Convex Collective Matrix Factorization

Guillaume Bouchard[1] **Shengbo Guo**[1] **Dawei Yin**[2]
[1]Xerox Research Centre Europe, France [2]Lehigh University, USA

Abstract

In many applications, multiple interlinked sources of data are available and they cannot be represented by a single adjacency matrix, to which large scale factorization method could be applied. *Collective matrix factorization* is a simple yet powerful approach to *jointly* factorize multiple matrices, each of which represents a relation between two entity types. Existing algorithms to estimate parameters of collective matrix factorization models are based on non-convex formulations of the problem; in this paper, a *convex* formulation of this approach is proposed. This enables the derivation of large scale algorithms to estimate the parameters, including an iterative eigenvalue thresholding algorithm. Numerical experiments illustrate the benefits of this new approach.

1 Introduction

Knowledge is often encoded in large graphs representing relational data, such as DBPedia and Freebase; many smaller size dedicated knowledge bases are also represented using relational graphs: more flexible than traditional databases, they represent data without well-defined relational schema. Examples are Wordnet, OpenCyc or Yago [19].

We propose a novel scalable algorithm to curate, improve, predict and recommend links in large traditional relational databases, as well as knowledge graphs. One of the main challenges with such relational data is *link prediction*: the prediction of relations between two entities represented by edges linking two nodes in the knowledge graph. A standard method for link prediction is to use distributed representation, i.e. each entity is identified by a compact vectorial signature. Link prediction is then done efficiently via a simple dot-product between signatures. To learn these signatures, several authors have proposed to represent these relational databases as tensors. Each relation can be represented by a sparse matrix and each of the sparse matrices can be stacked together to form a 3rd order tensor. Signatures are then extracted by tensor factorization techniques. The problem is that tensors are notably hard to approximate despite some recent effort to derive convex relaxations [9, 15, 7].

In this work, we introduce an alternative method based on *collective matrix factorization*, a novel technique to jointly factorize multiple relations [16, 21]. Unlike generic tensor-factorization techniques, we assume that two entity types do not share more than one relation, which is relevant in many real situations. This assumption leads to a *convex* formulation of the problem. More specifically, our method extends the (matrix) nuclear norm to a *collective nuclear norm* on a *set* of matrices representing an arbitrary number of relations in a dataset. The proposed norm is a strict generalization of the nuclear norm for a single matrix in the sense that they are equivalent to each other when there is a single relation.

We present a novel scalable algorithm based on an iterative *Singular Value Thresholding* algorithm. The main operation of the algorithm is to solve a partial eigenvalue problem, for which efficient (and distributed) code exists.

For the case of two relations amongst three entity types, our collective nuclear norm can be understood as the (matrix) nuclear norm applied to the concatenation of the matrices. But for the more general cases where three or more relations are involved in a data set, we argue that our extension to the collective nuclear norm is nontrivial, and differs in general from the nuclear norm applied to concatenated matrices, especially when the structure exhibits loops in the graph of relations.

Figure 1 illustrates such loopy structure (bottom) where there are three entity types E_1, E_2 and E_3, forming a loop through the three relations, $E_1 - E_2$,

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31.

$E_2 - E_3$ and $E_3 - E_1$, encoded in matrices $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$, respectively. A naive approach would be to apply the nuclear norm on the concatenated matrix[1] $\begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 \\ \cdot & \boldsymbol{X}_3 \end{bmatrix}$. However, such an approach would only model two out of the three relations (in this case $E_1 - E_2$ and $E_2 - E_3$ but not $E_3 - E_1$). In contrast, our proposed collective nuclear norm is based on the eigen-decomposition of the symmetric block matrix shown in the top of Figure 1; it naturally handles such loopy structure based on an appropriate convex formulation of the problem.

We further show that the proposed collective nuclear norm enjoys three essential properties including (1) inducing low-rank solutions, (2) its formulation as a decomposition norm and the weighted version of the norm, and (3) allowing us to derive an efficient *Singular Value Thresholding* (SVT) algorithm based on an iterative soft-thresholding of the eigen-decomposition of a block symmetric matrix. First, the proposed collective nuclear norm induces low-rank solutions, leading to automatic model selection for collective matrix factorization by allowing the solution of regularized pointwise estimations to have a lower dimension than the rank of the observation matrix. Second, the decomposition norm formulation enables one to directly interpret estimation problems regularized by the collective nuclear norm as *global* solutions of the original collective trace norm formulation of Singh and Gordon[16], Finally, the SVT algorithm originally designed for a single matrix is extended to our collective matrix factorization case so that we can find the global optimum efficiently.

In the following sections, we recall the nuclear norm definitions, extend them for convex collective matrix factorization, propose efficient algorithms for finding the global solutions, and demonstrate the superiority of the proposed convex collective matrix factorization on simulated and real-world datasets.

