

# Symmetric Non-negative Latent Factor Models for Undirected Large Networks

Xin Luo, Ming-Sheng Shang

Institute of Green and Intelligence Technology, Chinese Academy of Sciences  
 Chongqing, China  
 {luoxin21, msshang}@cigit.ac.cn

## Abstract

Undirected, high dimensional and sparse networks are frequently encountered in industrial applications. They contain rich knowledge regarding various useful patterns. Non-negative latent factor (NLF) models have proven to be effective and efficient in acquiring useful knowledge from asymmetric networks. However, they cannot correctly describe the symmetry of an undirected network. For addressing this issue, this work analyzes the NLF extraction processes on asymmetric and symmetric matrices respectively, thereby innovatively achieving the symmetric and non-negative latent factor (SNLF) models for undirected, high dimensional and sparse networks. The proposed SNLF models are equipped with a) high efficiency, b) non-negativity, and c) symmetry. Experimental results on real networks show that they are able to a) represent the symmetry of the target network rigorously; b) maintain the non-negativity of resulting latent factors; and c) achieve high computational efficiency when performing data analysis tasks as missing data estimation.

## 1 Introduction

Undirected, high dimensional and sparse networks are frequently encountered in industrial applications. They contain rich knowledge regarding various useful patterns. Non-negative latent factor (NLF) models have proven to be effective and efficient in acquiring useful knowledge from asymmetric networks. However, they cannot correctly describe the symmetry of an undirected network.

For addressing this issue, this work analyzes the NLF extraction processes on asymmetric and symmetric matrices respectively, thereby innovatively achieving the symmetric and non-negative latent factor (SNLF) models for undirected, high dimensional and sparse networks. The proposed SNLF models are equipped with a) high efficiency, b) non-negativity, and c) symmetry. Experimental results on real networks show that they are able to a) represent the symmetry of the target network rigorously; b) maintain the non-negativity of resulting latent factors; and c) achieve high

computational efficiency when performing data analysis tasks as missing data estimation.

With the information explosion, more and more entities are involved in various industrial applications like social network services [Ghahramani, 2015; Yang et al., 2015], e-commerce systems [Adomavicius and Tuzhilin, 2005; Luo et al., 2015], bioinformatics applications [You et al., 2010; Hofree et al., 2013], internet analysis [Liao et al., 2013] and wireless sensor networks [Piao et al., 2014; Nguyen and Shin, 2016]. Due to the difficulty in observing the full relationship among numerous entities, high dimensional and sparse (HiDS) matrices describing partial relationship among them are frequently encountered.

In spite of their sparsity, HiDS matrices contain rich information regarding various desired patterns, e.g., user preferences in recommender systems [Adomavicius and Tuzhilin, 2005; Luo et al., 2015], protein connections in protein interactomes [You et al., 2010; Hofree et al., 2013], relative distances among unconnected nodes in wireless sensor networks [Piao et al., 2014; Nguyen and Shin, 2016], and potential communities in social networks [Yang et al., 2015]. Great efforts have been made for acquiring such knowledge from HiDS matrices, resulting in various sophisticated models. Among them, one important kind of models are the latent factor (LF)-based ones [Koren et al., 2009]. Originated from matrix factorization (MF)-based techniques [Lee and Seung, 1999], an LF model seeks for the low-rank approximation to a HiDS matrix. It works by a) mapping a HiDS matrix's column and row entities into the same LF space with low dimension, b) designing the objective function based on known entries of this HiDS matrix with respect to the desired LFs, c) extracting the desired LFs through minimizing the objective function, and d) applying the obtained LFs to various data analysis tasks.

On HiDS matrices where most entries are unknown, an LF model focusing on its known entries can achieve high computational and storage efficiency as well as competitive performance in various data analysis tasks [Koren et al., 2009; He et al., 2011]. However, most LF models do not fulfill the non-negativity constraints, i.e., their resulting LFs might be negative. Nonetheless, industrial data like the social network trusts [Yang et al., 2015] are commonly defined to be positive. When addressing such data, a non-negative model

can precisely describe hidden patterns [Lee and Seung, 1999; Kim and Park, 2007].

For extracting non-negative LFs from HiDS matrices, the NLF model is proposed [Luo et al., 2014; Luo et al., 2016]. In accordance with the principle of LF analysis [Koren et al., 2009], the NLF model transforms the matrix manipulation-based non-negatively multiplicative update [Lee and Seung, 1999] into single LF dependent multiplicative update [Luo et al., 2014], thereby extracting non-negative LFs from a HiDS matrix with its known entries only. Thus, non-negative LFs can be obtained from a HiDS matrix efficiently.

However, the NLF model does not consider the possible symmetry of a HiDS matrix. Symmetric, high dimensional and sparse (SHiDS) matrices correspond to undirected, high dimensional and sparse networks, which are frequently encountered in real applications focusing on the relationship among entities belonging to one mono set, e.g., social network services among a mono set of users [Ghahramani, 2015]. For addressing non-negative, symmetric and full matrices, the symmetric and non-negative matrix factorization (SNMF) model is proposed [Wang et al., 2008]. Its idea is also involved in other tasks related with the non-negative factorization of symmetric and full matrices [He et al., 2011; Huang et al., 2014]. Nonetheless, to the authors' best knowledge, there are no prior works regarding extracting non-negative LFs from SHiDS matrices.

