knowledge is present in many applications and determines if specific correlations between the variables are more or less likely. Extending the PMFG paradigm, the current work aims to propose a novel formulation to account for such prior knowledge.

4 Proposed Formulation

4.1 Incorporating the prior knowledge as regularization

It is expected that one expert has access to some prior information about the position of the graph edges (i.e. non zero elements in the precision matrices Γ_{U_1} and Γ_{U_2}). Following our previous work on drug-drug interaction [15], we propose to incorporate such expert knowledge through an extra regularization term in the loss function. Two symmetric adjacency matrices, $A_{U_1} \in [0, +\infty)^{N \times N}$ and $A_{U_2} \in [0, +\infty)^{M \times M}$, play a fundamental role in this new prior. These matrices, set by the user, are used to eliminate erroneous edges and favor anticipated ones in the estimated graph:

- $[\Gamma_{U_1}]_{ij} = [\Gamma_{U_1}]_{ji}$ should be encouraged to be high if $[A_{U_1}]_{ij}$ is large for some (i, j) with $i \neq j$.
- $[\Gamma_{U_1}]_{ij} = [\Gamma_{U_1}]_{ji}$ should be encouraged to be low if $[A_{U_1}]_{ij}$ is small for some (i, j) with $i \neq j$.

Similar prior is enforced on the second pair (A_{U_2}, Γ_{U_2}) .

In order to build a suitable regularization function, we introduce the following sets:

$$\mathcal{E}_{U_1} = \{(i, j) \in \{1, \dots, N\}^2, i \neq j \text{ and } [A_{U_1}]_{ij} > \tau\} \quad (6)$$

$$\bar{\mathcal{E}}_{U_1} = \{(i, j) \in \{1, \dots, N\}^2, i \neq j \text{ and } [A_{U_1}]_{ij} \leq \tau\} \quad (7)$$

$$\mathcal{E}_{U_2} = \{(i, j) \in \{1, \dots, M\}^2, i \neq j \text{ and } [A_{U_2}]_{ij} > \tau\} \quad (8)$$

$$\bar{\mathcal{E}}_{U_2} = \{(i, j) \in \{1, \dots, M\}^2, i \neq j \text{ and } [A_{U_2}]_{ij} \leq \tau\} \quad (9)$$

with $\tau \geq 0$ a given detection threshold. We propose to use an ℓ_1 term to promote sparsity in the regions where edges should not appear (i.e., $\bar{\mathcal{E}}_{U_1}$ and $\bar{\mathcal{E}}_{U_2}$), while we introduce a log-barrier term, smoothed by $\delta > 0$, in regions where edges should be promoted (i.e., \mathcal{E}_{U_1} and \mathcal{E}_{U_2}). A quadratic term that can be viewed

as an elastic-net penalty is added to avoid unbounded values in the entries of the sought covariance matrices.

In previously introduced MF-based formulations, for instance in PMFG, the non-linearity introduced by the factor $B \odot U_1 U_2$ inside the least-squares data fidelity term makes the minimization problem intricate. To stabilize convergence, we finally propose to introduce an auxiliary variable $X \in \mathbb{R}^{N \times M}$, that we penalize so as to be close to the product $U_1 U_2$. As a result, the binary mask B and the factorized expression $U_1 U_2$ are splitting, which simplifies the resolution through an alternating minimization method.

In a nutshell, we end up with solving:

$$\begin{aligned}
& \underset{X, U_1, U_2, \Gamma_{U_1}, \Gamma_{U_2}}{\text{minimize}} && \frac{1}{2\sigma^2} \|Y - B \odot X\|_F^2 \\
& && + \frac{1}{2} \text{tr}(U_1^\top \Gamma_{U_1} U_1) + \frac{1}{2} \mathcal{L}(\Gamma_{U_1}) + \frac{1}{2} \text{tr}(U_2 \Gamma_{U_2} U_2^\top) + \frac{1}{2} \mathcal{L}(\Gamma_{U_2}) \\
& && + \lambda_{U_1} \sum_{i,j \in \bar{\mathcal{E}}_{U_1}} |[\Gamma_{U_1}]_{ij}| + \lambda_{U_2} \sum_{i,j \in \bar{\mathcal{E}}_{U_2}} |[\Gamma_{U_2}]_{ij}| \\
& && - \lambda_{U_1} \sum_{i,j \in \mathcal{E}_{U_1}} \ln(|[\Gamma_{U_1}]_{ij}| + \delta) - \lambda_{U_2} \sum_{i,j \in \mathcal{E}_{U_2}} \ln(|[\Gamma_{U_2}]_{ij}| + \delta) \\
& && + \frac{\lambda_{U_1}}{2} \|\Gamma_{U_1}\|_F^2 + \frac{\lambda_{U_2}}{2} \|\Gamma_{U_2}\|_F^2 + \frac{\lambda_R}{2} \|X - U_1 U_2\|_F^2. \quad (10)
\end{aligned}$$

Parameter $\lambda_R > 0$ controls the fulfillment of the equality constraint $X = U_1 U_2$ while parameters $(\lambda_{U_1}, \lambda_{U_2}) > 0$ control the regularization imposed on both precision matrices.