2 Nuclear Norm on Multiple Matrices

2.1 Nuclear Norm

The nuclear norm $\|\boldsymbol{X}\|_*$ of a rectangular matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ is defined as the sum of its singular values:

$$\|\boldsymbol{X}\|_* = \sum_{i=1}^{\min(n_1,n_2)} \varsigma_i(\boldsymbol{X}) = \frac{1}{2} \sum_{i=1}^{n_1+n_2} |\sigma_i(\mathcal{B}(X))| \quad , \quad (1)$$

[1] "." represents the block matrix that is missing

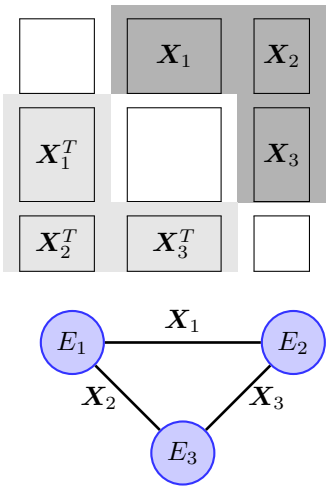

Figure 1: A symmetric block matrix representation of the collective nuclear norm applied on three relations exhibiting loopy relations among three entities E_1, E_2, E_3. Shaded blocks represent available relations (possibly with missing data in each relation), while unshaded blocks represent unavailable relations. The bottom graph represents the corresponding set of entities (nodes) and relations (edges).

where $N = n_1 + n_2$, $\mathcal{B}(X)$ is the symmetric matrix:

$$\mathcal{B}(X) := \begin{bmatrix} \boldsymbol{0} & \boldsymbol{X} \\ \boldsymbol{X}^T & \boldsymbol{0} \end{bmatrix} ,$$

$\varsigma_i(\boldsymbol{X})$ (resp. $\sigma_i(\mathcal{B}(X))$) is the i-th singular value (resp. eigenvalue) of the rectangular matrix $\boldsymbol{X}$ (resp. symmetric matrix $\mathcal{B}(X)$), both sorted in decreasing order. It is well known that $\mathcal{B}(X)$ has a symmetric spectrum, i.e. $\sigma_i(\mathcal{B}(X)) = -\sigma_{N-i}(\mathcal{B}(X))$, which simplifies the computation of the nuclear norm by summing over the set of positive eigenvalues of $\mathcal{B}(X)$; they are equal to the singular values of the rectangular matrix $\boldsymbol{X}$: $\varsigma_i(\boldsymbol{X}) = \sigma_i(\mathcal{B}(X))$ for $i = 1, \cdots, \min(n_1, n_2)$.

The nuclear norm can also be expressed as a decomposition norm [18, 2], i.e., a formulation that decomposes $\boldsymbol{X}$ into a product of two matrices:

$$\|\boldsymbol{X}\|_* = \frac{1}{2} \min_{\boldsymbol{U}\boldsymbol{V}^T = \boldsymbol{X}} \|\boldsymbol{U}\|_{\mathrm{F}}^2 + \|\boldsymbol{V}\|_{\mathrm{F}}^2 \quad (2)$$

where $\boldsymbol{U} \in \Re^{n_1 \times \min(n_1,n_2)}$, $\boldsymbol{V} \in \Re^{n_2 \times \min(n_1,n_2)}$, and $\|\cdot\|_{\mathrm{F}}^2$ the Frobenius norm. Finally, the nuclear norm can be equivalently defined as the value of a Semi-Definite Program (SDP) [18, 12]:

$$\|\boldsymbol{X}\|_* = \frac{1}{2} \min_{\boldsymbol{X}_1, \boldsymbol{X}_2} (\mathrm{trace}(\boldsymbol{X}_1) + \mathrm{trace}(\boldsymbol{X}_2))$$
$$\text{s.t.} \quad \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X} \\ \boldsymbol{X}^T & \boldsymbol{X}_2 \end{bmatrix} \in \mathcal{S}_N^+. \quad (3)$$

where $\mathcal{S}_N^+$ denotes the (convex) set of symmetric positive definite matrices of size $N \times N$.

2.2 Collective Nuclear Norm

We extend the nuclear norm described above and derive a convex collective factorization framework when we observe an arbitrary number of relations between pairs of entity types, i.e., relations can be represented by matrices. Given K entity types repesented by $\boldsymbol{E} = \{E_1, \cdots, E_K\}$, we assume that there are V relations (or views) where each relation $v \in \{1, \cdots, V\}$ is a pair $(r_v, c_v) \in \boldsymbol{E}^2$. Here, r_v and c_v are the indices of the row and column entity types, respectively. The set $\boldsymbol{X} = (\boldsymbol{X}_1, \cdots, \boldsymbol{X}_V)$ represents the V available relations, and takes values in the space:

$$\mathcal{X} = \Re^{n_{r_1} \times n_{c_1}} \otimes \cdots \otimes \Re^{n_{r_V} \times n_{c_V}}. \tag{4}$$

which is simply the V-ary Cartesian product of view-specific matrix spaces. We have two restrictions to the relational graph: (1) at most one relation can be defined between a given pair of entity types k and k' and (2) self-relations are allowed if they are symmetric, i.e. $r_v = c_v$ implies that $\boldsymbol{X}_v(i,j) = \boldsymbol{X}_v(j,i)$. The set of unique indices associated to a given entity type k is denoted by $\boldsymbol{i}_k = \{\bar{n}_{k-1} + 1, \cdots, \bar{n}_k\}$, where $\bar{n}_k = \sum_{i=1}^{k-1} n_i$ is the cumulative sum of the entities.