This work aims at designing symmetric and non-negative LF (SNLF) models for SHiDS matrices corresponding to undirected, high dimensional and sparse networks. With such models, one can extract non-negative LFs from SHiDS matrices with high efficiency in computation and storage, as well as reflecting their symmetry. The main contributions of this work include:

- Two symmetric non-negative latent factor models for non-negative LF analysis on SHiDS matrices;
- Detailed theoretical analysis regarding the connections between the SNLF and asymmetric NLF models;
- Empirical studies on four SHiDS matrices from real applications.

The rest of this paper is organized as follows: Section 2 states the problem of building SNLF models; Section 3 presents the SNLF models; Section 4 gives the experimental results; Section 5 discusses; and finally, Section 6 concludes this paper.

## 2 Problem Statement

An undirected network describes the relationships among entities belonging to one mono set. Let  $N$  denote this entity set, then such a network can be denoted by a symmetric matrix  $R$  defined as follows,

**Definition 1.** Given  $N$ ,  $R^{N \times N}$  describes certain kind of relationships among entities in  $N$ , whose entries  $r_{ij}=r_{ji}$  for each pair of entities  $i, j \in N$ .

Once the undirected network is high dimensional and sparse, which is frequently encountered in industrial applications where only incomplete relationships can be observed [Wang et al., 2008; Szklarczyk et al., 2015], then  $R$

becomes a SHiDS matrix. In other words, let  $\Lambda$  and  $\Gamma$  denote the known and unknown entries of  $R$ , then we have  $|\Lambda| \ll |\Gamma|$ .

For extracting useful knowledge from  $R$ , an SNLF model seeks for its rank- $d$  approximation  $\hat{R}$  as defined next:

**Definition 2.** Given  $R$  and  $\Lambda$ , an SNLF model is  $R$ 's rank- $d$  approximation  $\hat{R} = A \cdot A^T$  with  $A^{N \times d} \geq 0$  being the non-negative LF matrix and  $d \ll |N|$ .

Note that  $d$  can also be interpreted as the dimension of the LF space, and  $A$  is the non-negative LF matrices hidden in  $\Lambda$ .  $A$  describes the latent characteristics of entities in  $N$ . It can be constructed by minimizing the objective function measuring the difference between  $\Lambda$  and corresponding entries in  $\hat{R}$ . For instance, with the most commonly seen Euclidean distance [Wang et al., 2008; Huang et al., 2014], such an objective function is formulated by:

$$\arg \min_A \varepsilon(A) = \frac{1}{2} \sum_{r_{i,j} \in \Lambda} \left( r_{i,j} - \sum_{k=1}^d a_{i,k} a_{j,k} \right)^2, \quad (1)$$

$$s.t. \quad \forall i, j \in N, k \in \{1, 2, \dots, d\} : a_{i,k} \geq 0, a_{j,k} \geq 0;$$

where  $\varepsilon(A)$  denotes the objective function regarding  $A$ ,  $r_{i,j}$ ,  $a_{i,k}$  and  $a_{j,k}$  denote corresponding entries in  $R$  and  $A$ , respectively. Note that this work adopts the Euclidean distance (1) to present the methods. However, the presented methods are also compatible with other loss functions like the Kullback-Leibler divergence.

As unveiled by prior works [Koren et al., 2009], (1) is ill-posed. It should be regularized to avoid overfitting. With the Tikhonov regularization, (1) is extended to:

$$\arg \min_A \varepsilon(A) = \frac{1}{2} \sum_{r_{i,j} \in \Lambda} \left( r_{i,j} - \sum_{k=1}^d a_{i,k} a_{j,k} \right)^2 + \lambda \left( \sum_{k=1}^d a_{i,k}^2 + \sum_{k=1}^d a_{j,k}^2 \right), \quad (2)$$

$$s.t. \quad \forall i, j \in N, k \in \{1, 2, \dots, d\} : a_{i,k} \geq 0, a_{j,k} \geq 0;$$

where  $\lambda$  denotes the regularizing coefficient for  $A$ .

## 3 Methods

### 3.1 Symmetric Non-negative Latent Factor Model

First of all, we consider building a symmetric non-negative latent factor (SNLF) model on (2). To do so, we firstly apply the additive gradient descent (AGD) to (2) to obtain:

$$\arg \min_A \varepsilon(A) \stackrel{AGD}{\Rightarrow} \forall i, j \in N, k \in \{1, 2, \dots, d\} :$$

$$a_{i,k} \leftarrow a_{i,k} + \eta_{i,k} \sum_{j \in \Lambda(i)} \left( a_{j,k} \left( r_{i,j} - \sum_{k=1}^d a_{i,k} a_{j,k} \right) - \lambda a_{i,k} \right), \quad (3)$$

$$a_{j,k} \leftarrow a_{j,k} + \eta_{j,k} \sum_{i \in \Lambda(j)} \left( a_{i,k} \left( r_{i,j} - \sum_{k=1}^d a_{i,k} a_{j,k} \right) - \lambda a_{j,k} \right).$$

$$\text{In (3),} \quad -\eta_{i,k} \sum_{j \in \Lambda(i)} \left( a_{j,k} \sum_{k=1}^d a_{i,k} a_{j,k} + \lambda a_{i,k} \right) \quad \text{and}$$

$$-\eta_{j,k} \sum_{i \in \Lambda(j)} \left( a_{i,k} \sum_{k=1}^d a_{i,k} a_{j,k} + \lambda a_{j,k} \right) \quad \text{are the negative terms.}$$