4.2 Integrating deep structure within PMF

The widespread use of deep learning has led to the development of deeper versions of shallow models [16, 17, 18]. Recently, [19] proposed a matrix completion problem involving deep factorization. Following these recent lines, we propose to generalize the previous model by considering that X can be factorized into more than 2 factors. Without losing in generality, we explicit our methodology in the case of three factors (i.e., 2 layers), denoted $U_1 \in \mathbb{R}^{N \times k_1}$, $U_2 \in \mathbb{R}^{k_1 \times k_2}$, $U_3 \in \mathbb{R}^{k_2 \times M}$. Parameters k_1 and k_2 represent the (usually small) latent factor dimensions, typically finetuned through cross-validation over a training set. We preserve the graphical priors on the first and last factors, here U_1 and U_3 , related to precision matrices Γ_{U_1} and Γ_{U_3} . No expert knowledge is introduced for

the middle factors (here, U_2). The resulting minimization problem now reads:

$$\begin{aligned}
& \text{minimize} \\
& X \in \mathbb{R}^{N \times M}, U_1 \in \mathbb{R}^{N \times k_1}, U_2 \in \mathbb{R}^{k_1 \times k_2}, \\
& U_3 \in \mathbb{R}^{k_2 \times M}, \Gamma_{U_1} \in S_N, \Gamma_{U_3} \in S_M \\
& \frac{1}{2\sigma^2} \|Y - B \odot X\|_F^2 + \frac{1}{2} \text{tr}(U_1^\top \Gamma_{U_1} U_1) \\
& + \frac{1}{2} \mathcal{L}(\Gamma_{U_1}) + \frac{1}{2} \text{tr}(U_3 \Gamma_{U_3} U_3^\top) + \frac{1}{2} \mathcal{L}(\Gamma_{U_3}) \\
& + \lambda_{U_1} \sum_{i,j \in \bar{\mathcal{E}}_{U_1}} |[\Gamma_{U_1}]_{ij}| + \lambda_{U_3} \sum_{i,j \in \bar{\mathcal{E}}_{U_3}} |[\Gamma_{U_3}]_{ij}| \\
& - \lambda_{U_1} \sum_{i,j \in \mathcal{E}_{U_1}} \ln(|[\Gamma_{U_1}]_{ij}| + \delta) - \lambda_{U_3} \sum_{i,j \in \mathcal{E}_{U_3}} \ln(|[\Gamma_{U_3}]_{ij}| + \delta) \\
& + \frac{\lambda_{U_1}}{2} \|\Gamma_{U_1}\|_F^2 + \frac{\lambda_{U_3}}{2} \|\Gamma_{U_3}\|_F^2 + \frac{\lambda_R}{2} \|X - U_1 U_2 U_3\|_F^2. \quad (11)
\end{aligned}$$

Note that, in the case of two factors, we retrieve the model from Section 4.1.

4.3 Optimization Algorithm

We propose to use an alternating minimization strategy to solve (10). Hereagain, we specify it, without loss of generality, in the case of three factors.

Let us denote $F(X, U_1, U_2, U_3, \Gamma_{U_1}, \Gamma_{U_3})$ the loss function presented in (10). The initial matrix $U_1^{(0)}$ is obtained through the singular valued decomposition of $Y = SDZ^\top$. We further decompose the right part of the decomposition, DZ^\top , to initialize $U_2^{(0)}$ and $U_3^{(0)}$ respectively (for the case with 4 factor matrices, $U_3^{(0)}$ and $U_4^{(0)}$ would be obtained by decomposing again the right part of the decomposition). Then, starting for an initialization $(X^{(0)}, U_1^{(0)}, U_2^{(0)}, U_3^{(0)}, \Gamma_{U_1}^{(0)}, \Gamma_{U_3}^{(0)})$, for every iteration $p \in \mathbb{N}$, the algorithm updates follow an alternating minimization scheme. The complete method is given in Algorithm 1, when using $P \geq 1$ iterations. The explicit update rules (S1) to (S6) are provided in the supplementary file.