Given a matrix $\boldsymbol{A}$, the notation $\boldsymbol{A}(\boldsymbol{i}_k, \boldsymbol{i}_{k'})$ represents the $\Re^{|\boldsymbol{i}_k| \times |\boldsymbol{i}_{k'}|}$ matrix made up by the rows of $\boldsymbol{A}$ with indices $\boldsymbol{i}_k$ and columns with indices $\boldsymbol{i}_{k'}$. In addition, the notation $\boldsymbol{A}(\boldsymbol{i}_k, :)$ identifies the matrix made up by the rows of $\boldsymbol{A}$ with indices $\boldsymbol{i}_k$. We define the new *co-factorization nuclear norm* for multiple interlinked sources of data $\boldsymbol{X} \in \mathcal{X}$ below:

$$\|\boldsymbol{X}\|_\sharp = \frac{1}{2} \sum_{i=1}^{N} |\sigma_i(\mathcal{B}(\boldsymbol{X}))| \tag{5}$$

where $N = \bar{n}_{K+1}$ is the total number of entities in all the views and the function $\mathcal{B} : \mathcal{X} \to \boldsymbol{S}_N$ creates a symmetric block-matrix representing the set of views: formally, $\boldsymbol{S} = \mathcal{B}(\boldsymbol{X})$ if

- for all $v \in \{1, \cdots, V\}$, $\boldsymbol{S}(\boldsymbol{i}_{r_v}, \boldsymbol{i}_{c_v}) = \boldsymbol{X}_v$,
- for all $v \in \{1, \cdots, V\}$, $\boldsymbol{S}(\boldsymbol{i}_{c_v}, \boldsymbol{i}_{r_v}) = \boldsymbol{X}_v^T$ (by symmetry),
- $\boldsymbol{S}_{ij} = \boldsymbol{0}$ at any other entry with index (i,j), i.e. $\boldsymbol{S}(\boldsymbol{i}_k, \boldsymbol{i}_{k'}) = \boldsymbol{0}$ if there is no relation linking entity types k and k'.

Hence, the collective nuclear norm is a simple generalization of the matrix nuclear norm given in Equation (1), but the spectrum of the block-matrix $\mathcal{B}(\boldsymbol{X})$ is no longer symmetric, i.e., the negative eigenvalues do not have corresponding positive counterparts. As a toy example, we show in Figure 2 three relational matrices corresponding to 3 entity types with cardinalities $n_1 = 2$, $n_2 = 3$ and $n_3 = 4$, so that $N = 9$. All the relations are observed: $r_1 = r_3 = 1$, $c_1 = r_2 = 2$ and $c_2 = c_3 = 3$ and there are 3 non-zero eigenvalues.

$$\begin{aligned}
\boldsymbol{X}_1 &= \begin{bmatrix} 3 & 4 & 5 \\ 6 & 8 & 10 \end{bmatrix} \quad \boldsymbol{X}_2 = \begin{bmatrix} 18 & 21 & 24 & 27 \\ 24 & 28 & 32 & 36 \\ 30 & 35 & 40 & 45 \end{bmatrix} \\
\boldsymbol{X}_3 &= \begin{bmatrix} 6 & 7 & 8 & 9 \\ 12 & 14 & 16 & 18 \end{bmatrix} \\
\mathcal{B}(\boldsymbol{X}) &= \begin{bmatrix}
0 & 0 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 0 & 6 & 8 & 10 & 12 & 14 & 16 & 18 \\
3 & 6 & 0 & 0 & 0 & 18 & 21 & 24 & 27 \\
4 & 8 & 0 & 0 & 0 & 24 & 28 & 32 & 36 \\
5 & 10 & 0 & 0 & 0 & 30 & 35 & 40 & 45 \\
6 & 12 & 18 & 24 & 30 & 0 & 0 & 0 & 0 \\
7 & 14 & 21 & 28 & 35 & 0 & 0 & 0 & 0 \\
8 & 16 & 24 & 32 & 40 & 0 & 0 & 0 & 0 \\
9 & 18 & 27 & 36 & 45 & 0 & 0 & 0 & 0
\end{bmatrix} \\
\sigma(\mathcal{B}(\boldsymbol{X})) &= (118, 0, 0, 0, 0, 0, 0, -8.97, -109) \\
\|\boldsymbol{X}\|_\sharp &= \frac{1}{2}(|118| + |-8.97| + |-109|) = 117.8
\end{aligned}$$

Figure 2: Collective nuclear norm on a toy example.

We can show that $\|X\|_\sharp$ is indeed a norm:

Proposition 1 (Norm) *$\|X\|_\sharp$ is a norm on $\mathcal{X}$.*

Positive homogeneity, symmetry and separation properties are straightforward. Triangle inequality is obtained by using the SDP formulation and relaxing the constraint. The collective nuclear norm also admits a representation as a decomposition norm:

Proposition 2 (Decomposition norm)

$$\|\boldsymbol{X}\|_\sharp = \frac{1}{2} \min_{\{\boldsymbol{U}_{r_v} (\boldsymbol{U}_{c_v})^T = \boldsymbol{X}_v\}_{v=1}^V} \sum_{k=1}^{K} \|\boldsymbol{U}_k\|_F^2, \tag{6}$$

where $\boldsymbol{U}_k$ are latent matrices for the entity type k with n_k rows and N columns.

This can be shown by explicitely writing the eigen-decomposition of $\mathcal{B}(\boldsymbol{X})$. Note that Equation (6) is also a straightforward generalization of the standard nuclear norm characterization given in Equation (2). An interesting property of this formulation is that it has strong similarity with the existing work on collective matrix factorization [16], where the problem is directly parameterized by low-rank matrices: each view is constrained to have a factored representation

and the factors of a given entity type are shared across all the views in which this type is involved. This connection will be used in the following to derive an unconstrained gradient descent algorithm.