Then  $\eta_{i,k} = a_{i,k} / \left( \sum_{j \in \Lambda(i)} \left( a_{j,k} \sum_{k=1}^d a_{i,k} a_{j,k} + \lambda a_{i,k} \right) \right)$  and  $\eta_{j,k} = a_{j,k} / \left( \sum_{i \in \Lambda(j)} \left( a_{i,k} \sum_{k=1}^d a_{i,k} a_{j,k} + \lambda a_{j,k} \right) \right)$  are adopted,

following the principle of single LF dependent multiplicative update [Luo et al., 2014] to obtain:

$$\begin{aligned} \arg \min_A \varepsilon(A) &\stackrel{\text{Single LF-dependent Multiplicative Update}}{\Rightarrow} \forall i, j \in N, k \in \{1, 2, \dots, d\}: \\ a_{i,k} &\leftarrow a_{i,k} \frac{\sum_{j \in \Lambda(i)} a_{j,k} r_{i,j}}{\sum_{j \in \Lambda(i)} \left( a_{j,k} \sum_{k=1}^d a_{i,k} a_{j,k} + \lambda a_{i,k} \right)}, \\ a_{j,k} &\leftarrow a_{j,k} \frac{\sum_{i \in \Lambda(j)} a_{i,k} r_{i,j}}{\sum_{i \in \Lambda(j)} \left( a_{i,k} \sum_{k=1}^d a_{i,k} a_{j,k} + \lambda a_{j,k} \right)}. \end{aligned} \quad (4)$$

In (4), the update rules for  $a_{i,k}$  and  $a_{j,k}$  are equivalent. Hence, we fold (4) into the following concise form:

$$\begin{aligned} \arg \min_A \varepsilon(A) &\stackrel{\text{Single LF-dependent Multiplicative Update}}{\Rightarrow} \forall i \in N, k \in \{1, 2, \dots, d\}: \\ a_{i,k} &\leftarrow a_{i,k} \frac{\sum_{j \in \Lambda(i)} a_{j,k} r_{i,j}}{\sum_{j \in \Lambda(i)} \left( a_{j,k} \sum_{k=1}^d a_{i,k} a_{j,k} + \lambda a_{i,k} \right)}. \end{aligned} \quad (5)$$

(5) leads to an SNLF model with the following characteristics: a) given that  $\hat{R} = A \cdot A^T$ ,  $\hat{r}_{i,j} = \sum_{k=1}^d a_{i,k} a_{j,k} = \hat{r}_{j,i} = \sum_{k=1}^d a_{j,k} a_{i,k}$  always stands for  $\forall i, j \in N$ ; and b) the multiplicative update keeps the non-negativity of involved LFs in  $A$ .

### 3.2 Connections between SNLF and NLF

As discussed in Section 3.1, SNLF extracts a unique LF matrix, i.e.,  $A$ , which ensures the symmetry of the resulting low-rank approximation to the target SHiDS matrix with  $\hat{R} = A \cdot A^T$ . On the other hand, the NLF model can also extract two different LF matrices, i.e.,  $P$  and  $Q$  as given in [Luo et al., 2014] from a SHiDS matrix, but the resulting low-rank matrix is asymmetric.

As a matter of fact, NLF and SNLF models are strongly connected with each other: with proper initializations, NLF can achieve similar outputs with those by SNLF. By making  $P=Q$  before a specified training iteration in NLF, we have:

$$P = Q \Rightarrow \arg \min_{P,Q} \varepsilon(P,Q) = \arg \min_P \varepsilon(P). \quad (6)$$

Analogously, we have

$$Q = P \Rightarrow \arg \min_{P,Q} \varepsilon(P,Q) = \arg \min_Q \varepsilon(Q). \quad (7)$$

Note that (6) and (7) are equivalent. Hence, we assert that if we have  $P=Q$  before a specified training iteration in NLF, then we keep  $P=Q$  after this iteration. Given that NLF is an iterative model, we conclude that if we have:

$$\forall i \in N, k \in \{1, 2, \dots, d\}: p_{i,k}^0 \geq 0, q_{i,k}^0 \geq 0, p_{i,k}^0 = q_{i,k}^0, \quad (8)$$

then  $P$  and  $Q$  remain equal and non-negative with the single LF dependent multiplicative update [Luo et al., 2014]. With (6)~(8), we further have:

$$\begin{aligned} \begin{cases} \hat{r}_{i,j} = \sum_{k=1}^d p_{i,k} q_{j,k}, \hat{r}_{j,i} = \sum_{k=1}^d q_{i,k} p_{j,k}, \\ \forall i, j \in N, k \in \{1, 2, \dots, d\}: p_{i,k} = q_{i,k}, p_{j,k} = q_{j,k}; \end{cases} \\ \Rightarrow \hat{r}_{i,j} = \hat{r}_{j,i}. \end{aligned} \quad (9)$$

(9) indicates that NLF can also achieve a symmetric low-rank approximation to a SHiDS matrix by adopting the same initial hypothesis for  $P$  and  $Q$ , at the cost of lower computational efficiency than SNLF.