Algorithm 1 DPMFG (2 layers) for DVA Prediction

- 1: **Input:** Y, A_{U_1}, A_{U_2} .
 - 2: **Parameters:** $(\sigma, s^{(0)}, \lambda_R, \lambda_{U_1}, \lambda_{U_3}) > 0, (k_1, k_2, P, L) \geq 1$.
 - 3: **Initialization:** Set $U_1^{(0)}, U_2^{(0)}, U_3^{(0)}$ with SVD of Y .
 - 4: Set $\Gamma_{U_1}^{(0)} = s^{(0)}\text{Id}_N$ and $\Gamma_{U_3}^{(0)} = s^{(0)}\text{Id}_M$.
 - 5: Compute \mathcal{E}_{U_1} and $\bar{\mathcal{E}}_{U_1}$ using (6) and (7).
 - 6: Compute \mathcal{E}_{U_3} and $\bar{\mathcal{E}}_{U_3}$ using (8) and (9).
 - 7: **for** $p = 1, 2, \dots, P$ iterations
 - 8: Update $X^{(p+1)}$ using (S1)
 - 9: Update $U_1^{(p+1)}$ using (S2)
 - 10: Update $U_2^{(p+1)}$ using (S3)
 - 11: Update $U_3^{(p+1)}$ using (S4)
 - 12: Update $\Gamma_{U_1}^{(p+1)}$ using (S5)
 - 13: Update $\Gamma_{U_3}^{(p+1)}$ using (S6)
 - 14: **end**
 - 15: **Return:** $R = U_1^{(P)}U_2^{(P)}U_3^{(P)}$
-

5 Results

5.1 Dataset Description

We used the DVA dataset [20] to train and assess how well our novel algorithm performed. It combines drugs shown via research to be effective against viruses affecting people. This dataset contains $M = 38$ viruses and $N = 119$ drugs. In addition to the new monkeypox virus, this dataset is very helpful in analyzing and shortlisting the best effective antivirals for other viruses. Additionally, it can be used to find viruses that a potential new medicine might target computationally. Medical professionals can benefit from performing analysis and improving their comprehension with the help of supplementary metadata (information about how the drugs and viruses are similar; described in the next section). The database contains information on human-infecting DNA and RNA viruses.

5.2 Metadata Computation

The metadata is computed in the similar way as in our previous work [12]. We computed two types of drug-drug similarities. One is a chemical structure based similarity by using SIMCOMP scores [21]. The second is a similarity based on the mechanism of action of drugs, which is obtained by finding cosine similarities between one hot encoded representation of drug classes. Moreover, we computed two types of similarities between the viruses. One is a genomic structure based similarity using the ONF (Oligonucleotide frequency) measure by computing $d2^*$ distance [22]. Second, we calculate the cosine similarities between one hot encoded representations of the symptoms caused by a virus.

Hence, we obtain two drug-drug similarity matrices, each of size 119×119 that we sum to form A_{U_1} , and two virus-virus similarity matrices, each of size

38×38 , that we sum to form A_{U_2} .

5.3 Data Preprocessing

After computing the similarity matrices between drugs and viruses, the matrices are sparsified using a p-nearest neighbour graph [23], which is generated by keeping the similarity values of the closest neighbours for each drug or virus in the similarity matrices. This is done to remove noise and keep only the relevant information to improve the model performance. We have utilised the normalised versions of the graph laplacians instead of the conventional ones. Normalised laplacians outperform their non-normalised counterparts.

5.4 Experimental Evaluation

We used three cross-validation (CV) settings, each consisting of ten repetitions of 10-fold cross-validation.

- In CV1 setting, 10% of all drug-target pairs are hidden randomly and used as the test set while the remaining 90 % are used as the train set.
- In CV2 setting, 10% of the viral profiles are hidden randomly and used as the test set.
- In CV3 setting, 10% of the drug profiles are hidden randomly and used as the test set.

The accuracy of interaction prediction tasks can be assessed using the validation approach [24, 25]. The CV2 and CV3 settings are of significant importance. CV2 helps to evaluate the accuracy of drug recommendations for a novel virus, while CV3, helps to evaluate the efficiency of a recently introduced drug against a virus.

Further, cross-validation on the training set is used to determine the values of the parameters $(\sigma, s^0, \lambda_R, \lambda_{U_1}, \lambda_{U_3}, k_1, k_2, \theta_1, \theta_2)$ for DPMFG. Additionally, we set the number of iterations $P = 10$, which is enough to achieve stability.