Finally we can show that the collective nuclear norm can also be obtained as the solution of a SDP, similarly to the (matrix) nuclear norm:

$$\|\boldsymbol{X}\|_\sharp = \frac{1}{4} \min_{\substack{\boldsymbol{Z}_1 \in \mathcal{S}_N^+, \boldsymbol{Z}_2 \in \mathcal{S}_N^+ \\ \{\boldsymbol{Z}_1(\boldsymbol{i}_{r_v}, \boldsymbol{i}_{c_v}) = \boldsymbol{X}_v\}_{v=1}^V \\ \{\boldsymbol{Z}_2(\boldsymbol{i}_{r_v}, \boldsymbol{i}_{c_v}) = -\boldsymbol{X}_v\}_{v=1}^V}} \operatorname{trace}(\boldsymbol{Z}_1) + \operatorname{trace}(\boldsymbol{Z}_2). \tag{7}$$

There are two SDP matrices Z_1 and Z_2 involved in this definition due to the negative eigenvalues of the block matrix $\mathcal{B}(\boldsymbol{X})$. One can see that for a single relation, the symmetry of the spectrum of $\boldsymbol{B}(\boldsymbol{X})$ implies that $Z_1 = Z_2$ at the optimum, leading to the matrix SDP formulation of Equation (3). Note that the linear objective implies that the SDP constraints are satisfied in a collective nuclear norm regularized problem, which implies that the norm is *low-rank inducing*, i.e., solutions of estimation problems regularized by this norm will tend to have low rank, similar to the matrix case.

Examples In Figure 1 and Figure 2, we illustrate loopy structured relations involving three entity types with three relations. A simple example of such a setting could be the interactions between users, websites and products: users access websites, websites display product ads and products are bought by users. As stated in Section 1, this example is important because a naive application of the nuclear norm on the concatenation of the matrices (illustrated by the block matrices colored with dark gray in Figure 1) would not be satisfactory in general. The top symmetric matrix in Figure 1 cannot be easily cast in the SDP formulation of the nuclear norm in Equation (3), showing that the co-factorization nuclear norm is not in general a special case of the nuclear norm applied to concatenated matrices.

Finally, it has recently been shown that the nuclear norm benefits from a weighted variant that significantly improves predictive performances [14]. This applies to our case as well.

3 Collective Nuclear Norm Regularization

Given that we have defined the collective nuclear norm, the problem of interest is to minimize a convex loss $\mathcal{O}_\lambda(\boldsymbol{X})$ regularized by this collective nuclear norm, i.e:

$$\min_{\boldsymbol{X} \in \mathcal{X}} \mathcal{O}_\lambda(\boldsymbol{X}), \quad \mathcal{O}_\lambda(\boldsymbol{X}) = \ell(\boldsymbol{X}) + \lambda \|\boldsymbol{X}\|_\sharp, \tag{8}$$

where $\ell(\boldsymbol{X})$ is the loss function (e.g., the negative log-likelihood function) of the noisy observations, λ is the regularization parameter.

To formalize the matrix completion problem (a.k.a. data imputation), we define the projection $P_\Omega : \mathcal{X} \to \mathcal{X}$ which maps the set of matrices equal to zeros outside the observed values defined by the set Ω, so that the (i,j)-th component of the view v in $P_\Omega(\boldsymbol{X})$ is equal to the element indexed by the i-th row and j-th column in the view $\boldsymbol{X}_v$ if (v,i,j) is in Ω and zero otherwise. Using the squared Frobenius norm loss function, the problem (8) becomes:

$$\min_{X \in \mathcal{X}} \|P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{Y})\|_F^2 + \lambda \|\boldsymbol{X}\|_\sharp \ , \tag{9}$$

where $\boldsymbol{Y} \in \mathcal{X}$ is the set of observed matrices. The strength of regularization $\lambda > 0$ can be optimized on held-out data removed from the initial set of observations Ω. This problem could also be written as the constrained optimization problem $\min_{X \in \mathcal{X}} \|\boldsymbol{X}\|_\sharp$ under the constraint $\|P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{Y})\|_F \leq \epsilon$, where ϵ is chosen by cross-validation.

In the following, we propose two algorithms to solve the optimization problem: one is based on the singular value thresholding algorithm, and the other is based on the unconstrained minimization by stochastic gradient descent, which are described in the following subsections.

3.1 Algorithm 1: Singular Value Thresholding

For the standard matrix nuclear norm (i.e., on a single matrix), the solution to least-square problems regularized by the nuclear norm can be found in terms of SVD with a shrinkage of the eigenvalues [3]. If the data matrix is fully observed, a single SVD is needed; otherwise a simple first order iterative singular value thresholding (SVT) algorithm, sometimes called Proximal Forward-Backward Splitting [5, 4] can be derived, alternating between the SVD computation and the imputation of missing values. Now we show SVT can be extended to the collective matrix factorization framework based on the connection between the SVD solution and its corresponding SDP formulation, which leads to an eigenvalue decomposition of the symmetric block-matrix discussed in Section 2.

