Preprint. Pages: 1–26

Spectral Geometric Matrix Completion

Amit Boyarski* AMITBOY@CS.TECHNION.AC.IL
Technion, Israel.

Sanketh Vedula* SANKETH@CS.TECHNION.AC.IL
Technion, Israel.

Alex Bronstein BRON@CS.TECHNION.AC.IL
Technion, Israel.

Abstract

Deep Matrix Factorization (DMF) is an emerging approach to the problem of matrix completion. Recent works have established that gradient descent applied to a DMF model induces an implicit regularization on the rank of the recovered matrix. In this work we interpret the DMF model through the lens of spectral geometry. This allows us to incorporate explicit regularization without breaking the DMF structure, thus enjoying the best of both worlds. In particular, we focus on matrix completion problems with underlying geometric or topological relations between the rows and/or columns. Such relations are prevalent in matrix completion problems that arise in many applications, such as recommender systems and drug-target interaction. Our contributions enable DMF models to exploit these relations, and make them competitive on real benchmarks, while exhibiting one of the first successful applications of deep linear networks.

Keywords: matrix completion, deep matrix factorization, deep linear networks, graph signal processing, spectral geometry, recommendation system, drug-target interaction.

1. Introduction

Matrix completion deals with the recovery of missing values of a matrix from a subset of its entries,

$$\text{Find } \boldsymbol{X} \text{ s.t. } \boldsymbol{X} \odot \boldsymbol{S} = \boldsymbol{M} \odot \boldsymbol{S}. \tag{1}$$

Here $\boldsymbol{X}$ stands for the unknown matrix, $\boldsymbol{M} \in \mathbb{R}^{m \times n}$ for the ground truth matrix, $\boldsymbol{S}$ is a binary mask representing the input support, and $\odot$ denotes the Hadamard product. Since problem (1) is ill-posed, it is common to assume that $\boldsymbol{M}$ belongs to some low dimensional subspace. Under this assumption, the matrix completion problem can be cast via the least-squares variant,

$$\min_{\boldsymbol{X}} \operatorname{rank}(\boldsymbol{X}) + \frac{\mu}{2} \|(\boldsymbol{X} - \boldsymbol{M}) \odot \boldsymbol{S}\|_F^2 . \tag{2}$$

Relaxing the intractable rank penalty to its convex envelope, namely the nuclear norm, leads to a convex problem whose solution coincides with (2) under some technical conditions (Candès and Recht, 2009). Another way to enforce low rank is by explicitly parametrizing $\boldsymbol{X}$ in factorized form, $\boldsymbol{X} = \boldsymbol{X}_1 \boldsymbol{X}_2$. The rank is upper-bounded by the minimal dimension of $\boldsymbol{X}_1, \boldsymbol{X}_2$. Further developing this idea, $\boldsymbol{X}$ can be parametrized as a product of several matrices $\boldsymbol{X} = \prod_{i=1}^{N} \boldsymbol{X}_i$, a model we denote as *deep matrix factorization* (DMF). Following

the nomenclature used by Arora et al. (2019), a factorization with $N > 2$ is called *deep* while for $N \leq 2$ it is called *shallow*. Nevertheless, to simplify notation, we shall simply call any such model as DMF while explicitly specifying N. Gunasekar et al. (2017); Arora et al. (2019) investigated the minimization of overparametrized DMF models using gradient descent, and came to the following conclusion (which we will formally state in Section 2): whereas in some restrictive settings minimizing DMF using gradient descent is equivalent to nuclear norm minimization (i.e., convex relaxation of (2)), in general these two models produce different results, with the former enforcing a stronger regularization on the rank of $\boldsymbol{X}$. This regularization gets stronger as N (the depth) increases. In light of these results, we shall henceforth refer by "DMF" to the aforementioned model coupled with the specific algorithm used for its minimization, namely, gradient descent.

In this work we focus on matrix completion problems with underlying geometric or topological relations between the rows and/or columns. Such relations are prevalent in matrix completion problems that arise in applications such as recommender systems and drug-target interaction. A common way of representing these relations is in the form of a graph. For example, in the Netflix problem (Candès and Recht, 2009) the rows correspond to users, the columns correspond to movies, and the matrix elements represent the ratings given by the users to the movies. The goal is to recover the full matrix of ratings from an incomplete matrix. In this setting there might exist side information on the users and movies which can be used to construct two graphs representing relations between users and relations between movies, and the matrix $\boldsymbol{X}$ can be viewed as a signal on the *product* of these graphs. A useful prior on $\boldsymbol{X}$ can be, for example, modeling it as a smooth or band-limited signal on this graph, encouraging similar movies to be assigned similar rating from similar users, and vice versa.

This kind of geometric structure is generally overlooked by purely algebraic entities such as rank, and becomes invaluable in the data poor regime, where the theorems governing reconstruction guarantees (i.e., Candès and Recht (2009)) do not hold. Our work leverages the recent advances in DMF theory to marry the two concepts: a framework for matrix completion that is *explicitly* motivated by geometric considerations, while *implicitly* promoting low-rank via its DMF structure.

Contributions. Our contributions are as follows:

- We propose geometrically inspired DMF models for matrix completion and study their dynamics.
- We successfully apply those models to matrix completion problems in recommendation systems and drug-target interaction, outperforming various complicated methods with only a few lines of code (Figure 6). This serves as an example to the power of deep linear networks, being one of their first successful applications to real problems.
- Our findings challenge the quality of the side information available in recommendation systems datasets, and the ability of contemporary methods to utilize it in a meaningful and efficient way.

2. Preliminaries

Spectral graph theory. Let $\mathcal{G} = (V, E, \boldsymbol{\Omega})$ be a (weighted) graph specified by its vertex set V and edge set E, with its adjacency matrix denoted by $\boldsymbol{\Omega}$. Given a function $\boldsymbol{x} \in \mathbb{R}^{|V|}$ on the vertices, we define the following quadratic form (also known as *Dirichlet energy*) measuring the variability of the function $\boldsymbol{x}$ on the graph,

$$\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \sum_{(a,b)\in E} \omega_{a,b} \left(x(a) - x(b)\right)^2. \tag{3}$$

The matrix $\boldsymbol{L}$ is called the *(combinatorial) graph Laplacian*, and is given by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{\Omega}$, where $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{\Omega}\mathbf{1})$ is the *degree matrix*, with $\mathbf{1}$ denoting the vector of all ones. $\boldsymbol{L}$ is symmetric and positive semi-definite and therefore admits a spectral decomposition $\boldsymbol{L} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top$. Since $\boldsymbol{L}\mathbf{1} = \mathbf{0}$, $\lambda_1 = 0$ is always an eigenvalue of $\boldsymbol{L}$. The graph Laplacian is a discrete generalization of the continuous Laplace-Beltrami operator, and therefore has similar properties. One can think of the eigenpairs $(\boldsymbol{\phi}_i, \lambda_i)$ as the graph analogues of "harmonic" and "frequency". Structural information about the graph is encoded in the spectrum of the Laplacian. For example, the number of connected components in the graph is given by the multiplicity of the zero eigenvalue, and the second eigenvalue (counting multiple eigenvalues separately) is a measure for the connectivity of the graph (Spielman, 2009).

A function $\boldsymbol{x} = \sum_{i=1}^{|V|} \alpha_i \boldsymbol{\phi}_i$ on the vertices of the graph whose coefficients α_i are small for large i, demonstrates a "smooth" behaviour on the graph in the sense that the function values on nearby nodes will be similar. A standard approach to promoting such smooth functions on graphs is by using the Dirichlet energy (3) to regularize some loss term. For example, this approach gives rise to the popular bilateral and non-local means filters (Singer et al., 2009; Gadde et al., 2013). We call $\boldsymbol{x}$ a k-bandlimited signal on the graph $\mathcal{G}$ if $\alpha_i = 0 \; \forall \; i > k$.

Product graphs and functional maps. Let $\mathcal{G}_1 = (V_1, E_1, \boldsymbol{\Omega}_1)$, $\mathcal{G}_2 = (V_2, E_2, \boldsymbol{\Omega}_2)$ be two graphs, with $\boldsymbol{L}_1 = \boldsymbol{\Phi}\boldsymbol{\Lambda}_1\boldsymbol{\Phi}^\top$, $\boldsymbol{L}_2 = \boldsymbol{\Psi}\boldsymbol{\Lambda}_2\boldsymbol{\Psi}^\top$ being their corresponding graph Laplacians. The bases $\boldsymbol{\Phi}, \boldsymbol{\Psi}$ can be used to represent functions on these graphs. We define the Cartesian product of $\mathcal{G}_1$ and $\mathcal{G}_2$, denoted by $\mathcal{G}_1 \square \mathcal{G}_2$, as the graph with vertex set $V_1 \times V_2$, on which two nodes $(u, u'), (v, v')$ are adjacent if either $u = v$ and $(u', v') \in E_2$ or $u' = v'$ and $(u, v) \in E_1$. The Laplacian of $\mathcal{G}_1 \square \mathcal{G}_2$ is given by the tensor sum of $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$,

$$\boldsymbol{L}_{\mathcal{G}_1 \square \mathcal{G}_2} = \boldsymbol{L}_1 \oplus \boldsymbol{L}_2 = \boldsymbol{L}_1 \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{L}_2, \tag{4}$$

and its eigenvalues are given by the Cartesian sum of the eigenvalues of $\boldsymbol{L}_1, \boldsymbol{L}_2$, i.e., all combinations $\lambda_1 + \lambda_2$ where λ_1 is an eigenvalue of $\boldsymbol{L}_1$ and λ_2 is an eigenvalue of $\boldsymbol{L}_2$. Let $\boldsymbol{X}$ be a function defined on $\mathcal{G}_1 \square \mathcal{G}_2$. Then it can be represented using the bases $\boldsymbol{\Phi}, \boldsymbol{\Psi}$ of the individual Laplacians, $\boldsymbol{C} = \boldsymbol{\Phi}^\top \boldsymbol{X} \boldsymbol{\Psi}$. In the shape processing community, such $\boldsymbol{C}$ is called a *functional map* (Ovsjanikov et al., 2012), as it it used to map between the functional spaces of $\mathcal{G}_1$ and $\mathcal{G}_2$. For example, given two functions, $\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{\alpha}$ on $\mathcal{G}_1$ and $\boldsymbol{y} = \boldsymbol{\Psi}\boldsymbol{\beta}$ on $\mathcal{G}_2$, one can use $\boldsymbol{C}$ to map between their representations $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, i.e., $\boldsymbol{\alpha} = \boldsymbol{\Phi}^\top \boldsymbol{x} = \boldsymbol{C}\boldsymbol{\Psi}^\top \boldsymbol{y} = \boldsymbol{C}\boldsymbol{\beta}$. We shall henceforth interchangeably switch between the terms "signal on the product graph" and "functional map".