We evaluate our approach with two, three or four factor matrices, denoted by DPMFG-1L, DPMFG-2L and DPMFG-3L respectively. Note that DPMFG-1L can be viewed as a modified version of our previously proposed algorithm in [15] to fit in our non-symmetric drug-virus interaction task. We compare with state-of-the-art matrix completion methods, including Deep Matrix Completion (DMF) [26], Graph Regularized Matrix Completion (GRMC-ADMM) [27], Graph Regularized Matrix Factorization (GRMF) [28], Graph Regularized Binary Matrix Completion (GRBMC) [29], Probabilistic Matrix Factorization with Graph Regularization (PMFG) [6].

Table 1: **AUC and AUPR obtained for the novel algorithm and the benchmarks**

Parameter CV Setting	AUC CV1	AUPR CV1	AUC CV2	AUPR CV2	AUC CV3	AUPR CV3
DPMFG-3L	0.8131	0.5934	0.6975	0.3002	0.8241	0.5588
DPMFG-2L	0.8215	0.5951	0.6809	0.3078	0.8317	0.5600
DPMFG-1L	0.8297	0.5482	0.7140	0.3036	0.8295	0.5569
GRMF	0.8418	0.5016	0.6934	0.2994	0.8151	0.4717
GRMC(ADMM)	0.8670	0.4825	0.6601	0.3390	0.8166	0.4723
GRBMC	0.8378	0.3870	0.6975	0.2927	0.8120	0.5272
DMF	0.6974	0.2615	0.5557	0.0911	0.5722	0.1389
PMFG	0.6136	0.3007	0.4943	0.0895	0.5633	0.1158

We look at the standard metrics such as the area under the ROC curve (AUC) and area under the precision-recall curve (AUPR) in order to statistically quantify and compare the performance of the proposed algorithm and the benchmarks. The AUPR metric is more significant here as the classes are highly imbalanced [30]. The results are shown in Table 1. The AUC and AUPR values in bold represent the best values obtained amongst all the algorithms considered. As seen clearly, our proposed algorithm (DPMFG-2L) gives the best results in terms of AUPR and is comparable to the benchmarks in terms of AUC. Also, we can see that the results did not improve for the deeper version of our algorithm (DPMFG-3L) for the current dataset. Therefore, for the DVA dataset, the version with 3 factor matrices performs the best.

Table 2: **Execution times (in seconds) for all techniques**

Parameter/ Algorithm	Execution Time (in seconds)
DPMFG-3L	0.1678
DPMFG-2L	0.1843
DPMFG-1L	0.1915
GRMF	0.0854
GRMC(ADMM)	16.9083
GRBMC	0.8501
DMF	0.1184
PMFG	0.1946

Table 2 shows the execution times of all the algorithms for one fold for CV setting 1. It can be clearly seen that the proposed method takes about the same amount of time as most of the benchmarks and is much faster than two of them, i.e. GRMC(ADMM) and GRBMC.

Finally, we display in Figure 1 the evolution of the loss function along iterations of the proposed Algorithm 1. This confirms the validity of the proposed

minimization method.

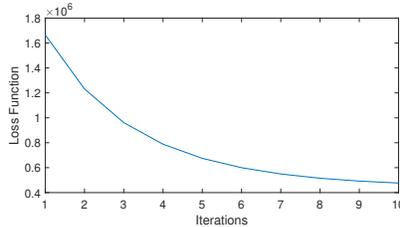


Figure 1: Loss function evolution across iterations of DPGMF-2L for CV1 setting.

5.5 Monkeypox Prediction

Globally the number of monkeypox cases have risen to 53,000⁵. It has spread to 106 countries. Monkeypox virus is symptomatically similar to smallpox, albeit less severe⁶. Smallpox has been eradicated by vaccination [31]; it was empirically seen that the smallpox vaccine was able to prevent monkeypox infection by about 85%. However, since the eradication of smallpox, its vaccination programme has also been discontinued in many countries. Perhaps the reduction in vaccination is a reason for the current global monkeypox outbreak; in such a condition the only option is to depend on therapeutics for treating the patients. Furthermore an animal study from 2006 [32] showed that antiviral treatment is indeed more effective than vaccination. In this sub-section, we study how good the antiviral recommendations are from the different algorithms. We did such a comparison in our previous study [20] with coronavirus as a case study. The top 5 recommended antivirals recommended by different algorithms are shown in the following table.