### 3.3 Biased SNLF Model

As discussed in prior works [Koren et al., 2009; Luo et al., 2016], to integrate linear biases into an LF/NLF model can further improve its performance. SNLF is compatible with linear biases. With them, the objective function (2) is extended into:

$$\begin{aligned} \arg \min_{A,B} \varepsilon(A,B) &= \frac{1}{2} \sum_{r_{i,j} \in \Lambda} \left( \left( r_{i,j} - b_i - b_j - \sum_{k=1}^d a_{i,k} a_{j,k} \right)^2 \right. \\ &\quad \left. + \lambda_B (b_i^2 + b_j^2) + \lambda_A \left( \sum_{k=1}^d a_{i,k}^2 + \sum_{k=1}^d a_{j,k}^2 \right) \right), \end{aligned} \quad (10)$$

$$s.t. \forall i, j \in N, k \in \{1, 2, \dots, d\}:$$

$$b_i \geq 0, b_j \geq 0, a_{i,k} \geq 0, a_{j,k} \geq 0;$$

where the length- $|N|$  vector  $B$  is the bias vector for  $N$ .

By applying AGD to (10) with respect to  $A$  and  $B$ , we obtain:

$$\begin{aligned} \arg \min_{A,B} \varepsilon(A,B) &\stackrel{AGD}{\Rightarrow} \forall i \in N, k \in \{1, 2, \dots, d\}: \\ b_i &\leftarrow b_i + \eta_i \sum_{j \in \Lambda(i)} \left( (r_{i,j} - \hat{r}_{i,j}) - \lambda_B b_i \right), \end{aligned} \quad (11)$$

$$a_{i,k} \leftarrow a_{i,k} + \eta_{i,k} \sum_{j \in \Lambda(i)} \left( a_{j,k} (r_{i,j} - \hat{r}_{i,j}) - \lambda_A a_{i,k} \right);$$

where  $\hat{r}_{i,j} = b_i + b_j + \sum_{k=1}^d a_{i,k} a_{j,k}$  denotes the approximation to  $r_{i,j}$ . Through manipulating  $\eta_i$  and  $\eta_{i,k}$ , we cancel the negative terms in (11) to obtain the following update rule:

$$\begin{aligned} \arg \min_{A,B} \varepsilon(A,B) &\stackrel{\text{Single LF-dependent Multiplicative Update}}{\Rightarrow} \forall i \in N, k \in \{1, 2, \dots, d\}: \\ b_i &\leftarrow b_i \frac{\sum_{j \in \Lambda(i)} r_{i,j}}{\sum_{j \in \Lambda(i)} (\hat{r}_{i,j} + \lambda_B b_i)}, a_{i,k} \leftarrow a_{i,k} \frac{\sum_{j \in \Lambda(i)} a_{j,k} r_{i,j}}{\sum_{j \in \Lambda(i)} (a_{j,k} \hat{r}_{i,j} + \lambda_A a_{i,k})}. \end{aligned} \quad (12)$$

With (12), we obtain the biases SNLF (BSNLF) model for SHiDS matrices.

## 4 Experimental Results and Analysis

**Compared Models.** The experiments are conducted with respect to six models, whose details are given in Table I.

**Evaluation Protocol.** For industrial applications [Koren et al., 2009; Luo et al., 2014; Szklarczyk et al., 2015], to estimate the missing data of a SHiDS matrix is highly important, due to the great desire to recover the full relationship among involved entities. Hence, we adopt the task of missing data estimation as the first evaluation protocol to validate involved models' performance.

**Evaluation Metrics.** For a tested model, its accuracy in missing data estimation is usually measured by root mean squared error (RMSE) [Koren et al., 2009; Luo et al., 2014]:

$$RMSE = \sqrt{\left( \sum_{r_{u,v} \in \Gamma} (r_{u,v} - \hat{r}_{u,v})^2 \right) / |\Gamma|}, \quad (13)$$

where  $\Gamma$  denotes the validation set and is naturally disjoint with the training set  $\Lambda$ , and  $\hat{r}_{u,v}$  denotes the prediction for the testing instance  $r_{u,v} \in \Gamma$ . Naturally, low RMSE stands for high prediction accuracy for missing data in  $\Gamma$ .

Meanwhile, we are concerned with the computational efficiency of tested models. So we have recorded the consumed time to make a model converge. All experiments are conducted on a tablet with a 3.4 GHz i7 CPU and 16 GB RAM. The programming language is JAVA SE 7U60.

**Datasets.** The experiments are conducted on four datasets, whose details are given in Table II. All datasets are SHiDS matrices from real applications. D1 and D2 are from the STRING database [Szklarczyk et al., 2015], which contains protein interactome weights in various organs. D3 records the temperature data of a steel cylinder, where the surface nodes form an undirected network. D4 records the vibration stiffness data of a piece of a special kind of material, where the material kernels form an undirected network. Both D3

and D4 are from the University of Florida sparse matrix collection [Davis and Hu, 2011]. In general, D1 and D2 are related to bioinformatics, while D3 and D4 are related to more conventionally industrial applications.