We will call a functional map *smooth* if it maps close points on one graph to close points on the other. A simple way to construct a smooth map is via a linear combination of

eigenvectors of $\boldsymbol{L}_{\mathcal{G}_1 \square \mathcal{G}_2}$ corresponding to small eigenvalues ("low frequencies"). Notice that while the singular vectors of $\boldsymbol{L}_{\mathcal{G}_1 \square \mathcal{G}_2}$ are outer products of the columns of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, their ordering with respect to the eigenvalues of $\boldsymbol{L}_{\mathcal{G}_1 \square \mathcal{G}_2}$ might be different than their lexicographic order.

Implicit regularization of DMF. Let $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ be a matrix parametrized as a product of N matrices $\boldsymbol{X} = \prod_{i=1}^{N} \boldsymbol{X}_i$ (which can be interpreted as N linear layers of a neural network), and let $\ell(\boldsymbol{X})$ be an analytic loss function. We are interested in the following optimization problem,

$$\min_{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N} \ell \left(\prod_{i=1}^{N} \boldsymbol{X}_i \right). \tag{5}$$

Without loss of generality, we will assume that $m < n$. Arora et al. (2018, 2019) analyzed the evolution of the singular values and singular vectors of $\boldsymbol{X}$ throughout the gradient flow $\dot{\boldsymbol{X}}(t) = -\nabla \ell(\boldsymbol{X}(t))$, i.e., gradient descent with an infinitesimal step size, with *balanced initialization*,

$$\boldsymbol{X}_{i+1}(0)^\top \boldsymbol{X}_{i+1}(0) = \boldsymbol{X}_i(0) \boldsymbol{X}_i(0)^\top, \quad \forall i = 1 \ldots N. \tag{6}$$

As a first step, we state that $\boldsymbol{X}(t)$ admits an analytic singular value decomposition.

Lemma 1 *(Lemma 1 in Arora et al. (2019)). The product matrix $\boldsymbol{X}(t)$ can be expressed as:*

$$\boldsymbol{X}(t) = \boldsymbol{U}(t) \boldsymbol{S}(t) \boldsymbol{V}^\top(t), \tag{7}$$

where: $\boldsymbol{U}(t) \in \mathbb{R}^{m \times m}$, $\boldsymbol{S}(t) \in \mathbb{R}^{m \times m}$ and $\boldsymbol{V}(t) \in \mathbb{R}^{n \times m}$ are analytic functions of t; and for every t, the matrices $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$ have orthonormal columns, while $\boldsymbol{S}(t)$ is diagonal (elements on its diagonal may be negative and may appear in any order).

The diagonal elements of $\boldsymbol{S}(t)$, which we denote by $\sigma_1(t), \ldots, \sigma_m(t)$, are signed singular values of $\boldsymbol{X}(t)$; the columns of $\boldsymbol{U}(t)$ and $\boldsymbol{V}(t)$, denoted $u_1(t), \ldots, u_m(t)$ and $v_1(t), \ldots, v_m(t)$, are the corresponding left and right singular vectors (respectively). Using the above lemma, Arora et al. (2019) characterized the evolution of singular values as follows:

Theorem 2 *(Theorem 3 in Arora et al. (2019)). The signed singular values of the product matrix $\boldsymbol{X}(t)$ evolve by:*

$$\begin{aligned} &\dot{\sigma}_r(t) = -N \cdot \left(\sigma_r^2(t)\right)^{1-1/N} \cdot \langle \nabla \ell(\boldsymbol{X}(t)), \mathbf{u}_r(t) \mathbf{v}_r^\top(t) \rangle, \\ &r = 1, \ldots, m. \end{aligned} \tag{8}$$

If the matrix factorization is non-degenerate, i.e., has depth $N \geq 2$, the singular values need not be signed (we may assume $\sigma_r(t) \geq 0$ for all t).

The above theorem implies that the evolution rates of the singular values are dependent on their magnitude exponentiated by $2 - 2/N$. Ignoring the term $\langle \nabla \ell(\boldsymbol{X}(t)), \mathbf{u}_r(t) \mathbf{v}_r^\top(t) \rangle$, as N increases,the evolution rate of the large singular values is enhanced while the evolution rate of the small ones is dampened. The increasing gap between the evolution rates of the large and small singular values induces an implicit regularization on the effective rank of $\boldsymbol{X}(t)$. However, the evolution of the singular values also depends on the gradient of the loss function via the term $\langle \nabla \ell(\boldsymbol{X}(t)), \mathbf{u}_r(t) \mathbf{v}_r^\top(t) \rangle$, making the choice of the loss consequential. While exact analysis of (8) is hard in the absence of perfect characterization of this term, one can still leverage these dynamics by empirically exploring different loss functions.

3. Spectral geometric matrix completion

We assume that we are given a set of samples from the unknown matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$, encoded as a binary mask $\boldsymbol{S}$, and two graphs $\mathcal{G}_{\mathrm{r}}, \mathcal{G}_{\mathrm{c}}$, encoding relations between the rows and the columns, respectively. Denote the Laplacians of these graphs and their spectral decompositions by $\boldsymbol{L}_{\mathrm{r}} = \boldsymbol{\Phi}\boldsymbol{\Lambda}_{\mathrm{r}}\boldsymbol{\Phi}^\top$, $\boldsymbol{L}_{\mathrm{c}} = \boldsymbol{\Psi}\boldsymbol{\Lambda}_{\mathrm{c}}\boldsymbol{\Psi}^\top$. We denote the Cartesian product between $\mathcal{G}_{\mathrm{c}}$ and $\mathcal{G}_{\mathrm{r}}$ by $\mathcal{G} \equiv \mathcal{G}_{\mathrm{c}} \square \mathcal{G}_{\mathrm{r}}$, and will henceforth refer to it as our reference graph. Our approach relies on a minimization problem of the form

$$\begin{aligned} &\min_{\boldsymbol{X}} \; E_{\mathrm{data}}(\boldsymbol{X}) + \mu E_{\mathrm{dir}}(\boldsymbol{X}) \\ &\text{s.t.} \quad \operatorname{rank}(\boldsymbol{X}) \leq r, \end{aligned} \tag{9}$$

with E_{data} denoting a data term of the form

$$E_{\mathrm{data}}(\boldsymbol{X}) = \|(\boldsymbol{X} - \boldsymbol{M}) \odot \boldsymbol{S}\|_F^2 \,, \tag{10}$$

and E_{dir} is the Dirichlet energy of $\boldsymbol{X}$ on $\mathcal{G}$, given by (see (4))[1]

$$E_{\mathrm{dir}}(\boldsymbol{X}) = \operatorname{tr}\left(\boldsymbol{X}^\top \boldsymbol{L}_{\mathrm{r}} \boldsymbol{X}\right) + \operatorname{tr}\left(\boldsymbol{X}\boldsymbol{L}_{\mathrm{c}}\boldsymbol{X}^\top\right). \tag{11}$$

To that end, we parametrize $\boldsymbol{X}$ via a matrix product $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{Z}\boldsymbol{B}^\top$, and discard the rank constraint,

$$\min_{\boldsymbol{A},\boldsymbol{Z},\boldsymbol{B}} \; E_{\mathrm{data}}(\boldsymbol{A}\boldsymbol{Z}\boldsymbol{B}^\top) + \mu E_{\mathrm{dir}}\left(\boldsymbol{A}\boldsymbol{Z}\boldsymbol{B}^\top\right). \tag{12}$$

Since (12) is now a DMF model of the form (5), according to Theorem 2 the discarded rank constraint will be captured by the implicit regularization induced by gradient descent even if the factors are full size matrices. To emphasize this point, in most of our experiments we used $\boldsymbol{A}$ of size $m \times m$, $\boldsymbol{Z}$ of size $m \times n$ and $\boldsymbol{B}$ of size $n \times n$.

To interpret this matrix factorization geometrically, we interpret $\boldsymbol{Z}$ as a signal living on a latent product graph $\mathcal{G}'$. Via the linear transformation $\boldsymbol{A}\boldsymbol{Z}\boldsymbol{B}^\top$ this signal is transported onto the reference graph $\mathcal{G}$, where it is assumed to be both low-rank and smooth (see Figure 1). Notice that the latent graph is used only for the purpose of illustrating the geometric interpretation, and there is no need to find it explicitly. Nevertheless, it is possible to promote particular properties of it via spectral constraints that can sometime improve the performance. We demonstrate these extensions in the sequel.

To give a concrete example, suppose $\boldsymbol{X}$ is a band-limited signal on a 2D Euclidean grid $\mathcal{G}$, then there is some low rank signal $\boldsymbol{Z}$ that can be made smooth on $\mathcal{G}$ via an appropriate ordering of its rows and columns[2], i.e., $\boldsymbol{X} = \boldsymbol{\Pi}_1 \boldsymbol{Z} \boldsymbol{\Pi}_2^\top$. By smooth we mean that it has low Dirichlet energy (3), where $\boldsymbol{L}$ is the discrete 2D Euclidean Laplacian, i.e., $\operatorname{tr}\left(\boldsymbol{X}^\top \boldsymbol{L} \boldsymbol{X}\right) = \sum_{i,j} \left(x_{i+1,j} - x_{i,j}\right)^2 + \left(x_{i,j+1} - x_{i,j}\right)^2$.

1. Note that it is possible to weigh the two terms differently, as we do in some of our experiments.
2. On a side-note, that is exactly the goal of the well known and closely related seriation problem (Recanati, 2018).

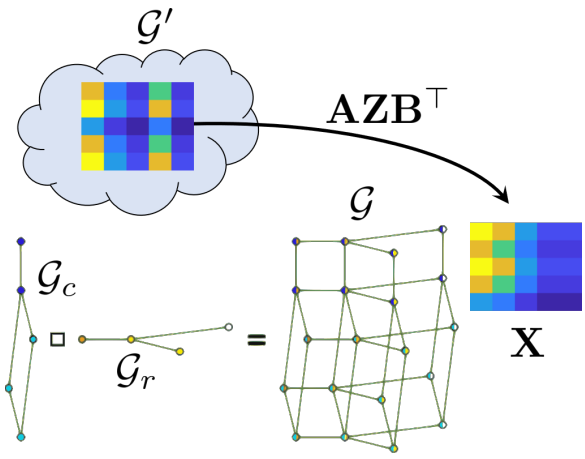


Figure 1: An illustration of the geometric interpretation of (12). A low-rank signal $\boldsymbol{Z}$ that lives on a latent product graph $\mathcal{G}'$ is transported onto the reference product graph $\mathcal{G}$. The transported signal $\boldsymbol{AZB}^\top$ will be smooth on the target graph due to the Dirichlet energy.