Table 3: **Top-5 ranked recommendations/drugs predicted for MPV by DPMFG and the bench-marks.**

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
DPMFG	Darunavir	Bictegravir	Indinavir	Dolutegravir	Etravirine
GRMF	Bictegravir	Ganciclovir	Darunavir	Amprenavir	Fosamprenavir
GRMC	Bictegravir	Darunavir	Ganciclovir	Amprenavir	Fosamprenavir
GRBMC	Bictegravir	Darunavir	Ganciclovir	Amprenavir	Fosamprenavir
DMF	Cidofovir	Pleconaril	Baloxavir mbxl	Ibuprofen	Favipiravir
PMF	Pleconaril	Ribavirin	Tenofovir	Ibuprofen	Trifluridine

DPMFG-2L is used for comparison here as it has a less complex structure while showing better performance than DPMFG-3L. We refer to the standard treatment guideline for sever monkeypox from <https://www.hivguidelines.org/antiretroviral-therapy/monkeypox/>.

⁵https://en.wikipedia.org/wiki/2022_monkeypox_outbreak

⁶<https://www.who.int/news-room/fact-sheets/detail/monkeypox>

The drugs in bold represent the ones that are actually being used to treat monkeypox. We see that ours have correctly predicted 4 out of the 5 drugs; the other benchmarks have only predicted 2 out of 5. This clearly shows the efficacy of our algorithm compared to the others. One must note that the drug Darunavir (predicted by all three) needs to be administered with caution owing to potential adverse drug-drug- interaction effects.

Our code is available at <https://github.com/Stuti-code-dot/DGPMF-for-DVA>.

6 Conclusion

In the discipline of bioinformatics, computational analysis and methodologies facilitate the quick and effective discovery of new knowledge, technical advancements, and accurate analysis. Recently, it was suggested that there were chances of a monkeypox virus epidemic after the several waves of COVID-19. Fortunately, there has been active research in the past few years in utilization of AI to predict drug-target interactions and therefore, existing drugs can be repurposed and polypharmacology avoided, especially since the COVID-19 pandemic. Studies have shown that purposing an existing drug would qualify it for Phase II clinical trials directly [33, 34]. This work is our contribution in helping to control and cure monkeypox infections through drug repurposing. As part of this work, we propose a deep probabilistic matrix factorization framework with multiple graph regularization. The algorithm penalizes the non-interacting drug-virus pairs and promotes the interacting pairs. Also, the parameters' updates over the iterations are novel and efficient in terms of optimization and results in a stable convergence. The biological contributions of this work are the ability of the algorithm in providing drug recommendations for the monkeypox virus and ranking them w.r.t. their predicted effectiveness level.

Acknowledgements

The authors acknowledge support from Inria Saclay, through the International Inria Team program COMPASS. S.J. and E.C. acknowledge support from the European Research Council Starting Grant MAJORIS ERC-2019- STG-850925.

Supplementary

Supplementary Material.

References

- [1] J. G. Rizk, G. Lippi, B. M. Henry, D. N. Forthal, and Y. Rizk. Prevention and treatment of monkeypox. *Drugs*, pages 1–7, 2022.

- [2] E. M. Bunge, B. Hoet, L. Chen, F. Lienert, H. Weidenthaler, L. R. Baer, and R. Steffen. The changing epidemiology of human monkeypox—a potential threat? a systematic review. *PLoS Neglected Tropical Diseases*, 16(2):e0010141, 2022.
- [3] M. Bagherian, E. Sabeti, K. Wang, M. A. Sartor, Z. Nikolovska-Coleska, and K. Najarian. Machine learning approaches and databases for prediction of drug–target interaction: a survey paper. *Briefings in Bioinformatics*, 22(1):247–269, 2021.
- [4] K. Li, X. Zhou, F. Lin, W. Zeng, and G. Alterovitz. Deep probabilistic matrix factorization framework for online collaborative filtering. *IEEE Access*, 7:56117–56128, 2019.
- [5] S. Xu, C. Zhang, and J. Zhang. Bayesian deep matrix factorization network for multiple images denoising. *Neural Networks*, 123:420–428, 2020.
- [6] J. Strahl, J. Peltonen, H. Mamitsuka, and S. Kaski. Scalable probabilistic matrix factorization with graph-based priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5851–5858, 2020.
- [7] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [8] B. Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(null):3413–3430, 2011.
- [9] Q. Gu, J. Zhou, and C. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the SIAM International Conference on Data Mining*, pages 199–210, 2010.
- [10] A. Mongia and A. Majumdar. Matrix completion on multiple graphs: Application in collaborative filtering. *Signal Processing*, 165(C):144–148, 2019.
- [11] A. Mongia, V. Jain, E. Chouzenoux, and A. Majumdar. Deep latent factor model for predicting drug target interactions. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1254–1258, 2019.
- [12] A. Mongia, S. Jain, E. Chouzenoux, and A. Majumdar. Deepvir: Graphical deep matrix factorization for in silico antiviral repositioning—application to covid-19. *Journal of Computational Biology*, 29(5):441–452, 2022.
- [13] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.
- [14] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 2. Springer, 2009.