On all datasets we adopt the 80%-20% train-test settings and five-fold cross-validations: a) each time we select four subsets to train a model, predicting the remaining one subset, and b) we sequentially repeat this process for five times.

**Model Settings.** Note that M6, i.e., the SNMF model, cannot address a SHiDS matrix directly [Wang et al., 2008; Huang et al., 2014]. Hence, for M6, the missing data of each dataset is pre-filled with the average of  $\Lambda$ . As discussed in [Luo et al., 2014, Luo et al., 2016], the regularization coefficients affect the performance of NLF models. For a fair comparison, on each dataset we tune the regularization coefficients of each tested model on one fold, and adopt the same value on the remaining four folds. Meanwhile, since an NLF model's performance further relies on the LF dimension  $d$ , we have tested  $d$  in the [5, 80] interval. The training process of each tested model terminates if a) the number of consumed iterations reaches a preset threshold, i.e., 1,000, and b) the model converges, i.e., the difference in the training error of two consecutive iterations is smaller than  $10^{-5}$ .

### 4.1 Validation of Model Symmetry

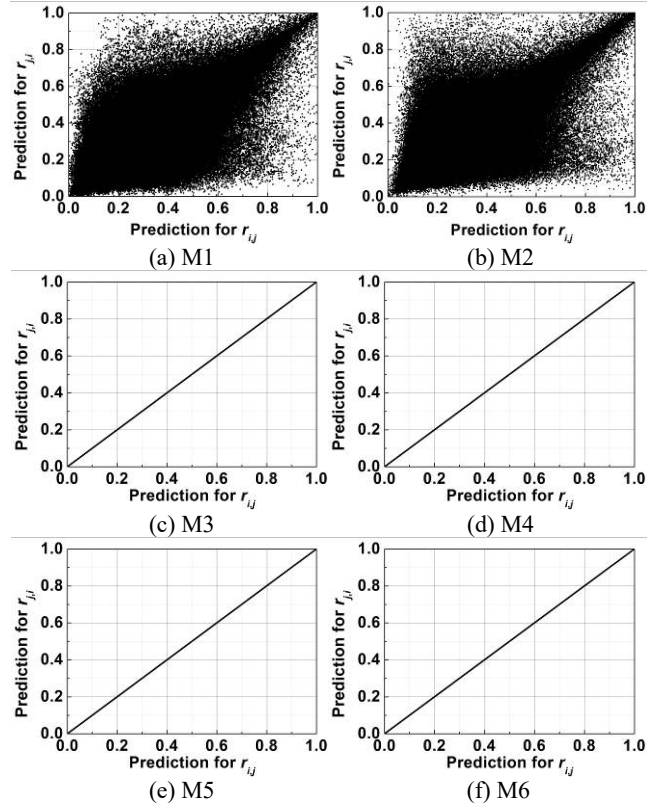


Figure 1. Data distributions in the low-rank approximation to D1 by all models with  $d=20$ , where  $x$  and  $y$  axes respectively denote  $\hat{r}_{i,j}$  and  $\hat{r}_{j,i}$ . If the resulting approximation is symmetric, then the data concentrate along  $x=y$ .

Table 1. Compared Models in the Experiments

No.	Name	Description
M1	NLF	Asymmetric NLF [Luo et al., 2014].
M2	BNLF	Asymmetric and biased NLF [Luo et al., 2016].
M3	NLF with symmetry	The NLF model whose LF matrices are assigned with the same initial hypothesis as in Section 3.2.
M4	SNLF	Symmetric NLF presented in Section 3.1.
M5	BSNLF	Biased SNLF presented in Section 3.3.
M6	SNMF	Symmetric NMF described in [Wang et al., 2008].

Table 2. Adopted Networks in the Experiments

No.	Description	$ \Lambda $	$ \mathcal{N} $	Density
D1	Protein interactome in <i>Pseudomonas Aeruginosa</i> PAO1.	1,248,802	4,260	6.88%
D2	Protein interactome in <i>Arabidopsis Thaliana</i> .	10,637,352	24,283	1.80%
D3	Temperature data of a steel cylinder.	1,751,178	24,696	0.28%
D4	Vibration stiffness data of a piece of special material.	711,558	102,158	0.007%

First of all, it is important to check that whether a tested model can correctly describe the symmetry of a SHiDS matrix. Figure 1 depicts the data distribution in the low-rank approximation to D1 by each tested model. Note that similar situations can be found on D2-D4. From Figures 1(a) and 1(b), we clearly see that without the symmetric design, M1 and M2 cannot correctly describe the symmetry of the target matrix.

On the other hand, as depicted in Figures 1(d) and (e), both M4 and M5, i.e., SNLF and BSNLF, correctly grasp the symmetry of the target network. Data in the low-rank approximations achieved by both models distribute along the line  $x=y$  strictly. Meanwhile, as shown in Figure 1(c), we see that M3 can also achieve symmetric low-rank approximations to the target SHiDS matrix. This phenomenon verifies the inferences in Section 3.2. Meanwhile, M6, i.e., the SNMF model, can also describe the symmetry of the target SHiDS matrix.