For later reference let us rewrite (12) in the spectral domain. We will denote the Laplacians of the latent graph factors comprising $\mathcal{G}$ by $\boldsymbol{L}'_{\mathrm{r}}, \boldsymbol{L}'_{\mathrm{c}}$ and their eigenbases by $\boldsymbol{\Phi}', \boldsymbol{\Psi}'$. Using those eigenbases and the eigenbases of the reference Laplacians $\boldsymbol{L}_{\mathrm{r}}, \boldsymbol{L}_{\mathrm{c}}$, we can write,

$$\boldsymbol{Z} = \boldsymbol{\Phi}' \boldsymbol{C} \boldsymbol{\Psi}'^\top \tag{13}$$

$$\boldsymbol{A} = \boldsymbol{\Phi} \boldsymbol{P} \boldsymbol{\Phi}'^\top \tag{14}$$

$$\boldsymbol{B} = \boldsymbol{\Psi} \boldsymbol{Q} \boldsymbol{\Psi}'^\top. \tag{15}$$

Under this reparametrization we get

$$\boldsymbol{AZB}^\top = \boldsymbol{\Phi} \boldsymbol{PCQ}^\top \boldsymbol{\Psi}^\top. \tag{16}$$

With some abuse of notation, (12) becomes

$$\min_{\boldsymbol{P},\boldsymbol{C},\boldsymbol{Q}} \; E_{\mathrm{data}}(\boldsymbol{PCQ}^\top) + \mu E_{\mathrm{dir}}\left(\boldsymbol{PCQ}^\top\right), \tag{17}$$

with

$$E_{\mathrm{data}}(\boldsymbol{PCQ}^\top) = \left\| \left(\boldsymbol{\Phi} \boldsymbol{PCQ}^\top \boldsymbol{\Psi}^\top - \boldsymbol{M} \right) \odot \boldsymbol{S} \right\|_F^2, \tag{18}$$

and

$$\begin{aligned} E_{\mathrm{dir}}(\boldsymbol{PCQ}^\top) = \mathrm{tr}\left(\boldsymbol{QC}^\top \boldsymbol{P}^\top \boldsymbol{\Lambda}_{\mathrm{r}} \boldsymbol{PCQ}^\top\right) \\ + \mathrm{tr}\left(\boldsymbol{PCQ}^\top \boldsymbol{\Lambda}_{\mathrm{c}} \boldsymbol{QC}^\top \boldsymbol{P}^\top\right). \end{aligned} \tag{19}$$

3.1. Extensions

Additional regularization via spectral filtering. We propose a stronger explicit regularization by demanding that both $\boldsymbol{X}$ and $\boldsymbol{Z}$ be smooth on their respective graphs. Since we do not know the Laplacian of $\mathcal{G}'$, we smooth $\boldsymbol{Z}$ via *spectral filtering*, i.e., through direct manipulation of its spectral representation $\boldsymbol{C}$. To that end, we pass $\boldsymbol{C}$ through a bank of pre-chosen spectral filters $\{\boldsymbol{F}_p\}_{p\in\mathcal{P}}, \{\boldsymbol{G}_q\}_{q\in\mathcal{Q}}$, i.e., diagonal positive semi-definite matrices, and transport the filtered signals to $\mathcal{G}$ according to

$$\boldsymbol{X}_{p,q} = \boldsymbol{\Phi}\boldsymbol{P}\boldsymbol{F}_p\boldsymbol{C}\boldsymbol{G}_q^\top\boldsymbol{Q}^\top\boldsymbol{\Psi}^\top, \quad p\in\mathcal{P}, q\in\mathcal{Q}. \tag{20}$$

In particular, we use the following filters,

$$\boldsymbol{F}_p = \operatorname{diag}(\mathbf{1}_p), \ \boldsymbol{G}_q = \operatorname{diag}(\mathbf{1}_q), \tag{21}$$

where $\mathbf{1}_k = \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \end{bmatrix}^\top$ denotes a vector with k ones followed by zeros. For these manipulations to take effect, we replace E_{data} in (17) with the following loss function,

$$E_{\text{z}}(\boldsymbol{P},\boldsymbol{Z},\boldsymbol{Q}) = \sum_{\substack{p\in\mathcal{P}\\ q\in\mathcal{Q}}} \|(\boldsymbol{X}_{p,q}-\boldsymbol{M})\odot\boldsymbol{S}\|_F^2. \tag{22}$$

Despite the fact that we used separable filters in (20), these filters are coupled through the loss (22). This results in an overall inseparable spectral filter that still retains a DMF structure, since (20) is a 5-layer DMF with two fixed layers. While Theorem 2 does not cover the case of a multi-layer DMF where only a subset of the layers are trainable, our empirical evaluations suggest that the implicit rank regularization is still in place. This additional regularization allows us to get decent reconstruction errors even when the number of measurements is extremely small, as we show in Section 5.1.

Regularization of the individual layers. Another extension we explore is imposing further regularization on the individual layers. For example, one could ask $\boldsymbol{L}'_{\text{r}}$ and $\boldsymbol{A}^\top\boldsymbol{L}_{\text{r}}\boldsymbol{A}$ to be jointly diagonalized by $\boldsymbol{\Phi}'$. Using (13)-(14) we get,

$$\boldsymbol{\Lambda}'_{\text{r}} = \boldsymbol{\Phi}'^\top\boldsymbol{A}^\top\boldsymbol{L}_{\text{r}}\boldsymbol{A}\boldsymbol{\Phi}' = \boldsymbol{P}^\top\boldsymbol{\Phi}^\top\boldsymbol{L}_{\text{r}}\boldsymbol{\Phi}\boldsymbol{P} = \boldsymbol{P}^\top\boldsymbol{\Lambda}_{\text{r}}\boldsymbol{P}. \tag{23}$$

Thus, we can approximately enforce this constraint with the following penalty term,

$$E^r_{\text{diag}}(\boldsymbol{P}) = \|\operatorname{off}\left(\boldsymbol{P}^\top\boldsymbol{\Lambda}_{\text{r}}\boldsymbol{P}\right)\|_F^2, \tag{24}$$

where $\operatorname{off}(\cdot)$ denotes the off-diagonal elements. A similar treatment to the columns graph gives,

$$E^c_{\text{diag}}(\boldsymbol{Q}) = \|\operatorname{off}\left(\boldsymbol{Q}^\top\boldsymbol{\Lambda}_{\text{c}}\boldsymbol{Q}\right)\|_F^2. \tag{25}$$

While these penalty terms are not a function of the product matrix, their inclusion did not harm the implicit regularization.

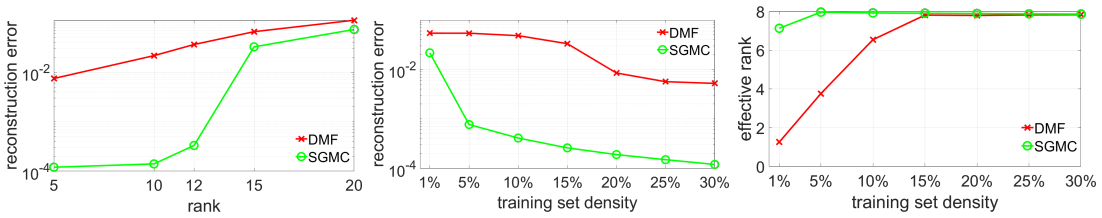


Figure 2: In these experiments we generated band-limited (low-rank and smooth) matrices using the synthetic Netflix graphs (see Figure 11) to test the dependence of SGMC and DMF on the rank of the underlying matrix and on the number of training samples. **Left**: reconstruction error (on the test set) vs. the rank of the ground-truth matrix. As the rank increases, the reconstruction error increases, but it increases slower for SGMC than for DMF. For the training set we used 15% of the points chosen at random (same training set for all experiments). μ was set to 0.001. **Middle**: reconstruction error (on the test set) vs. density of the sampling set in % of the number of matrix elements, for a random rank 10 matrix of size 150×200. As we increase the number of samples, the gap between DMF and SGMC reduces. Still, even when using 30% of the samples, SGMC performs better for the same number of iterations. For all the experiments we set $\mu = 0.01$, $lr = 0.001$, `maxiter`$=3 \times 10^6$. **Right**: effective rank (Roy and Vetterli, 2007) vs. training set density, for a random rank 10 matrix. Even for extremely data-poor regimes, SGMC was able to recover the effective rank of the ground-truth matrix, whereas is underestimating it.

4. Experimental study on synthetic data

The goal of this section is to compare between our approach and vanilla DMF on a simple example of a community structured graph. We exhaustively compare between the following distinct methods:

- **Deep matrix factorization (DMF):**

$$\min_{\boldsymbol{P},\boldsymbol{C},\boldsymbol{Q}} \left\| \left(\boldsymbol{P}\boldsymbol{C}\boldsymbol{Q}^\top - \boldsymbol{M} \right) \odot \boldsymbol{S} \right\|_F^2, \tag{26}$$

- **Spectral geometric matrix completion (SGMC):** The proposed approach defined by the optimization problem (17).

- **Functional Maps (FM, SGMC1):** This method is like SGMC with a single layer, i.e., we optimize only for $\boldsymbol{C}$, while $\boldsymbol{P}$ and $\boldsymbol{Q}$ are set to identity.

Since SGMC uses additional information, it is expected to perform better than DMF. However, proper utilization of the graph information is not trivial (as is evident by Table 1), and we conduct this set of controlled experiments to attest for it.

We use the graphs taken from the synthetic Netflix dataset. Synthetic Netflix is a small synthetic dataset constructed by Kalofolias et al. (2014) and Monti et al. (2017), in which the user and item graphs have strong communities structure. See Figure 11 in Appendix A for a visualization of the user/item graphs. It is useful in conducting controlled experiments to understand the behavior of geometry-exploiting algorithms. In all our tests we use a randomly generated band-limited matrix on the product graph $\mathcal{G}_\mathrm{c} \square \mathcal{G}_\mathrm{r}$. For the complete details please refer to the captions of the relevant figures.