- [15] Stuti Jain, Emilie Chouzenoux, Kriti Kumar, and Angshul Majumdar. Graph regularized probabilistic matrix factorization for drug-drug interactions prediction. *bioRxiv*, 2022.
- [16] R. Socher, Y. Bengio, and C. D. Manning. Deep learning for nlp (without magic). In *Tutorial Abstracts of ACL 2012*, pages 5–5. 2012.
- [17] D. Shen, G. Wu, and H. Suk. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19:221–248, 2017.
- [18] P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H. Luigs, A. Mahlein, and K. Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.
- [19] A. Mongia and A. Majumdar. Drug-target interaction prediction using multi-graph regularized deep matrix factorization. *BioRxiv*, page 774539, 2019.
- [20] A. Mongia, S. K. Saha, E. Chouzenoux, and A. Majumdar. A computational approach to aid clinicians in selecting anti-viral drugs for covid-19 trials. *Scientific Reports*, 11(1):1–12, 2021.
- [21] M. Hattori, N. Tanaka, M. Kanehisa, and S. Goto. Simcomp/subcomp: chemical structure search servers for network analyses. *Nucleic Acids Research*, 38(suppl_2):W652–W656, 2010.
- [22] N. A. Ahlgren, J. Ren, Y. Y. Lu, J. A. Fuhrman, and F. Sun. Alignment-free oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences. *Nucleic Acids Research*, 45(1):39–53, 2017.
- [23] A. Ezzat, P. Zhao, M. Wu, X. Li, and C. Kwok. Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(3):646–656, 2017.
- [24] Y. Dai and X. Zhao. A survey on the computational approaches to identify drug targets in the postgenomic era. *BioMed Research International*, 2015.
- [25] M. Wang, C. Tang, and J. Chen. Drug-target interaction prediction via dual laplacian graph regularized matrix completion. *BioMed Research International*, 2018.
- [26] A. Mongia, D. Sengupta, and A. Majumdar. deepmc: Deep matrix completion for imputation of single-cell rna-seq data. *Journal of Computational Biology*, 27, 2019.
- [27] A. Mongia and A. Majumdar. Drug-target interaction prediction using multi graph regularized nuclear norm minimization. *PLoS ONE*, 15(1), 2018.

- [28] A. Ezzat, P. Zhao, M. Wu, X. Li, and C. Kwoh. Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(3):646–656, 2016.
- [29] A. Mongia, E. Chouzenoux, and A. Majumdar. Computational prediction of drug-disease association based on graph-regularized one bit matrix completion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.
- [30] J. Burez and D. Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636, 2009.
- [31] F. Fenner, D. A. Henderson, I. Arita, Z. Jezek, I. D. Ladnyi, et al. *Smallpox and its eradication*, volume 6. World Health Organization Geneva, 1988.
- [32] K. J. Stittelaar, J. Neyts, L. Naesens, G. Van Amerongen, R. F. Van Lavieren, A. Holý, E. De Clercq, H. G.M. Niesters, E. Fries, C. Maas, et al. Antiviral treatment is more effective than smallpox vaccination upon lethal monkeypox virus infection. *Nature*, 439(7077):745–748, 2006.
- [33] D. Paul, G. Sanap, S. Shenoy, D. Kalyane, K. Kalia, and R. K. Tekade. Artificial intelligence in drug discovery and development. *Drug Discovery Today*, 26(1):80, 2021.
- [34] K. Mak and M. R. Pichika. Artificial intelligence in drug development: present status and future prospects. *Drug Discovery Today*, 24(3):773–780, 2019.

SUPPLEMENTARY:

Deep Probabilistic Matrix Factorization on Graphs: Application to Drug Repositioning for Monkeypox

In this supplementary file, we explicit the updates for a given iteration $p \in \mathbb{N}$ of Algorithm 1 presented in the main file.

1 Update of X

The update reads:

$$X^{(p+1)} = \operatorname{argmin}_{X \in \mathbb{R}^{N \times M}} \frac{1}{2\sigma^2} \|Y - B \odot X\|_F^2 + \frac{\lambda_R}{2} \|X - U_1^{(p)} U_2^{(p)} U_3^{(p)}\|_F^2.$$

This is a strictly convex quadratic problem, whose solution satisfies the following optimality condition:

$$\frac{1}{\sigma^2} Y + \lambda_R U_1^{(p)} U_2^{(p)} U_3^{(p)} = \frac{1}{\sigma^2} B \odot X^{(p+1)} + \lambda_R X^{(p+1)}. \quad (\text{S1})$$

The above equation is a linear system that can be solved efficiently by linear conjugate gradient solver.