Definition 1 *The co-factorization thresholding operator applied to a set of matrices* $\boldsymbol{X} = (\boldsymbol{X}_1, \cdots, \boldsymbol{X}_V)$ *is defined as:*

$$\begin{aligned} \boldsymbol{S}_\lambda(\boldsymbol{X}) &:= (\boldsymbol{L}_{r_1} \boldsymbol{D} \boldsymbol{L}_{c_1}^T, \cdots, \boldsymbol{L}_{r_V} \boldsymbol{D} \boldsymbol{L}_{c_V}^T), \\ D &= \operatorname{diag}(\{S_\lambda(\sigma_i)\}_{i=1}^N) \end{aligned} \tag{10}$$

where $\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}_T = \mathcal{B}(\boldsymbol{X})$ *is the eigen-decomposition of the symmetric block matrix* $\mathcal{B}(\boldsymbol{X})$*,* $S_\lambda : \Re \mapsto \Re$ *is the soft thresholding function* $S_\lambda(x) = \text{sign}(x)\max(|x| - \lambda, 0))$ *and* $\boldsymbol{L}_k = \boldsymbol{L}(\boldsymbol{i}_k, :)$*.*

The computation of $\boldsymbol{S}_\lambda(\boldsymbol{X})$ corresponds to the shrinkage of the positive and negative eigenvalues of the symmetric matrix $\mathcal{B}(\boldsymbol{X})$ toward 0, keeping the same eigenvectors. In the toy example of Figure 2, choosing $\lambda = 10$ reduces the rank of $\mathcal{B}(\boldsymbol{X})$ from 3 to 2, because $S_\lambda(\sigma(\mathcal{B}(\boldsymbol{X}))) = (108, 0, 0, 0, 0, 0, 0, 0, -99)$. The following results shows that $S_\lambda(\boldsymbol{X})$ is the proximity operator of $\lambda\|\boldsymbol{X}\|_\sharp$:

Proposition 3 *For every* $\lambda \geq 0$ *and every* $Y \in \mathcal{X}$*, the co-factorization thresholding operator* (10) *satisfies:*

$$S_\lambda(\boldsymbol{Y}) = \arg\min_{\boldsymbol{X}} \frac{1}{2}\sum_{v=1}^{V} \|\boldsymbol{X}_v - \boldsymbol{Y}_v\|_F^2 + \lambda\|\boldsymbol{X}\|_\sharp. \quad (11)$$

Based on the result in Proposition 3, one can directly use Equation (10) to solve the Equation (9) for fully observed relations. For partially observed relations, i.e., when some matrices Y_v have missing values, the solution of Equation (9) is characterized by the fixed point equation $\boldsymbol{X} = \boldsymbol{S}_{\lambda\gamma}(\boldsymbol{X} + \gamma P_\Omega(\boldsymbol{Y} - \boldsymbol{X}))$ for $\gamma > 0$. Algorithm 1 implements this procedure.

Algorithm 1 Singular Value Thresholding for Co-Factorization (Partial observations)

1: **INPUT:** Observations Ω, values $\boldsymbol{Y}$, λ
2: **OUTPUT:** $X^{(T)}$ approximating Eq. (9)
3: INITIALIZE $\boldsymbol{Z}^{(0)} = (\boldsymbol{0}, \cdots, \boldsymbol{0})$ to the zero matrix set in $\mathcal{X}$
4: **for** $t = 1, 2, \cdots, T$ **do**
5: $\quad(\boldsymbol{X}_1^{(t)}, \cdots, \boldsymbol{X}_V^{(t)}) = \boldsymbol{S}_{\lambda\gamma_t}(\boldsymbol{Z}^{(t-1)})$
6: $\quad$**for** $v = 1, 2, \cdots, V$ **do**
7: $\quad\quad\boldsymbol{Z}_v^{(t)} = \boldsymbol{X}_v^{(t)} + \gamma_t P_\Omega(\boldsymbol{Y}_v - \boldsymbol{X}_v^{(t)})$
8: $\quad$**end for**
9: **end for**

Proposition 4 *For a sequence* $(\gamma_t)_{t\in\mathbb{N}}$ *such that* $\inf_{t\in\mathbb{N}} \gamma_t > 0$ *and* $\sup_{t\in\mathbb{N}} < \frac{2}{\lambda}$*, the output* $\boldsymbol{X}^{(T)}$ *of Algorithm 1 converges to the solution of* (9)*.*

Proof Since $\boldsymbol{S}_\lambda$ is a proximity operator, this result is a direct application of the SVT convergence Theorem (Theorem 3.4 in [4]) where the Lipchitz constant of the regularizer is equal to λ. □

3.2 Algorithm 2: Unconstrained Minimization with Collective Nuclear Norm

Our second algorithm is based on the decomposition norm formulation of the norm (Proposition 2) that allows one to express the objective as an unconstrained minimization problem. Plugging the solution of Equation (6) into Equation (8) leads to:

$$\{\hat{\boldsymbol{U}}_k\}_k \in \underset{\{\boldsymbol{U}_k \in \Re^{n_k \times N}\}_k}{\arg\min} \sum_{v=1}^{V} \ell\left(\boldsymbol{U}_{r_v}\boldsymbol{U}_{c_v}^T\right) + \lambda\sum_{k=1}^{K}\|\boldsymbol{U}_k\|_{\text{F}}^2 \quad (12)$$

The matrices $\boldsymbol{U}_k$, $k = \{1, \cdots, K\}$ can be interpreted as the feature representations of the entities of type k. Note that if we restrict the number of columns of the matrices $\boldsymbol{U}_k$ to a fixed number $r \leq N$, this objective function matches exactly the objective function for collective matrix factorization [16]. Hence, by using the fact that the nuclear norm is low-rank promoting, the standard algorithm in [16] can be used to solve Equation (12) provided the selected rank r is larger than $\text{rank}(\hat{\boldsymbol{U}})$, that is, the rank of the optimal solution. Although Equation (12) is a non-convex problem, it can lead to good performances in practice, similarly to the matrix case (single relation) to estimate the solution nuclear-norm penalized problems. In particular, an alternative least square procedure could be implemented by optimizing one factor U_k given the others. In this work, we used stochastic gradient descent (SGD) to scale to a large number of observations.