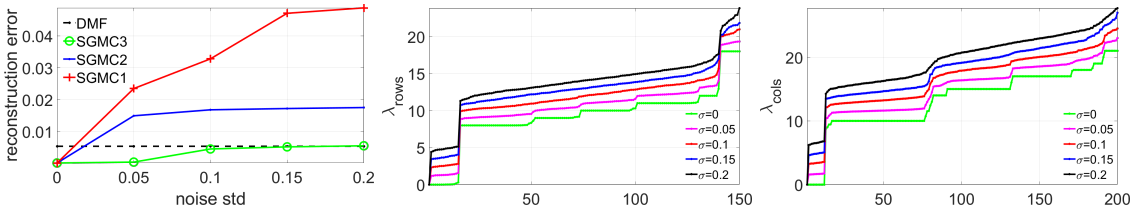


Figure 3: In this experiment, we study the robustness of SGMC in the presence of noisy graphs. We perturbed the edges of the graphs by adding random Gaussian noise with zero mean and tunable standard deviation to the adjacency matrix. We discarded the edges that became negative as a result of the noise, and symmetrized the adjacency matrix. SGMC1/SGMC2/SGMC3 stand for SGMC with 1 layer (training only $\boldsymbol{C}$), 2 layers (training $\boldsymbol{C}, \boldsymbol{P}$) and 3 layers ($\boldsymbol{C}, \boldsymbol{P}, \boldsymbol{Q}$). **Left**: With clean graphs all SGMC methods perform well. As the noise increases, the regularization induced by the depth kicks in and there is a clear advantage for SGMC3. For large noise, SGMC3 and DMF achieve practically the same performance. **Middle & Right**: eigenvalues of $\boldsymbol{L}_{\text{r}}, \boldsymbol{L}_{\text{c}}$ for different noise levels. Even for moderately large amounts of noise, the structure of the lower part of the spectrum is preserved, and the effect on the low-frequency (smooth) signal remains small.

Performance evaluation. To evaluate the performance of the algorithms in this section, we report the *root mean squared error*,

$$\text{RMSE}(\boldsymbol{X}, \boldsymbol{S}) = \sqrt{\frac{\|(\boldsymbol{X} - \boldsymbol{M}) \odot \boldsymbol{S}\|_F^2}{\sum_{i,j} \boldsymbol{S}_{i,j}}} \tag{27}$$

computed on the complement of the training set. Here $\boldsymbol{X}$ is the recovered matrix and $\boldsymbol{S}$ is the binary mask representing the support of the set on which the RMSE is computed.

We explore the following aspects:

Sampling density. We investigate the effect of the number of samples on the reconstruction error and the *effective rank* of the recovered matrix (Roy and Vetterli, 2007). We demonstrate that in the data-poor regime, the implicit regularization of DMF is too strong resulting in poor recovery, compared to a superior performance achieved by incorporating geometric regularization through SGMC. These experiments are summarized in Figure 2.

Initialization. In all of our experiments we initialize with balanced initialization (6), with scaled identity matrices $10^{-\alpha}\boldsymbol{I}$. We explore the effect of initialization in Figure 12 (in Appendix A).

Rank of the underlying matrix. We explore the effect of the rank of the underlying matrix, showing that as the rank increases it becomes harder for both SGMC and DMF to recover the matrix. A remarkable property of SGMC is that it is able to get a decent approximation of the effective rank of the matrix even with extremely low number of samples. These experiments are summarized in Figure 2.

Noisy graphs. We study the effect of noisy graphs on the preformance of SGMC. Figure 3 demonstrates that SGMC is able to utilize graphs with substantial amounts of noise before its performance drops to the level of vanilla DMF (which does not rely on any knowledge of the row/column graphs).

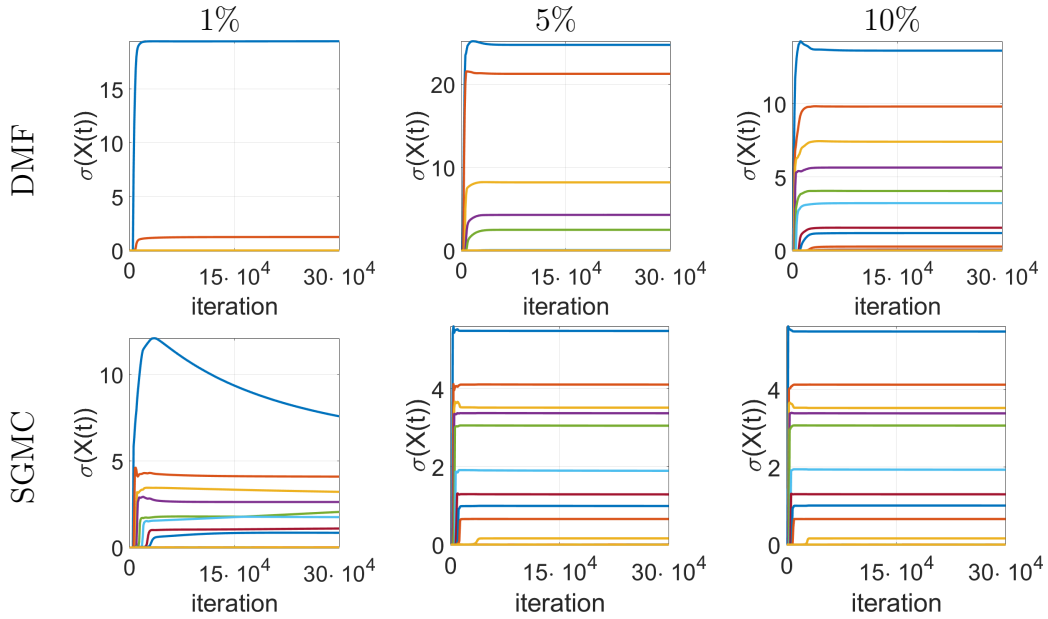


Figure 4: In these experiments, we plot the dynamics of the singular values of the product matrix $\boldsymbol{X}(t)$ during the gradient descent iterations. We show singular value convergence at different sampling densities (left-to-right: 1%, 5%, and 10%) for SGMC and DMF. We use the synthetic Netflix graphs on which we generate a random rank 10 matrix of size 150×200. In accordance with Figure 2, we see that SGMC is able to recover the rank even for a very data-poor regime, whereas DMF demands significantly higher sample complexity.

Dynamics. Figure 4 shows the dynamics of the singular values of $\boldsymbol{X}$ during the optimization. We visually verify that they behave according to (8) and that the convergence rate of the relevant singular values is much faster in SGMC than in DMF.

Code. An interactive jupyter notebook is available here.

5. Results on recommender systems datasets

We demonstrate the effectiveness of our approach on the following datasets: Synthetic Netflix, Flixster, Douban, Movielens (ML-100K) and Movielens-1M (ML-1M) as referenced in Table 1. The datasets include user ratings for items (such as movies) and additional features. For all the datasets we use the users and items graphs taken from Monti et al. (2017). The ML-1M dataset was taken from Berg et al. (2017), for which we constructed 10 nearest neighbor graphs for users/items from the features, and used a Gaussian kernel with $\sigma = 1$ for edge weights. See Table 4 in Appendix A for a summary of the dataset statistics. For all the datasets, we report the results for the same test splits as that of Monti et al. (2017) and Berg et al. (2017). The compared methods are referenced in Table 1.

Proposed baselines. We report the results obtained using the methods discussed above, with the addition of the following method:

- **SGMC-Z**: a variant of SGMC that uses (22) as a data term. For this method we chose a maximal value of $p_{\max}, q_{\max}$ (which can be larger than m, n) and a skip determining the spectral resolution, denoted by $p_{\text{skip}}, q_{\text{skip}}$. We use $p = 1 + kp_{\text{skip}}, q = 1 + kq_{\text{skip}},\ k \in \mathbb{N}$.

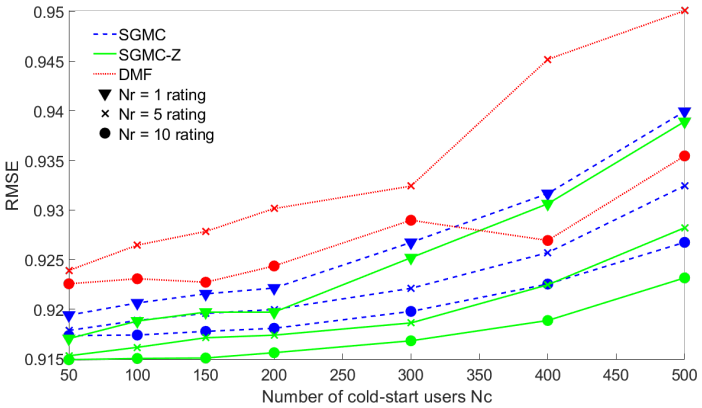
In addition, we add the diagonalization terms (24), (25) weighted by ρ_r, ρ_c, respectively, to the SGMC/SGMC-Z methods. The optimization is carried out using gradient descent with fixed step size (i.e., fixed learning rate), which is provided for each experiment alongside all the other hyper-parameters in Table 5.

Initialization. All our methods are deterministic and did not require multiple runs to account for initialization. We always initialize the matrices $\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$ with $10^{-\alpha}\boldsymbol{I}$. In Figure 12 we reported results on synthetic Netflix and ML-100K datasets for different values of α. We noticed that for SGMC and SGMC-Z it is best to use $10^{-\alpha} = 1$. According to (Gunasekar et al., 2017; Li et al., 2017), DMF requires a large α to decrease the generalization error. We used DMF with $10^{-\alpha} = 0.01$ for Synthetic Netflix and $10^{-\alpha} = 1$ for the real world datasets, in accordance with Figure 12 and our experimentation. In the cases where only one of the bases was available, such as in Douban and Flixster-user only benchmarks, we set the basis corresponding to the absent graph to identity.

Stopping condition. Our stopping condition for the gradient descent iterations is based on a validation set. We use 95% of the available entries for training (i.e., to construct the mask $\boldsymbol{S}$) and the rest 5% for validation. The 95/5 split was chosen at random. We stop the iterations when the RMSE (27), evaluated on the validation set, does not change by more than $\texttt{tol} = 0.000001$ between two consecutive iterations, $|\text{RMSE}_k - \text{RMSE}_{k-1}| < \texttt{tol}$. Since we did not apply any optimization into the choice of the validation set, we also report the best RMSE achieved on the test set via early stopping. In this regard, the number of iterations is yet another hyper parameter that has to be tuned for best performance.