2 Update of U_1

The update reads:

$$U_1^{(p+1)} = \operatorname{argmin}_{U_1 \in \mathbb{R}^{N \times k_1}} \frac{1}{2} \operatorname{tr}(U_1^\top \Gamma_{U_1}^{(p)} U_1) + \frac{\lambda_R}{2} \|X^{(p+1)} - U_1 U_2^{(p)} U_3^{(p)}\|_F^2.$$

This amounts to search for $U^{(p+1)}$ satisfying

$$\Gamma_{U_1}^{(p)} U_1 + \lambda_R (U_1 U_2^{(p)} U_3^{(p)} - X^{(p+1)}) (U_2^{(p)} U_3^{(p)})^\top = 0,$$

or, equivalently,

$$\Gamma_{U_1}^{(p)} U_1 + U_1 (\lambda_R U_2^{(p)} U_3^{(p)} (U_2^{(p)} U_3^{(p)})^\top) = \lambda_R X^{(p+1)} (U_2^{(p)} U_3^{(p)})^\top. \quad (\text{S2})$$

Solving the above Sylvester equation, we get $U_1^{(p+1)}$.

3 Update of U_2

The update is:

$$U_2 = (U_1^{(p+1)})^\dagger X^{(p+1)} (U_3^{(p)})^\dagger \quad (\text{S3})$$

4 Update of U_3

The update reads:

$$U_3^{(p+1)} = \operatorname{argmin}_{U_3 \in \mathbb{R}^{k_2 \times M}} \frac{\lambda_R}{2} \|X^{(p+1)} - U_1^{(p+1)} U_2^{(p+1)} U_3\|_F^2 + \frac{1}{2} \operatorname{tr}(U_3 \Gamma_{U_3}^{(p)} U_3^\top)$$

This amounts to search for $U_3^{(p+1)}$ satisfying

$$\lambda_R(U_1^{(p+1)}U_2^{(p+1)}U_3 - X^{(p+1)})U_1^{(p+1)}U_2^{(p+1)} + U_3\Gamma_{U_3}^{(p)} = 0,$$

or, equivalently,

$$\lambda_R(U_1^{(p+1)}U_2^{(p+1)})^\top U_1^{(p+1)}U_2^{(p+1)}U_3 + U_3\Gamma_{U_3}^{(p)} = \lambda_R X^{(p+1)}U_1^{(p+1)}U_2^{(p+1)} \quad (\text{S4})$$

Solving the above Sylvester equation, we get $U_3^{(p+1)}$.

5 Update of Γ_{U_1}

We must solve:

$$\begin{aligned} \Gamma_{U_1}^{(p+1)} = \operatorname{argmin}_{\Gamma_{U_1} \in S_N} & \operatorname{tr}(U_1^{(p+1)}(U_1^{(p+1)})^\top \Gamma_{U_1}) + \mathcal{L}(\Gamma_{U_1}) \\ & + 2\lambda_{U_1} \sum_{i,j \in \bar{\mathcal{E}}_{U_1}} |[\Gamma_{U_1}]_{ij}| - 2\lambda_{U_1} \sum_{i,j \in \mathcal{E}_{U_1}} \ln(|[\Gamma_{U_1}]_{ij}| + \delta) + \lambda_{U_1} \|\Gamma_{U_1}\|_F^2. \end{aligned}$$

Let us split the cost function involved above into two terms, so that

$$\Gamma_{U_1}^{(p+1)} = \operatorname{argmin}_{\Gamma_{U_1} \in S_N} f(\Gamma_{U_1}) + g(\Gamma_{U_1}),$$

with

$$(\forall U_1 \in S_N) \quad f(\Gamma_{U_1}) = \operatorname{tr}(U_1^{(p+1)}(U_1^{(p+1)})^\top \Gamma_{U_1}) + \mathcal{L}(\Gamma_{U_1}),$$

and

$$(\forall U_1 \in S_N) \quad g(\Gamma_{U_1}) = 2\lambda_{U_1} \sum_{i,j \in \bar{\mathcal{E}}_{U_1}} |[\Gamma_{U_1}]_{ij}| - 2\lambda_{U_1} \sum_{i,j \in \mathcal{E}_{U_1}} \ln(|[\Gamma_{U_1}]_{ij}| + \delta) + \lambda_{U_1} \|\Gamma_{U_1}\|_F^2.$$