4 Experiments

4.1 Simulations

We generated datasets using the loopy structure with 3 relations illustrated in Figure 1 by randomly generating low-rank view-specific matrices $\boldsymbol{X}_v$ (using standard normal variables as latent factors) with dimensions $n_1 = 20$, $n_2 = 30$ and $n_3 = 40$. Standard noise with unit variance was added to each observation to generate $\boldsymbol{Y} \in \mathcal{X}$. The regularization was selected by cross-validating the data imputation error on held-out values. We randomly selected 50% of observations to illustrate the ability to recover the true matrix. Root Mean Square Error (RMSE) was used to compute the matrix recovery error, i.e. $\sqrt{\sum_v \|\hat{\boldsymbol{X}}_v - \boldsymbol{X}_v^*\|_F^2}$, where $\hat{\boldsymbol{X}}_v$ (resp. $\boldsymbol{X}_v^*$) represents the estimated (resp. simulated) matrix . The SVT algorithm was used. Iterations were stopped when the relative improvement was lower than 1e-5. Results for various rank values are shown in Table 1. They confirm that the collective

Table 1: Comparison of independent matrix factorization vs. collective matrix factorization on simulations on a loopy structure with 3 relations. Errors are computed based on 10 random experiments.

Rank	error Indep.	error Collective
2	1.21 ± 0.346	0.673 ± 0.120
5	2.95 ± 0.354	2.39 ± 0.342
10	5.81 ± 0.701	5.34 ± 0.611

learning of multiple relations gives lower generalization error, compared to independent factorization of the views.

4.2 Data imputation experiments

To evaluate the performance of the proposed convex collective factorization, we conduct empirical evaluations on two real data sets: MovieLens and Flickr. We choose the MovieLens 1 Million Rating data set[2] that involves 6,000 users' ratings for 4,000 movies. Ratings are integers ranging from 1 to 5, and each rating is associated with a timestamp. Additionally, there is demographic information (e.g., age, gender, occupation, etc.) associated with users, and descriptors (e.g., titles, release dates, genres, etc.) associated with movies. We restrict the user features to the age, the gender, and the occupation, and only consider the genre to describe movies. Ages are partitioned into 7 groups, encoded by a 7-dimensional binary vector. The gender, user occupation and movie genre are also represented by binary vectors. There are three relations for the MovieLens data set: (user, movie), (user, profile), (movie, genre). For the first relation, the date 2001/01/01 is chosen to split the data into training and testing data, leading to 904,796 ratings as training data, and 95,413 as testing data. For the second and third relations, we randomly select 10% of them for testing and use the rest as training. Our objective is to predict ratings, unobserved user features and unobserved movie genres.

For the Flickr data set[3], we crawl data with the social-network connectors (i.e., Flickr API). In this data set, there are 2,866 users, 60,339 tags, 32,752 comment terms and 46,733 items (e.g., images). We study five relations: user-user interaction $C1$, user-tag $C2$, tag-item $C3$, user-tagged item $C4$, and item-item feature $C5$. Observations for relations $C1, C2, C3, C4$ are binary values where the observed pairs are positive samples (r=1). For each positive sample, we randomly generate 50 negative samples (r=0). In $C5$, we represent each item (image) by a 1024-dimension Fisher vector, each component of which is a real number. For all of these five relations, we randomly select 10% as testing data set, and use the rest as training data set. The performance measurements are the RMSE and negative log-likelihood.

Singular Value Shrinkage vs. Stochastic Gradient Descent In Table 2, we compared the proposed optimization algorithms (SGD and SVT) with the standard matrix factorization on three views for the MovieLens 1M rating data set. Results indicated similar predictive performances for the two proposed algorithms, both of which outperformed the individual matrix factorization on the user-item rating view. The error seemed to be higher in predicting the user-feature view. This was mainly due to the fact that the loss was dominated by the user-item ratings view as there are much more observations in this view.

Efficiency is important for real-world applications, and we reported the efficiency of the two algorithms in Figure 3 (left). Results indicated that the SVT algorithm outperformed SGD by orders of magnitudes in efficiency, even if SGD was implemented in an optimized C code and SVT was implemented in Matlab without code optimization.

Table 2: RMSE for MovieLens 1M data set split dynamically with 90% for training.

View	MF	SGD	SVT
user-item ratings	0.9020	0.8926	0.8962
user-feature	0.2781	0.3206	0.2916
item-feature	0.2955	0.3218	0.2825

Regularized Low Rank Approximations with Large Rank For the MovieLens 1M data set, we computed the RMSE of the SGD algorithm for various values of the rank and the regularization term for the testing data. Intuitively, the nuclear norm regularized solution should appear at the upper limit of the rank. To illustrate this point, we experimented with different rank r and regularization λ, and reported the results in Figure 3. We observed that the best performances were obtained for a regularization $\lambda = 0.1$, and the performances were not degrading when r increased, indicating that the solution remained at a constant rank. This confirms the fact that one does not need to impose the exact low-rank constraint to obtain the best performance.

Comparisons with Independent Matrix Factorization To illustrate the benefit of *jointly* factorizing multiple relations collectively, we compared the proposed convex collective matrix factorization with

[2] `http://www.grouplens.org/node/73`

[3] Flickr is a social photo bookmarking site.