5.1. Cold start analysis

A particularly interesting scenario in the context of recommender systems is the presence of *cold-start* users, referring to the users who have not rated enough movies yet. We perform an analysis of the performance of our method in the presence of such cold start users on the ML-100K dataset. In order to generate a dataset consisting of N_c cold start users, we sort the users according to the number of ratings provided by each user, and retain at most N_r ratings (chosen randomly) of the bottom N_c users (i.e., the users who provided the



Comparison of test RMSE in the presence of cold start users on the ML-100K dataset. The x-axis corresponds to the number of the cold start users $N_c = 50, 100, \ldots 500$. Red, blue and green correspond to DMF, SGMC and SGMC-Z methods respectively as also shown in the legend. Different shapes of the markers indicate different number of maximum ratings ($N_r = \{1, 5, 10\}$) available per cold-start user.

Model	Synthetic Neflix	Flixster	Douban	ML-100K
MC (Candès and Recht, 2009)	–	1.533	0.845	0.973
GMC (Kalofolias et al., 2014)	0.3693	–	–	0.996
GRALS (Rao et al., 2015)	0.0114	1.313/1.245	0.833	0.945
RGCNN (Monti et al., 2017)	0.0053[a]	1.179/0.926	0.801	0.929
GC-MC (Berg et al., 2017)	–	**0.941**/0.917	0.734	0.910[b]
FM (ours)	0.0064	3.32	3.15	1.10
DMF (Arora et al., 2019), (ours)	0.0468[d]	1.06	0.732	0.918[c]/ 0.922
SGMC (ours)	**0.0021**	0.971 / 0.900	**0.731**	0.912
SGMC-Z (ours)	0.0036	0.957 / **0.888**	0.733	**0.907**[c]/ 0.913

[a] This number corresponds to the inseparable version of MGCNN.
[b] This number corresponds to GC-MC.
[c] Early stopping.
[d] Initialization with $0.01\mathbf{I}$.

Table 1: RMSE test set scores for runs on Synthetic Netflix (Monti et al., 2017), Flixster (Jamali and Ester, 2010), Douban (Ma et al., 2011), and Movielens-100K (Harper and Konstan, 2016). For Flixster, we show results for both user/item graphs (right number) and user graph only (left number). Baseline numbers are taken from (Monti et al., 2017; Berg et al., 2017).

least ratings). We choose the values $N_c = \{50, 100, 150, 200, 300, 400, 500\}$ and $N_r = \{1, 5, 10\}$, and run our algorithms: DMF, SGMC and SGMC-Z, with the same hyperparameter settings used for obtaining Table 1. We use the official ML-100K test set for evaluation. Similar to before, we use 5% of the training samples as a validation set used for determining the stopping condition.

The results presented in the inline figure suggest that the SGMC and SGMC-Z outperform DMF significantly, indicating the importance of the geometry as data becomes scarcer. As expected, we can see that the performance drops as the number of ratings per user decreases. Furthermore, we can observe that SGMC-Z consistently outperforms SGMC by a small margin. We note that SGMC-Z, even in the presence of $N_c = 500$ cold start users with $N_r = 5$ ratings, is still able to outperform the full data performance of Monti et al. (2017), demonstrating the strength of geometry and implicit low-rank induced by SGMC-Z.

Scalability. All the experiments presented in the paper were conducted on a machine consisting of 64GB CPU memory, on an NVIDIA GTX 2080Ti GPU. Most of our large-scale experiments take upto 10-30 minutes of time until convergence, therefore, are rather quick. In this work we focused on the conceptual idea of solving matrix completion via the framework of deep matrix factorization by incorporating geometric regularization, paying little attention to the issue of scalability. There are two main computational bottlenecks to our approach: The spatial version (12) requires the computation of the matrix product $\boldsymbol{AZB}^\top$ in each gradient iteration, and the spectral version requires also the eigenvalue decomposition of $\boldsymbol{L}_\mathrm{r}, \boldsymbol{L}_\mathrm{c}$. These limitations apply to other graph neural networks as well (Hu et al., 2020), and we believe that they can be at least partially addressed by ad-hoc solutions.

5.2. Discussion

A few remarkable observations can be extracted from Table 1: First, on the Douban and ML-100K dataesets, vanilla DMF shows competitive performance with all the other methods.

This suggests that the geometric information is not very useful for these datasets. Second, the proposed SGMC algorithms outperform the other methods, despite their simple and fully linear architecture. This suggests that the other geometric methods do not exploit the geometry properly, and this fact is obscured by their cumbersome architecture. Third, while some of the experiments reported in Table 1 showed only slight margins in favor of SGMC/SGMC-Z compared to DMF, the results in the Synthetic Netflix column, the ones reported on Synthetic Movielens-100K (Table 3 in Appendix A) and the ones reported in Figure 2, suggest that when the geometric model is accurate our methods demonstrate superior results. Table 2 in Appendix A presents the results of Movielens-1M. First, we can deduce that vanilla DMF model is able to match the performance of complex alternatives. Furthermore, using graphs produces slight improvements over the DMF baseline and overall provides competitive performance compared to heavily engineered methods. On Synthetic Netflix, we notice that by using SGMC, we outperform Monti et al. (2017) by a significant margin, reducing the test RMSE by half. Additionally, it can be observed that DMF performs poorly on both synthetic datasets compared to SGMC/SGMC-Z, raising a question as to the quality of the graphs provided with those datasets on which DMF performed comparably.

A compelling argument for this behaviour is given by Table 4 in Appendix A. We can see that in the real datasets we tested on, the number of available samples is way below the density required by DMF to achieve good performance, in accordance with our findings in Section 4. With high quality graphs, we should have expected SGMC to outperform DMF by a large margin.

6. Results on drug-target interaction

In this section we demonstrate the effectiveness of our approach on the problem of predicting drug-target interaction (DTI). The task is to find effective interactions between chemical compounds (drugs) and amino-acid sequences/proteins (targets). This is traditionally done through wet-lab experiments which are costly and laborious, and lead to high attrition rate. One possible way to improve this procedure is to predict interaction probabilities through a computational model. To that end, DTI can be interpreted as a matrix completion problem where the rows correspond to different drugs and the columns correspond to different targets. Each entry in the matrix corresponds to the probability of interaction between a drug and target. We assume that we are given two graphs encoding similarities between drugs and similarities between targets. The similarity between two drugs is measured by the number of shared substructures within their chemical structures. The similarity between targets is given by their genomic sequence similarity. These similarity measures constitute a standard similarity score that is common in the DTI prediction task. For more information on the problem and the construction of the graphs, we refer to Mongia and Majumdar (2020) and references therein.

Datasets. We use three benchmark datasets introduced in Yamanishi et al. (2008), having three different classes of proteins: enzymes (Es), ion channels (ICs), and G protein-coupled receptors (GPCRs). The data was simulated from public databases KEGG BRITE (Kanehisa et al., 2006), BRENDA (Schomburg et al., 2004) SuperTarget (Günther et al., 2007) and

DrugBank (Wishart et al., 2008), and is publicly available[3]. The data from each of these databases is formatted as an adjacency matrix between drugs and targets encoding the interaction as 1 if drug-target pair are known to interact and 0 otherwise.

Baselines. We validated our proposed method by comparing it with three recent methods proposed in the literature: MGRNNM (Mongia and Majumdar, 2020), GRMF (Ezzat et al., 2016), CMF (Zheng et al., 2013). For all the baselines we ran the publicly available code[4] on the aforementioned datasets using the same graphs and same train-test splits.

Evaluation protocol. Similarly to (Mongia and Majumdar, 2020), we have performed 5 runs (with different random seeds) of 10-fold cross-validation for each of the algorithms under three cross-validation settings (CVS):

- CVS1/Pair prediction: random drug–target pairs are chosen randomly for the test set. It is the conventional setting for validation and evaluation.
- CVS2/Drug prediction: complete drug profiles are left out of the training set, i.e., some rows are absent. This tests the algorithm's ability to predict interactions for novel drugs for which no interaction information is available.
- CVS3/Target prediction: complete target profiles are left out of the training set. i.e., some columns are absent. It tests the algorithm's ability to predict interactions for novel targets.

Out of the 10 folds one was left out for testing whereas the remaining 9 folds were used as the training set. To evaluate performence we measure area under ROC curve (AUC), area under the precision-recall curve (AUPR), and RMSE. In biological drug discovery, AUPR is of more significance since it penalizes high ranked false positive interactions much more than AUC. Those pairs would be biologically validated later in the drug discovery process. The results are summarised in Table 6 in Appendix B.

Discussion. Table 6 clearly shows that SGMC mostly outperforms the other methods in all metrics. This is without applying any particular task specific optimization of the loss function and other hyper-parameters. In particular, the RMSE criterion, which is the one optimized by all the methods, is significantly lower for SGMC compared to other matrix factorization algorithms. This serves as a further reinforcement of the strength of the implicit regularization in SGMC compared to the nuclear norm (Mongia and Majumdar, 2020) and explicit low rank matrix factorization methods (Ezzat et al., 2016).

7. Related work

Geometric matrix completion. There is a vast literature on classical approaches for matrix completion, and covering it is beyond the scope of this paper. In recent years, the advent of deep learning platforms equipped with efficient automatic differentiation tools allows the exploration of sophisticated models that incorporate intricate regularizations. Some of these contemporary approaches to matrix completion fall under the umbrella term

3. http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/
4. https://github.com/aanchalMongia/MGRNNMforDTI

of *geometric deep learning*, which generalizes standard (Euclidean) deep learning to domains such as graphs and manifolds. For example, *graph convolutional neural networks* (GCNNs) follow the architecture of standard CNNs, but replace the Euclidean convolution operator with linear filters constructed using the graph Laplacian. We distinguish between *inductive* approaches to matrix completion, which work directly on the users and items features to predict the rating matrix (e.g., Berg et al. (2017)), and *transductive* approaches, which make use of side information to construct graphs encoding relations between rows/columns (Kovnatsky et al., 2014; Kalofolias et al., 2014; Monti et al., 2017).