### 4.2 Prediction Accuracy for Missing Data

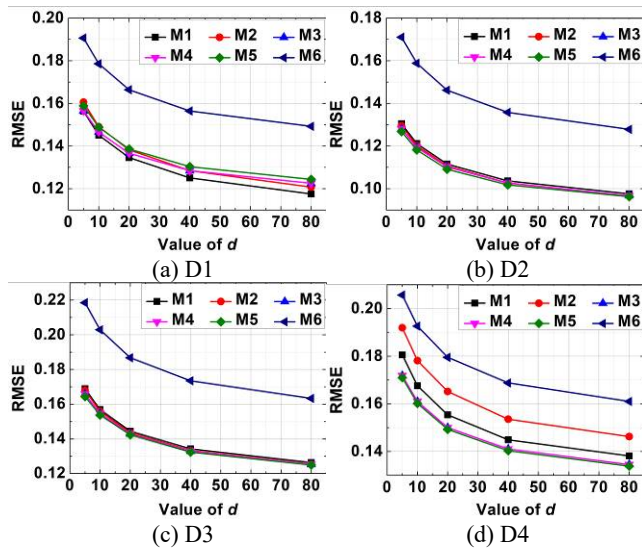


Figure 2. RMSE of tested models as  $d$  increases from 5 to 80.

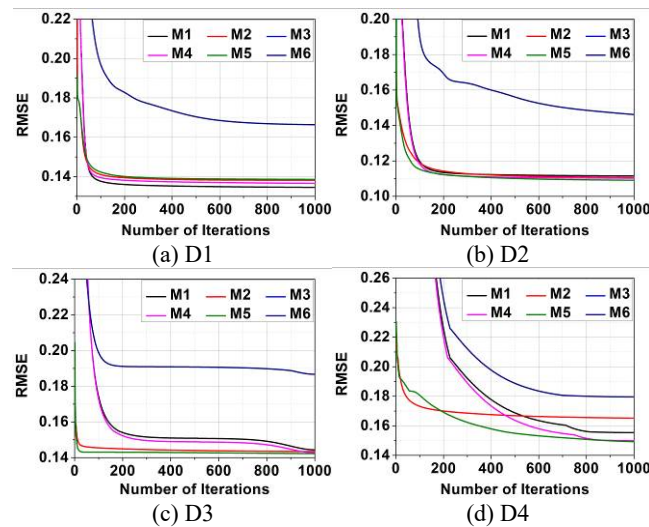


Figure 3. Training process of involved models with  $d=20$ .

Figure 2 depicts the RMSE of M1-M6 as  $d$  increases from 5 to 80. Figure 3 depicts the training process of M1-M6 with  $d=20$ . From these results, we have the following findings:

a) M3 and M4 generate the same predictions for missing data from a SHiDS matrix. As shown in Figure 2, as  $d$  increases, the accuracy curves of M3 and M4 overlap on all datasets. Moreover, as depicted in Figure 3, the training curves of M3 and M4 also overlap. These phenomena again verify Section 3.2: with  $P$  and  $Q$  initially identical, an NLF model achieves the same outputs as an SNLF model does.

b) Effects in prediction accuracy by the symmetric designs in SNLF models are data dependent. On D1, both M4 and M5 are outperformed by the corresponding asymmetric models, i.e., M1 and M2. As depicted in Figure 2(a), M4's RMSE is always higher than that of M1. Moreover, the accuracy gap increases as  $d$  increases. For instance, with  $d=5$ , the RMSE of M1 and M4 is 0.1564 and 0.1565, indicating the accuracy gap at 0.06%. With  $d=80$ , the RMSE of M1 and M4 is 0.1175 and 0.1225, where the accuracy gap increases to 4.08%. Similar situations can be found between M2 and M5 on D1.

However, the situations on D2-D4 are different. As depicted in Figures 2(b)-2(d), M4 and M5 outperform M1 and M2. For instance, with  $d=5$  on D2, the RMSE of M1, M2, M4 and M5 is 0.1305, 0.1294, 0.1280 and 0.1267, respectively. The asymmetric NLF model, i.e., M1, is outperformed by the SNLF model, i.e., M4, in prediction accuracy at 1.92%. The asymmetric and biased NLF model, i.e., M2, is outperformed by the BSNLF model, i.e., M5, in prediction accuracy at 2.09%. As  $d$  increases to 80, the RMSE of M1, M2, M4 and M5 is 0.0984, 0.0978, 0.0967 and 0.0962. The accuracy gap between M1 and M4 is 1.73%, between M2 and M5 is 1.64%, respectively. Similar situations also be found in the comparison on D3 and D4.

c) Effects in prediction accuracy by linear biases in SNLF models are also data dependent. On D1, M5 is outperformed by M4, as shown in Figures 2(a) and 3(a). This phenomenon indicates that the integration of linear biases into the SNLF model results in loss of prediction accuracy for missing data. Nonetheless, on D2-D4 the situations are different: M5 achieves lower RMSE than M4 does, as shown in Figures 2(b)-2(d) and 3(b)-3(d).

d) SNMF cannot generate accurate predictions for missing data in a SHiDS matrix as its peers do. From Figures 2 and 3, we see that the RMSE of SNMF is obviously higher than that of the other models. For instance, with  $d=5$  on D4, the RMSE of D6, i.e., SNMF, is 0.2057. Compared with 0.1806 by M1, 0.1919 by M2, 0.1719 by M4 and 0.1710 by M5, the accuracy gap is 12.20%, 6.71%, 16.43% and 16.87%, respectively. As  $d$  increases to 80, the RMSE of D6 is 0.1610. Compared with 0.1381 by M1, 0.1463 by M2, 0.1346 by M4 and 0.1339 by M5, the accuracy gap is 14.22%, 9.13%, 16.38% and 16.83%, respectively. Similar situations can be found on D1-D3.