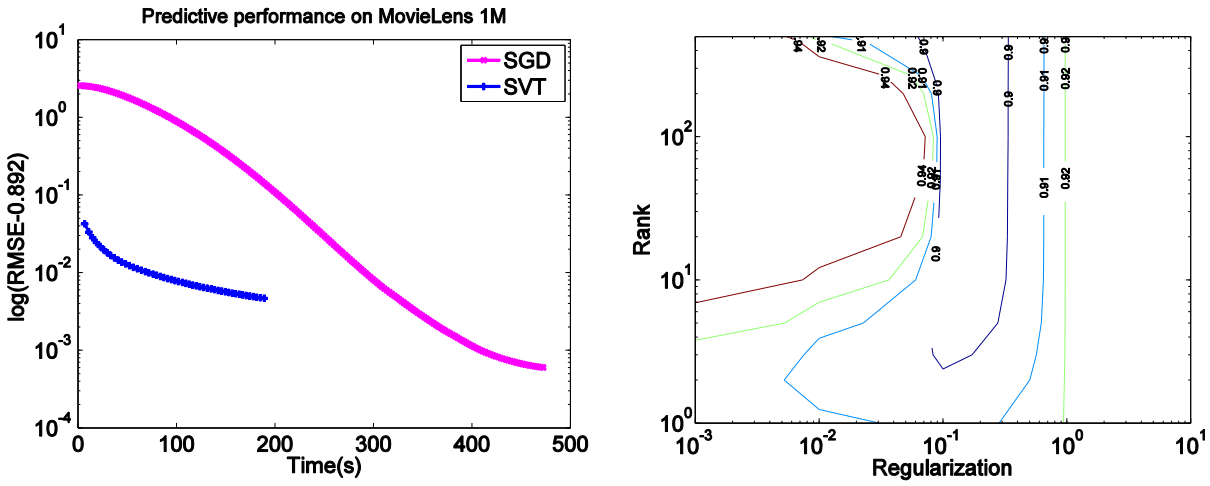

Figure 3: RMSE on the MovieLens 1M rating data set split temporally with 90% data for training with SGD and SVT. Left: Convergence of RMSE vs. iterations for SGD and SVT; Right: RMSE for different dimension of factors representing each entity and regularization.

independent probabilistic matrix factorizations (i.e., factorizing each relation independently) [13]. The results (Table 3(a)) indicated that the performance of our model was consistently better than the traditional probabilistic matrix factorization for the tasks of predicting ratings and item features across all settings (e.g., proportion of data used to training, static vs. dynamic testing). Note that we took into account the temporal information associated with ratings, leading to a dynamic setting that respects the temporal property when dividing the training/testing data.

Analysis of the convergence of the SGD algorithm We discussed earlier, one property of Algorithm 2 based on SGD is that a global minimum can be obtained by gradually increasing the rank of the solution and continuing to conduct the stochastic gradient descent algorithm until we obtain a rank-deficient local minimum. To illustrate this point, we plotted the objective function against the latent dimension for the MovieLens data in Figure 4(a). Results demonstrated that the objective function remained almost the same after $r = 200$. The slight decrease of the function for $r \geq 200$ was due to the fact that the SGD algorithm has not completely converged.

When using collective nuclear norm as regularization, our results (the four right most columns in Figure 4(b)) demonstrated that the weighted version of the collective factorization nuclear norm achieved better generalization performances than that of the unweighted collective nuclear norm.

Finally, we studied if the proposed SGD-based collective nuclear norm is sensitive w.r.t. the random initializations of the latent matrices representing the entity types. We randomly initialized the variables $\boldsymbol{U}_1, \cdots, \boldsymbol{U}_K$ for $r = 10$ and $r = 200$, and plotted the statistics of 10 repeated experiments in Figure 4(c). We observed that with a sufficiently large latent dimension $r = 200$, the objective function enjoyed a small variance, whereas smaller r caused large variance of the results. This illustrated the fact that the predictive performances of regularized collective matrix factorization became less sensitive to the initialization when r increased.

5 Related Work

To address the problem of jointly modeling multiple relations for factorization, several approaches [11, 22, 1, 20] have recently been proposed. Singh and Gordon [17] perhaps proposed the most generic view of this problem by introducing the notation of collective matrix factorization. In this model, parameter estimation is achieved by maximum *a posteriori* (MAP). A Bayesian version of collective matrix factorization have been proposed [17], where inference is based on a Hessian-based hybrid Monte-Carlo procedure. Note that this sampling-based inference can be much slower than the MAP estimation, considering that collaborative prediction problems are often designed for large scale problems and thus require a relatively large number of samples to be accurate. Amongst these existing works, [10] is perhaps the most similar work with our but their objective is not convex. Compared to tensor factorization techniques for interlinked data[6] for which convex formulations have also been proposed [9, 15, 7], our approach relies strongly on the

Table 3: RMSE on MovieLens and Flickr. Left (a): randomly selected data (Static) vs. Temporally split data (Dynamic) for MovieLens. Right (b): performances on the five relations for Flickr data with different λ and $r = 200$ trained with SGD for the weighted collective nuclear norm.