More recently, it has been demonstrated that some graph CNN architectures can be greatly simplified, and still perform competitively on several graph analysis tasks (Wu et al., 2019). Such simple techniques have the advantage of being easier to analyze and reproduce. One of the simplest notable approaches is *deep linear networks*, networks comprising of only linear layers. While these network are still mostly used for theoretical investigations, we note the recent results of Bell-Kligler et al. (2019) who successfully employed such a network for the tasks of blind image deblurring, and (Richardson and Weiss, 2020) who used it for image-to-image translation. Jing et al. (2020) showed that overparametrized linear layers can be used for implicit rank minimization within a generative model.

A closely related field dealing with reconstruction of signals defined on graphs is graph signal processing. In this field the problem is attacked by extending results from harmonic analysis to problems defined on graphs. For example, Puy and Pérez (2018); Puy et al. (2018) have developed a random sampling strategy that provides reconstruction guarantees for bandlimited signals on graphs. The reconstruction is performed via minimizing an l_2 data term with Dirichlet regularization (3). Random sampling schemes and reconstruction guarantees for bandlimited signals on product graphs were developed in Ortiz-Jiménez et al. (2018); Varma and Kovacevic (2018). While these results are extremely useful in designing sampling strategies for bandlimited signals on graphs, they are of less use when we are given the samples upfront and have no ability to control the sampling process. Nevertheless, their analysis sheds light on the success of spectral regularization in reconstruction problems on graphs and we intend to integrate these ideas with our approach in the future.

Product manifold filter & Zoomout. The inspiration for our paper stems from techniques for finding shape correspondence. In particular, the functional maps framework and its variants (Ovsjanikov et al., 2012, 2016). Most notably the work of Litany et al. (2017) who combined functional maps with joint diagonalization to solve partial shape matching problems, and the *product manifold filter* (PMF) (Vestner et al., 2017a,b) and *zoomout* (Melzi et al., 2019) – two greedy algorithms for correspondence refinement by gradual introduction of high frequencies.

8. Conclusion

In this work we have proposed a simple spectral technique for matrix completion, building upon recent practical and theoretical results in geometry processing and deep linear networks. We have shown, through extensive experimentation on real and synthetic datasets across domains, that combining the implicit regularization of DMF with explicit, and possibly noisy, geometric priors can be extremely useful in data-poor regimes. Our work is a step towards building interpretable models that are grounded in theory, and proves that such

simple models need not only be considered for theoretical study. Through a proper lens, they can be made useful.

```
Y = M * Omega
A = Parameter(ones(m, k), requires_grad=True)
Z = Parameter(ones(k, k), requires_grad=True)
B = Parameter(ones(n, k), requires_grad=True)

mu_row = mu_col = 0.4
opt = SGD([A, Z, B], lr=0.0001)
total_iters = 10**4
for iter in range(total_iters):
  opt.zero_grad()
  X = A @ Z @ B.T
  E_data     = norm(Omega * X  - Y)**2
  E_dir_row  = L_row @ X @ L_row.T
  E_dir_col  = L_col.T @ X @ L_col
  E_total = E_data + mu_row*E_dir_row + mu_col*E_dir_col
  E_total.backward()
  opt.step()
```

Figure 6: A few lines of code.

Acknowledgments

We thank Angshul Majumdar for useful discussions, and for providing datasets and code for the matrix completion methods used in drug-target interaction prediction. This research was supported by ERC StG RAPID and ERC CoG EARS.

References

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *arXiv preprint arXiv:1905.13655*, 2019.

Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems 32*, pages 284–293. 2019.

Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

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

Gintare Karolina Dziugaite and Daniel M. Roy. Neural network matrix factorization. *CoRR*, abs/1511.06443, 2015. URL http://arxiv.org/abs/1511.06443.

Ali Ezzat, Peilin Zhao, Min Wu, Xiao-Li Li, and Chee-Keong Kwoh. Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM transactions on computational biology and bioinformatics*, 14(3):646–656, 2016.

Akshay Gadde, Sunil K Narang, and Antonio Ortega. Bilateral filter: Graph spectral interpretation and extensions. In *2013 IEEE International Conference on Image Processing*, pages 1222–1226. IEEE, 2013.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6151–6159, 2017.

Stefan Günther, Michael Kuhn, Mathias Dunkel, Monica Campillos, Christian Senger, Evangelia Petsalaki, Jessica Ahmed, Eduardo Garcia Urdiales, Andreas Gewiess, Lars Juhl Jensen, et al. Supertarget and matador: resources for exploring drug-target relationships. *Nucleic acids research*, 36(suppl_1):D919–D922, 2007.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19, 2016. URL https://grouplens.org/datasets/movielens/.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

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*, pages 135–142. ACM, 2010.

Li Jing, Jure Zbontar, et al. Implicit rank-minimizing autoencoder. *Advances in Neural Information Processing Systems*, 33, 2020.

Vassilis Kalofolias, Xavier Bresson, Michael Bronstein, and Pierre Vandergheynst. Matrix completion on graphs. *arXiv preprint arXiv:1408.1717*, 2014.

Minoru Kanehisa, Susumu Goto, Masahiro Hattori, Kiyoko F Aoki-Kinoshita, Masumi Itoh, Shuichi Kawashima, Toshiaki Katayama, Michihiro Araki, and Mika Hirakawa. From genomics to chemical genomics: new developments in kegg. *Nucleic acids research*, 34 (suppl_1):D354–D357, 2006.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009. ISSN 0018-9162.

Artiom Kovnatsky, Michael M. Bronstein, Xavier Bresson, and Pierre Vandergheynst. Functional correspondence by matrix completion, 2014.

Joonseok Lee, Seungyeon Kim, Guy Lebanon, Yoram Singer, and Samy Bengio. Llorma: Local low-rank matrix approximation. *Journal of Machine Learning Research*, 17(15):1–24, 2016.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. *arXiv preprint arXiv:1712.09203*, 2017.

Or Litany, Emanuele Rodolà, Alexander M Bronstein, and Michael M Bronstein. Fully spectral partial shape matching. In *Computer Graphics Forum*, volume 36, pages 247–258. Wiley Online Library, 2017.

Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

Simone Melzi, Jing Ren, Emanuele Rodola, Maks Ovsjanikov, and Peter Wonka. Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865*, 2019.

Aanchal Mongia and Angshul Majumdar. Drug-target interaction prediction using multi graph regularized nuclear norm minimization. *Plos one*, 15(1):e0226484, 2020.

Federico Monti, Michael Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *Advances in Neural Information Processing Systems*, pages 3697–3707, 2017.

Guillermo Ortiz-Jiménez, Mario Coutino, Sundeep Prabhakar Chepuri, and Geert Leus. Sampling and reconstruction of signals on product graphs. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 713–717. IEEE, 2018.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012.

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, page 9. ACM, 2016.

Gilles Puy and Patrick Pérez. Structured sampling and fast reconstruction of smooth graph signals. *Information and Inference: A Journal of the IMA*, 7(4):657–688, 2018.

Gilles Puy, Nicolas Tremblay, Rémi Gribonval, and Pierre Vandergheynst. Random sampling of bandlimited signals on graphs. *Applied and Computational Harmonic Analysis*, 44(2): 446–475, 2018.

Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. In *Advances in neural information processing systems*, pages 2107–2115, 2015.

Antoine Recanati. *Relaxations of the Seriation problem and applications to de novo genome assembly*. PhD thesis, 2018.

Eitan Richardson and Yair Weiss. The surprising effectiveness of linear unsupervised image-to-image translation. *arXiv preprint arXiv:2007.12568*, 2020.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pages 606–610. IEEE, 2007.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1257–1264, USA, 2007. Curran Associates Inc. ISBN 978-1-60560-352-0.

Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 791–798, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3.

Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl_1):D431–D433, 2004.

Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 111–112, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3473-0.

Amit Singer, Yoel Shkolnisky, and Boaz Nadler. Diffusion interpretation of nonlocal neighborhood filters for signal denoising. *SIAM Journal on Imaging Sciences*, 2(1):118–139, 2009.

Daniel Spielman. Spectral graph theory. *Lecture Notes, Yale University*, pages 740–0776, 2009.

Rohan A Varma and Jelena Kovacevic. Sampling theory for graph signals on product graphs. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 768–772. IEEE, 2018.

Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodola, Alex Bronstein, Michael Bronstein, Ron Kimmel, et al. Efficient deformable shape correspondence via kernel matching. In *2017 International Conference on 3D Vision (3DV)*, pages 517–526. IEEE, 2017a.

Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3327–3336, 2017b.

David S Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl_1):D901–D906, 2008.

Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019.

Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24(13):i232–i240, 2008.

Xiaodong Zheng, Hao Ding, Hiroshi Mamitsuka, and Shanfeng Zhu. 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*, pages 1025–1033, 2013.

Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. A neural autoregressive approach to collaborative filtering. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 764–773, New York, New York, USA, 20–22 Jun 2016. PMLR.

Appendix A. Recommendation systems

Ablation study. We study the effects of different hyper-parameters of the algorithms on the final reconstruction of the matrix. We perform an ablation study on the effects of $\rho, \mu, p_{\max}, q_{\max}$ on DMF, SGMC and SGMC-Z. The results are summarized in Figures 7, 8, 9. It is interesting to note that in the case of DMF and SGMC, overparametrizing $\boldsymbol{C}, \boldsymbol{Q}, \boldsymbol{P}$ consistently improves the performance (see Figure 9), but it only holds up to a certain point, beyond which the overparametrization does not seem to effect the reconstruction error. Notice that in the Table 5, μ_r, μ_c control the Dirichlet energy of rows and columns; while ρ_r, ρ_c govern the weights of row/column diagonalization energy.