### 4.3 Computational Efficiency

Figure 4 depicts the time costs of tested models as  $d$  increases from 5 to 80. It is observed that the computational efficiency of M4 and M5 is much higher than that of M1-M3 and M6: a)



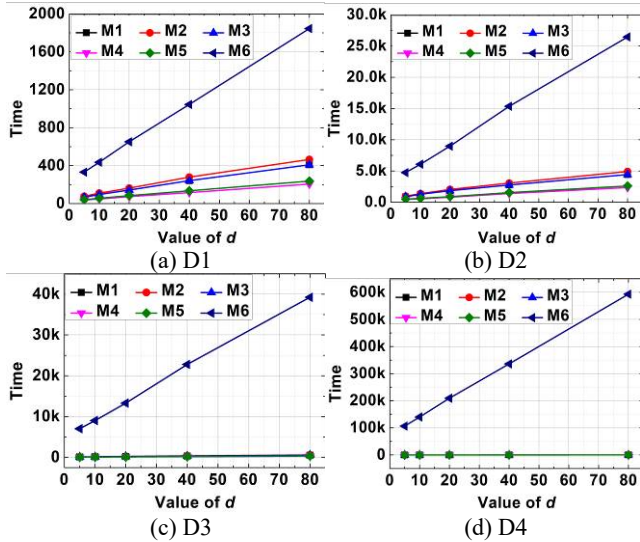


Figure 4. Training time of tested models as  $d$  increases from 5 to 80.

the time costs of M1 and M3 are approximately the same; b) M4’s time cost is about 1/2 that of M1 and M3’s time cost; c) M5’s time cost is about 1/2 that of M2’s time cost; d) M5’s time cost is slightly higher than that M4’s time cost; and e) the time cost of M6 is much higher than those of its peers.

From Section 3, we clearly see the reasons for the above phenomena:

a) M1 and M3 train two LF matrices, while M4 only trains one. Hence, time cost by M4 is about half of that by M1 and M3. Considering M2 and M5, the situation is similar;

b) Compared with M4, M5 trains linear biases in addition. This results in additional constant cost, making it consume a bit more time than M4 does;

c) M6, i.e., the SNMF model, manipulates the product between  $R$  and  $A$  where  $R$  is prefilled with average of  $\Lambda$ . To manipulate such a full approximation to a SHiDS matrix can be enormously time-consuming. For instance, D4 is a  $102,158 \times 102,158$  SHiDS matrix with 711,558 known entries. In other words, M6 has to manipulate a full matrix with more than 10 billion entries, while M1-M5 only focuses on the known entries which are less than one million. This phenomenon again indicates that it is computationally expensive to perform non-negative LF analysis on SHiDS matrices with SNMF models, which are actually designed for symmetric and full matrices.

#### 4.4 Summery

From the experimental results, we see that M4 and M5, i.e., the proposed SNLF and BSNLF, outperform their peers. They correctly grasp the symmetry of SHiDS networks with high computational efficiency, and their prediction accuracy for missing data is highly competent [Luo et al., 2014, Luo et al., 2016].

## 5 Discussions

As shown in Section 4, the efficiency of the proposed SNLF models is supported by the experimental results. In this

section, we further discuss some interesting issues regarding the characteristics and applications of the SNLF models.

#### a) Symmetry and prediction accuracy for missing data.

From Section 3.2, we see that SNLF models can correctly represent the symmetry of a SHiDS matrix, while asymmetric NLF models [Luo et al., 2014, Luo et al., 2016] cannot. From this point of view, SNLF models can better represent the natural characteristic of a SHiDS matrix than asymmetric NLF models do. However, from Section 4.2, we see that SNLF models might be outperformed by asymmetric NLF models in terms of prediction accuracy for missing data. This phenomenon is interesting, since it suggests that model symmetry is not necessarily connected with prediction accuracy.

One possible reason for this phenomenon is that without the symmetric design, i.e., with  $P$  and  $Q$  initially unequal in NLF, both  $P$  and  $Q$  actually represent different hypotheses for LFs corresponding to  $N$  hidden in  $\Lambda$ , and they achieve similar effect to that by an ensemble [Hui and Suganthan, 2016]. Thus, the resulting asymmetric NLF model may better grasp the numerical characteristics of the target SHiDS matrix than an SNLF model does, which only makes one unique hypothesis for the desired LF matrix.

Nevertheless, from Section 4.2 we also see that SNLF models can generate more accurate prediction for missing data on different SHiDS matrices than asymmetric NLF models do. This phenomenon indicates that the ensemble effects in an NLF model cannot guarantee the accuracy gain by the asymmetric NLF models over the SNLF models. An asymmetric NLF model cannot describe the symmetric of a SHiDS matrix, which also weakens its representativeness for the matrix structure. On different SHiDS matrices, this defect can cause accuracy loss of asymmetric NLF models in missing data estimation, thereby making them be outperformed by SNLF models.