% train	Model	Static	Dyna.
90%	CCMF	0.9287	1.0588
	PMF	0.9347	1.2159
80%	CCMF	0.9501	2.2722
	PMF	0.9540	3.2984
50%	CCMF	1.0320	3.0931
	PMF	1.0415	4.2614

view	no reg	0.0001	0.001	0.1	10
C1	8.3895	8.3619	5.069	0.7138	0.7071
C2	8.1373	8.1081	4.6362	0.229	0.2473
C3	13.7447	13.7402	12.8806	0.4758	0.3661
C4	9.7138	9.6808	5.1036	0.2246	0.2384
C5	10.6246	10.5901	5.4967	1.0179	1.0033

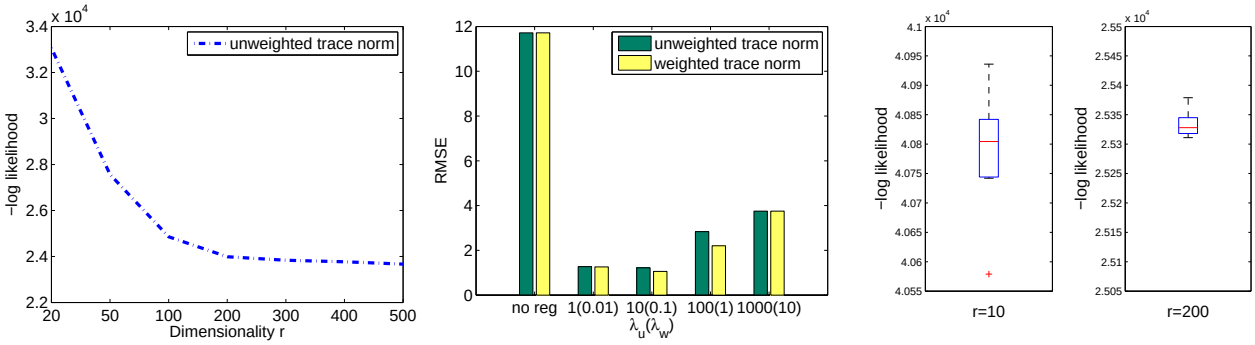

Figure 4: Explanatory experimental results for different aspects exhibited in the MovieLens data set. Left (a): Convergence of likelihood as r growing; Middle (b): Unweighted nuclear norm vs. weighted nuclear norm with comparable results; Right (c): sensitivity test based on 10 random initializations. The variance for $r = 10$ is 9272.9 but the variance for $r = 200$ is only 410.3.

fact that for two given entity types, there is at most one relation; this assumption does not seem to be too restrictive in practice.

6 Conclusion

In this paper, we proposed a novel collective nuclear norm for jointly factorizing multiple interlinked matrices representing relations amongst multiple entity types. It can leverage information across multiple relations to learn a meaningful representation of each entity; more importantly, the optimization problem based on this new collective nuclear norm is a convex optimization problem, and we proposed a new efficient algorithm based on singular value thresholding or gradient descent to find the solution. We provided empirical evidence on two real-world predictive tasks. We demonstrated with empirical results that the algorithms based on convex optimization can be much faster than the ones based on SGD, even for large datasets. Future work include better modeling capabilities to allow multiple relation between two entity types and non-symmetric self-relations. We are also considering robust versions of the problem, including multiple norms regularization (sparsity component, matrix-specific norms), and adapting recent greedy coordinate descent methods (Frank-Wolfe algorithms) to our case [8].

7 Acknowledgments

The research leading to these results has received funding from the European Commission Seventh Framework Programme (FP/2007-2013) through the projects Fupol and Fusepool.

References

[1] D. Agarwal, B.-C. Chen, and B. Long. Localized factor models for multi-context recommendation. In *SIGKDD*, pages 609–617, 2011.

[2] F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. Technical Report UMR 854, CNRS/ENS/INRIA, 2008.

[3] J. Cai, E. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, 2008.

[4] P. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4:1168, 2005.

[5] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

[6] T. Franz, A. Schultz, S. Sizov, and S. Staab. Triplerank: Ranking semantic web data by tensor decomposition. In *The Semantic Web-ISWC 2009*, pages 213–228. Springer, 2009.

[7] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[8] M. Jaggi and M. Sulovský. A simple algorithm for nuclear norm regularized problems. In J. Frnkranz and T. Joachims, editors, *ICML*, pages 471–478. Omnipress, 2010.

[9] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM review*, 51(3), 2009.

[10] C. Lippert, S. H. Weber, Y. Huang, V. Tresp, M. Schubert, and H.-P. Kriegel. Relation prediction in multi-relational domains using matrix factorization. In *NIPS Workshop on SISO*, 2008.

[11] B. Long, Z. M. Zhang, X. Wú, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML*, pages 585–592, 2006.

[12] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52:471–501, 2010.

[13] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS 21*, pages 1257–1264. 2008.

[14] R. Salakhutdinov and N. Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *NIPS 23*, pages 2056–2064. 2010.

[15] M. Signoretto, L. De Lathauwer, and J. Suykens. Nuclear norms for tensors and their use for convex multilinear estimation. *ESAT-SISTA, KU Leuven, Tech. Rep*, pages 10–186, 2010.

[16] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *SIGKDD*, pages 650–658, 2008.

[17] A. P. Singh and G. J. Gordon. A bayesian matrix factorization model for relational data. In P. Grünwald and P. Spirtes, editors, *In UAI*, 2010.

[18] N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *COLT*, pages 545–560, 2005.

[19] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[20] Y. K. Yilmaz, A. T. Cemgil, and Umutşimşekli. Generalised coupled tensor factorisation. In *NIPS 25*. 2011.

[21] D. Yin, S. Guo, B. Chidlovskii, B. Davison, C. Archambeau, and G. Bouchard. Connecting comments and tags: improved modeling of social tagging systems. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 547–556. ACM, 2013.

[22] Y. Zhang, B. Cao, and D.-Y. Yeung. Multi-domain collaborative filtering. In *UAI*, pages 725–732, 2010.