Synthetic MovieLens-100K. While the experiments reported in Table 1 showed slight margins in favor of methods using geometry, we further experimented with a synthetic model generated from the ML-100K dataset. The purpose of this experiment is to investigate whether the results are due to the DMF model or due to the geometry as incorporated by SGMC/SGMC-Z. The synthetic model was generated by projecting $\boldsymbol{M}$ on the first 50 eigenvectors of $\boldsymbol{L}_{\mathrm{r}}, \boldsymbol{L}_{\mathrm{c}}$, and then matching the ratings histogram with that of the original ML-100K dataset. This nonlinear operation increased the rank of the matrix from 50 to about 400. See Figure 10 in the Appendix A for a visualization of the full matrix, singular

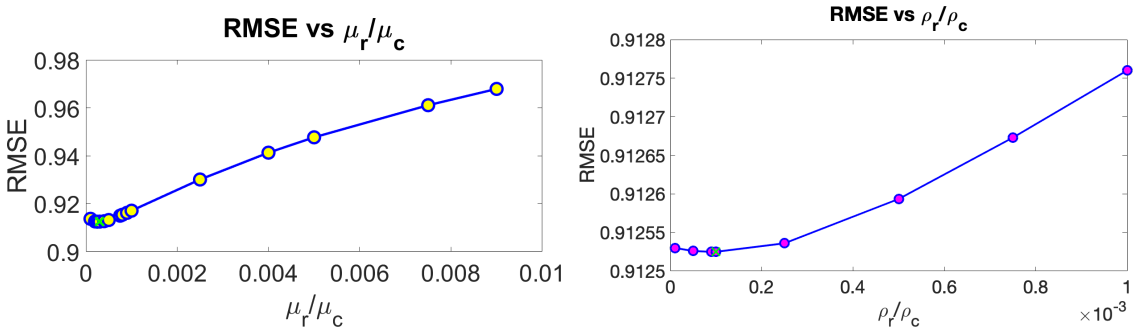
value distribution and the users/items graphs. The test set and training set were generated randomly and are the same size as those of the original dataset. The results reported in Table 3 and those on the Synthetic Netflix column in Table 1 clearly indicate that SGMC/SGMC-Z outperforms DMF, suggesting that when the geometric model is accurate it is possible to use it to improve the results.

Model	**ML-1M**
PMF (Salakhutdinov and Mnih, 2007)	0.883
I-RBM (Salakhutdinov et al., 2007)	0.854
BiasMF (Koren et al., 2009)	0.845
NNMF (Dziugaite and Roy, 2015)	0.843
LLORMA-Local (Lee et al., 2016)	0.833
I-AUTOREC (Sedhain et al., 2015)	0.831
CF-NADE (Zheng et al., 2016)	**0.829**
GC-MC (Berg et al., 2017)	0.832
DMF (Arora et al., 2019), (ours)	0.843
SGMC (ours)	0.839

Table 2: Comparison of test RMSE scores on Movielens-1M dataset. Baseline scores are taken from (Zheng et al., 2016; Berg et al., 2017)

Model	**Synthetic ML-100K**
DMF	0.9147
SGMC	0.5006
SGMC-Z	**0.4777**

Table 3: Comparison of average RMSE of DMF, SGMC and SGMC-Z baselines calculated on 5 randomly generated Synthetic Movielens-100K datasets.



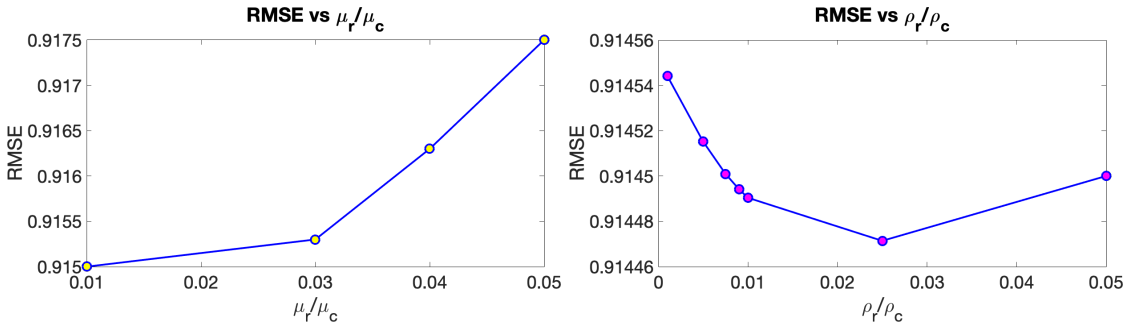
(a) (b)

Figure 7: Ablating $\rho_r = \rho_c$ and $\mu_r = \mu_c$ of SGMC on the ML-100K dataset. The rest of the parameters were set to the ones reported in Table 5. Green X denotes the baseline from Table 1.

Dataset	Users	Items	Features	Ratings	Density	Rating levels
Flixster	3,000	3,000	Users/Items	26,173	0.0029	$0.5, 1, \ldots, 5$
Douban	3,000	3,000	Users	136,891	0.0152	$1, 2, \ldots, 5$
MovieLens-100K	943	1,682	Users/Items	100,000	0.0630	$1, 2, \ldots, 5$
MovieLens-1M	6,040	3,706	Users/Items	1,000,209	0.0447	$1, 2, \ldots, 5$
Synthetic Netflix	150	200	Users/Items	4500	0.15	$1 \ldots 5$ [a]
Synthetic ML-100K	943	1,682	Users/Items	100,000	0.0630	$1, 2, \ldots, 5$

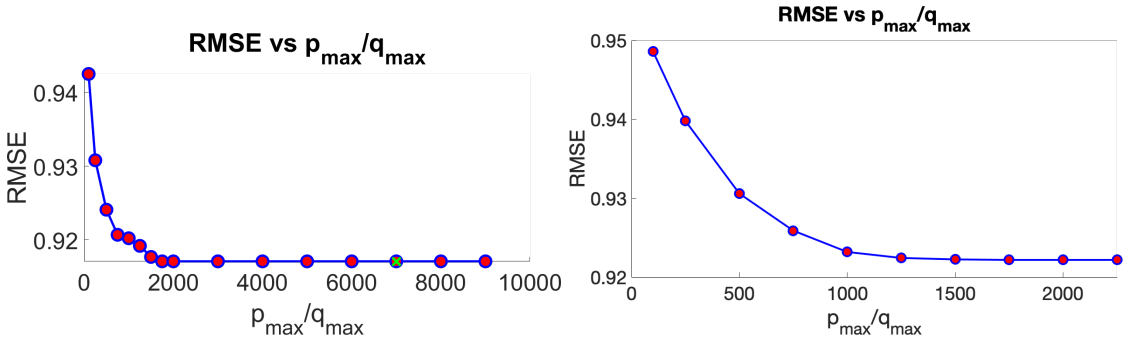
Table 4: Number of users, items and ratings for Flixster, Douban, Movielens-100K, Movielens-1M, Synthetic Netflix and Synthetic Movielens-100K datasets used in our experiments and their respective rating density and rating levels.

[a] The ratings are not integer-valued.



(a) (b)

Figure 8: Ablating ρ_r, ρ_c and μ_r, μ_c of SGMC-Z on the ML-100K dataset. The rest of the parameters were set to the ones reported in Table 5.



(a) (b)

Figure 9: Effect of overparametrization: SGMC (left) and DMF (right). x-axis indicates the values of $p_{\max}, q_{\max}$, and y-axis presents the RMSE. Green X denotes the baseline from Table 1.

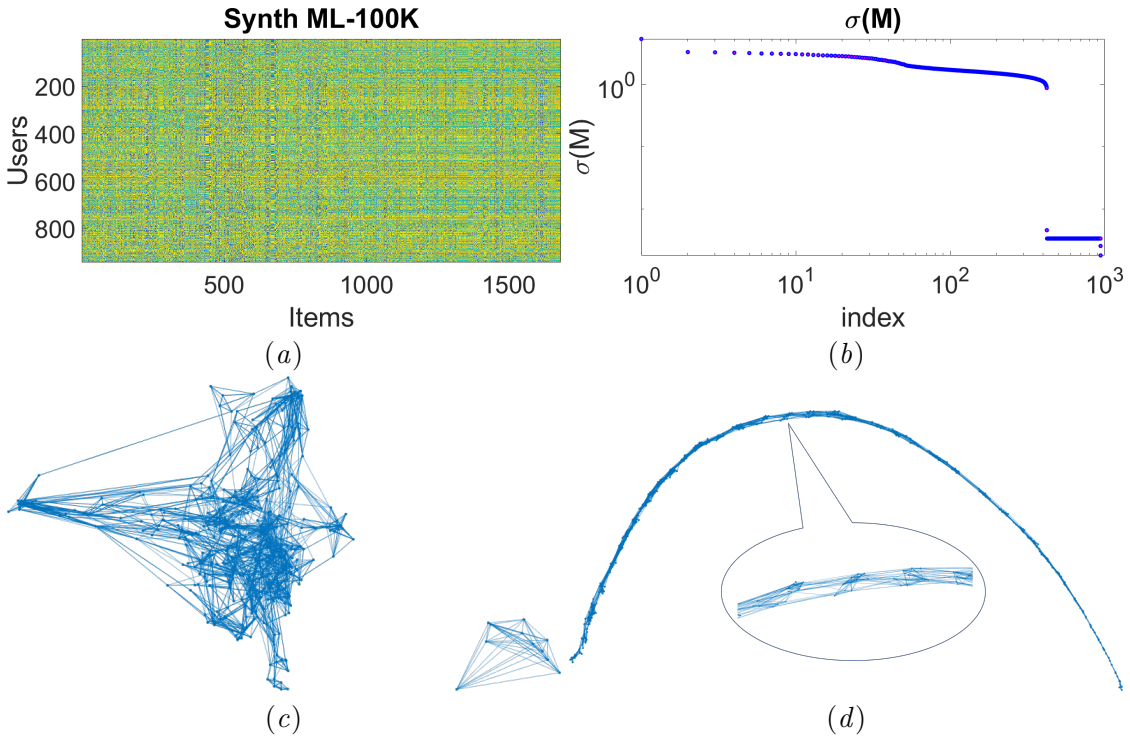


Figure 10: Synthetic Movielens-100k. Top-left: Full matrix. Top-right: singular values of the full matrix. Bottom left & right: items & users graph. Both graphs are constructed using 10 nearest neighbors.