The minimization of $f + g$ does not have a close form solution and an inner solver is required. Function f is convex, differentiable on S_N^{++} , while g is non-convex, non-differentiable. Luckily, function g is separable over the entries of Γ_{U_1} , that is:

$$(\forall U_1 \in S_N) \quad g(\Gamma_{U_1}) = \sum_{1 \leq i,j \leq N} g_{ij}([\Gamma_{U_1}]_{ij})$$

with, for every $(i, j) \in \{1, \dots, N\}^2$,

$$(\forall \omega \in \mathbb{R}) \quad g_{ij}(\omega) = \begin{cases} 2\lambda_{U_1}|\omega| + \lambda_{U_1}\omega^2 & \text{if } (i, j) \in \bar{\mathcal{E}}_{U_1}, \\ -2\lambda_{U_1} \ln(|\omega| + \delta) + \lambda_{U_1}\omega^2 & \text{if } (i, j) \in \mathcal{E}_{U_1}. \end{cases}$$

We thus opt for running $L \geq 1$ iterations of a proximal gradient algorithm, initialized in $\operatorname{dom} f = S_N^{++}$. This reads as follows:

$$\begin{aligned} \Gamma_{U_1}^0 &= \Gamma_{U_1}^{(p)} \\ \text{For } \ell &= 1, 2, \dots, L \\ \tilde{\Gamma}_{U_1}^\ell &= \Gamma_{U_1}^\ell - \theta_1 \nabla f(\Gamma_{U_1}^\ell) \\ \Gamma_{U_1}^{\ell+1} &= \operatorname{prox}_{\theta_1 g}(\tilde{\Gamma}_{U_1}^\ell) \\ \Gamma_{U_1}^{(p+1)} &= \Gamma_{U_1}^L. \end{aligned} \quad (\text{S5})$$

Hereabove, θ_1 is a positive stepsize such that all iterates belong to the domain of f . The expression for the gradient of f and the proximity operator of g can be found in [15].

6 Update of Γ_{U_3}

The update reads:

$$\begin{aligned} \Gamma_{U_3}^{(p+1)} = \operatorname{argmin}_{\Gamma_{U_3} \in S_M} & \frac{1}{2} \operatorname{tr}(U_3^{p+1} \Gamma_{U_3} (U_3^{p+1})^\top) + \frac{1}{2} \mathcal{L}(\Gamma_{U_3}) \\ & + \lambda_{U_3} \sum_{i,j \in \bar{\mathcal{E}}_{U_3}} |[\Gamma_{U_3}]_{ij}| - \lambda_{U_3} \sum_{i,j \in \mathcal{E}_{U_3}} \ln(|[\Gamma_{U_3}]_{ij}| + \delta) + \frac{\lambda_{U_3}}{2} \|\Gamma_{U_3}\|_F^2 \end{aligned}$$

Let us split the cost function involved above into two terms, so that

$$\Gamma_{U_3}^{(p+1)} = \operatorname{argmin}_{\Gamma_{U_3} \in S_M} f(\Gamma_{U_3}) + g(\Gamma_{U_3}),$$

with

$$(\forall U_3 \in S_M) \quad f(\Gamma_3) = \operatorname{tr}(U_3^{(p+1)} \Gamma_{U_3} (U_3^{(p+1)})^\top) + \mathcal{L}(\Gamma_{U_3}),$$

and

$$(\forall \Gamma_{U_3} \in S_M) \quad g(\Gamma_{U_3}) = 2\lambda_{U_3} \sum_{i,j \in \bar{\mathcal{E}}_{U_3}} |[\Gamma_{U_3}]_{ij}| - 2\lambda_{U_3} \sum_{i,j \in \mathcal{E}_{U_3}} \ln(|[\Gamma_{U_3}]_{ij}| + \delta) + \lambda_{U_3} \|\Gamma_{U_3}\|_F^2.$$

Hereagain, we can run $L \geq 1$ iterations of a proximal gradient algorithm, initialized in $\operatorname{dom} f = S_M^{++}$, which reads as follows:

$$\begin{aligned} \Gamma_{U_3}^0 &= \Gamma_{U_3}^{(p)} \\ \text{For } \ell &= 1, 2, \dots, L \\ \tilde{\Gamma}_{U_3}^\ell &= \Gamma_{U_3}^\ell - \theta_2 \nabla f(\Gamma_{U_3}^\ell) \\ \Gamma_{U_3}^{\ell+1} &= \operatorname{prox}_{\theta_2 g}(\tilde{\Gamma}_{U_3}^\ell) \\ \Gamma_{U_3}^{(p+1)} &= \Gamma_{U_3}^L, \end{aligned} \tag{S6}$$

with θ_2 a positive stepsize such that all iterates belong to the domain of f . The expression for the gradient of f and the proximity operator of g can be found in [15].