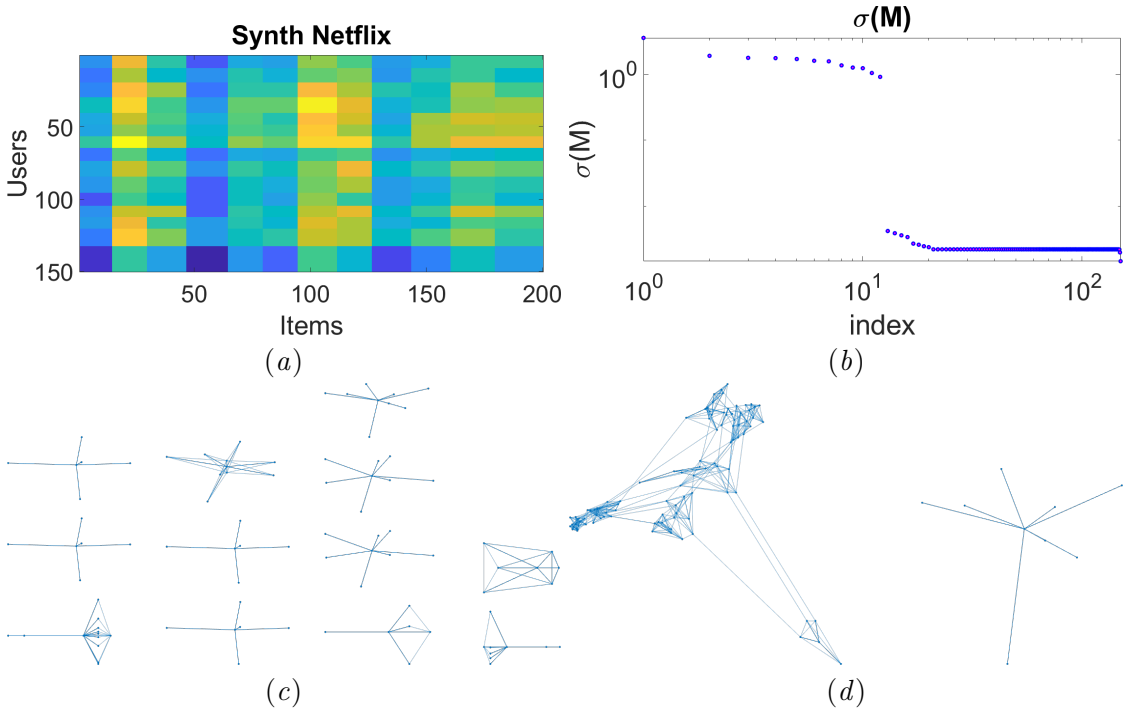


Figure 11: Synthetic Netflix. Top-left: Full matrix. Top-right: singular values of the full matrix. Bottom left & right: items & users graph. Taken from (Monti et al., 2017).

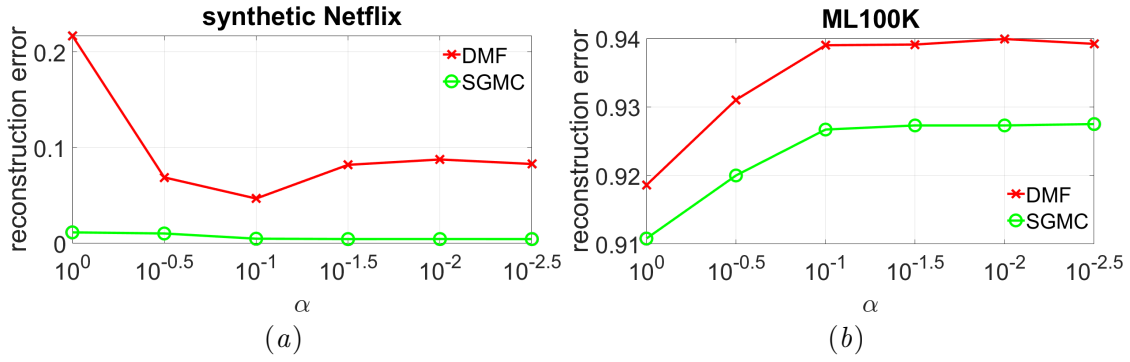


Figure 12: Reconstruction error (on the test set) vs. scale of initialization. For each method we initialized $\boldsymbol{P}, \boldsymbol{Q}$ with $\alpha \boldsymbol{I}$. SGMC consistently outperforms DMF for any initialization.

Dataset	Method	$p_{\max}/q_{\max}$	$p_{\text{skip}}/q_{\text{skip}}$	μ_r/μ_c	ρ_r/ρ_c	Trainable variables	learning rate
Synthetic Netflix	DMF	200/200	−/−	−/−	−/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	5×10^{-5}
	FM	200/200	1/1	0.4/0.4	−/−	$\boldsymbol{C}$	5×10^{-4}
	SGMC	20/20	−/−	0.001/0.001	0.1/−	$\boldsymbol{P}, \boldsymbol{C}$	5×10^{-3}
	SGMC-Z	500/500	3/1	0.4/0.4	0.1/0.1	$\boldsymbol{P}, \boldsymbol{C}$	5×10^{-5}
Flixster	DMF	3000/3000	−/−	−/−	−/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	1×10^{-4}
	SGMC	3000/3000	−/−	0.0001/0.0001	0.0001/0.0001	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	1×10^{-4}
	SGMC-Z	200/200	2/2	0.0025/0.0025	−/−	$\boldsymbol{P}, \boldsymbol{C}$	5×10^{-6}
Flixster (users only)	SGMC	3000/3000	−/−	0.0001/−	0.0001/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	5×10^{-5}
	SGMC-Z	200/200	20/20	0.0025/−	0.001/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	5×10^{-7}
Douban	DMF	3000/3000	−/−	−/−	−/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	6×10^{-6}
	SGMC	2500/2500	−/−	0.001/−	0.001/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	2×10^{-6}
	SGMC-Z	1000/1000	50/1000	0.011/0	0.004/0	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	2×10^{-6}
ML-100K	DMF	2000/2000	−/−	−/−	−/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	5×10^{-5}
	SGMC	4000/4000	−/−	0.0003/0.0003	0.0001/0.0001	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	5×10^{-5}
	SGMC-Z	3200/3200	30/35	0.03/0.03	0.2/0.2	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	3×10^{-7}
ML-1M	DMF	7000/7000	−/−	−/−	−/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	1×10^{-5}
	SGMC	7000/7000	−/−	0.0001/0.0001	−/−	$\boldsymbol{P}, \boldsymbol{C}$	8×10^{-5}
Synthetic ML-100K	DMF	8000/8000	−/−	−/−	−/−	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	9×10^{-5}
	SGMC	600/600	−/−	0.001/0.001	0.009/0.009	$\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{Q}$	2×10^{-5}
	SGMC-Z	500/500	3/1	0.001/0.001	0.009/0.009	$\boldsymbol{P}, \boldsymbol{C}$	5×10^{-6}

Table 5: Hyper-parameter settings for the algorithms: DMF, SGMC and SGMC-Z, reported in Tables 1, 2, 3.

Appendix B. Drug-target interaction

Es					
		MGRNNM	GRMF	CMF	SGMC
CVS1	AUC	0.9940 ± 0.0019	0.9900 ± 0.0017	0.8443 ± 0.0178	0.9967 ± 0.0016
	AUPR	0.9559 ± 0.0059	0.9295 ± 0.0081	0.6733 ± 0.0238	0.9729 ± 0.0033
	RMSE	0.0441 ± 0.0008	0.0476 ± 0.0008	0.2045 ± 0.0079	0.0432 ± 0.0007
CVS2	AUC	0.9333 ± 0.0229	0.9582 ± 0.0138	0.9183 ± 0.1068	0.9656 ± 0.0168
	AUPR	0.8350 ± 0.0364	0.8553 ± 0.0284	0.3406 ± 0.0726	0.8565 ± 0.0285
	RMSE	0.0776 ± 0.0065	0.0827 ± 0.0066	0.5427 ± 0.0628	0.0541 ± 0.0049
CVS3	AUC	0.9709 ± 0.0109	0.9674 ± 0.0159	0.8525 ± 0.0188	0.9760 ± 0.1009
	AUPR	0.0909 ± 0.0280	0.9011 ± 0.0292	0.1958 ± 0.0636	0.9134 ± 0.0373
	RMSE	0.0959 ± 0.0039	0.0925 ± 0.0036	0.1842 ± 0.0370	0.0512 ± 0.0037
GPCRs					
		MGRNNM	GRMF	CMF	SGMC
CVS1	AUC	0.9770 ± 0.0068	0.9765 ± 0.0061	0.9129 ± 0.0114	0.9831 ± 0.0065
	AUPR	0.7995 ± 0.023	0.8000 ± 0.0028	0.7306 ± 0.0164	0.8691 ± 0.02
	RMSE	0.1136 ± 0.0027	0.1139 ± 0.0026	0.7306 ± 0.0164	0.0954 ± 0.0026
CV-B	AUC	0.9664 ± 0.0087	0.9705 ± 0.0091	0.9601 ± 0.0153	0.9750 ± 0.0009
	AUPR	0.8936 ± 0.0187	0.8892 ± 0.0188	0.8754 ± 0.0364	0.8836 ± 0.0199
	RMSE	0.1440 ± 0.0070	0.1476 ± 0.0065	0.1381 ± 0.0151	0.1009 ± 0.0060
CVS3	AUC	0.8762 ± 0.0258	0.9297 ± 0.0170	0.7843 ± 0.0701	0.9299 ± 0.0258
	AUPR	0.6866 ± 0.0658	0.7149 ± 0.0493	0.2256 ± 0.1021	0.7232 ± 0.0566
	RMSE	0.1495 ± 0.0150	0.1499 ± 0.0173	1.5743 ± 0.2302	0.1179 ± 0.0099
ICs					
		MGRNNM	GRMF	CMF	SGMC
CVS1	AUC	0.9947 ± 0.0013	0.9922 ± 0.0015	0.8745 ± 0.0134	0.9964 ± 0.001
	AUPR	0.9584 ± 0.0038	0.9527 ± 0.0043	0.8172 ± 0.0259	0.9784 ± 0.0023
	RMSE	0.0874 ± 0.0047	0.0780 ± 0.0021	0.2487 ± 0.0085	0.0710 ± 0.0015
CVS2	AUC	0.0971 ± 0.0142	0.9689 ± 0.0138	0.9229 ± 0.0184	0.9714 ± 0.0156
	AUPR	0.9026 ± 0.0326	0.9014 ± 0.0314	0.6426 ± 0.0632	0.9044 ± 0.0308
	RMSE	0.1780 ± 0.0118	0.1548 ± 0.0122	0.4632 ± 0.1578	0.0948 ± 0.0117
CVS3	AUC	0.9547 ± 0.0188	0.9703 ± 0.0115	0.7781 ± 0.0344	0.9731 ± 0.0116
	AUPR	0.9030 ± 0.0341	0.9147 ± 0.0304	0.2198 ± 0.0580	0.9196 ± 0.0264
	RMSE	0.0901 ± 0.0084	0.1520 ± 0.0045	0.3598 ± 0.0615	0.0911 ± 0.0068

Table 6: Results obtained on three drug-target interaction datasets: enzymes (Es), ion channels (ICs), G protein-coupled receptors (GPCRs). Each entry presents the mean and standard deviation across 5 runs (with different random seeds) of 10-fold cross validation. Descriptions of the evaluated baselines are reported in the text. Colored in green are the cases where SGMC ranks first, and in red are the cases where SGMC ranks is second or third.