Moreover, in this work we adopt the strict constraints to make  $\hat{R} = A \cdot A^T$ . As a matter of fact, such assumptions might be over strict since we can also decompose a symmetric matrix into multiple LF matrices without violating the symmetry of the resultant low-rank approximation to the target matrix. To do so, it is necessary to apply the symmetric constraints into the training process of the desired and non-negative LFs. On the other hand, it is also highly interesting to investigate the structure of SNLF models according to recent progress in symmetric and non-negative matrix factorization [Borhani et al., 2016, Paul and Chen, 2016]. These works are included in our future plan.

#### b) Industrial applications of SNLF models.

SHiDS matrices are often encountered in various industrial applications. For information systems, such data exist in social network services [Ghahramani, 2015; Yang et al., 2015], e-commerce systems [Adomavicius and Tuzhilin, 2005; Luo et al., 2015], bioinformatics applications [You et al., 2010; Hofree et al., 2013], internet analysis applications [Liao et al., 2013], etc. For conventionally industrial applications, such data exist in wireless sensor networks [Piao et al., 2014; Nguyen and Shin, 2016], material analysis

[Davis and Hu, 2011], mechanical fault detection [Gao et al., 2016], etc. SNLF models are able to analyze such data efficiently and accurately, seeking for various desired patterns like feature extraction [Davis and Hu, 2011; Gao et al., 2016], missing data estimation [Koren et al., 2009; Luo et al., 2014; Szklarczyk et al., 2015], cluster detection [He et al., 2011; Yang et al., 2015 ], etc.

## 6 Conclusions

This work aims at designing the symmetric and non-negative latent factor (SNLF) models for non-negative latent factor (LF) analysis on undirected, high dimensional and sparse networks. Since such a network corresponds to a symmetric, high dimensional and sparse (SHiDS) matrix, this task is equivalent to conduct non-negative LF analysis on SHiDS matrices. The key issue is how to describe the symmetry of the target matrix as well as maintain the model non-negativity. Based on careful investigation into the single LF dependent multiplicative update [Luo et al., 2014, Luo et al., 2016], this paper proposes to conduct this process on a unique LF matrix, thereby achieving the SNLF model. Moreover, through integrating linear biases into the SNLF model, the biased and symmetric non-negative latent factor (BSNLF) model is further developed. Experimental results on four SHiDS networks from real and industrial applications demonstrate that the proposed SNLF models are able to conduct the non-negative LF analysis on such networks symmetrically, accurately and efficiently.

Future works include a) investigating the industrial applications of the SNLF models; b) improving the model structure and constrained training process to further boost the performance of the SNLF models.

## Acknowledgments

This research is supported in part by the Pioneer Hundred Talents Program of Chinese Academy of Sciences, in part by the International Joint Project funded jointly by the Royal Society of the UK and the National Natural Science Foundation of China under Grant 61611130209, in part by the Young Scientist Foundation of Chongqing under Grant No. cstc2014kjrc-qnrc40005, in part by the National Natural Science Foundation of China under Grant 61370150, Grant 61433014, and Grant 61402198; and in part by the Fundamental Research Funds for the Central Universities under Grant No. 106112016CDJXY180005 and No. CDJZR12180012.

## References

- [Adomavicius and Tuzhilin, 2005] Gediminas Adomavicius and Alexander Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005.
- [Borhani et al., 2016] Reza Borhani, Jeremy Watt, and Aggelos Katsaggelos, "Fast and Effective Algorithms for Symmetric Nonnegative Matrix Factorization," arXiv:1609.05342, 2016.
- [Davis and Hu, 2011] Timothy Davis and Yifan Hu, "The University of Florida Sparse Matrix Collection," *ACM Transactions on Mathematical Software*, vol. 38, no. 1, pp. 1-25, 2011.
- [Gao et al., 2016] Bin Gao, Wai-Lok Woo, Guiyun Tian, and Hong Zhang, "Unsupervised Diagnostic and Monitoring of Defects Using Waveguide Imaging With Adaptive Sparse Representation," *IEEE Trans. on Industrial Informatics*, vol. 12, no. 1, pp. 405 - 416, 2016.
- [Ghahramani, 2015] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452-459, 2015.
- [He et al., 2011] Zhaoshui He, Shengli Xie, Rafal Zdunek, Guoxu Zhou, and Andrzej Cichocki, "Symmetric Nonnegative Matrix Factorization: Algorithms and Applications to Probabilistic Clustering," *IEEE Trans. on Neural Networks*, vol. 22, no. 12, pp. 2017-2131, Dec., 2011.
- [Hofree et al., 2013] Matan Hofree, John Shen, Hannah Carter, Andrew Gross, and Trey Ideker, "Network-based stratification of tumor mutations," *Nature Methods*, vol. 10, no. 11, pp. 1108-1115, 2013.
- [Huang et al., 2014] Kejun Huang, Nicholas Sidiropoulos, and Ananthram Swami, "Non-Negative Matrix Factorization Revisited: Uniqueness and Algorithm for Symmetric Decomposition," *IEEE Trans. on Signal Processing*, vol. 62, no. 1, pp. 211-224, Jan., 2014.
- [Hui and Suganthan, 2016] Sheldon Hui and Ponnuthurai Suganthan, "Ensemble and Arithmetic Recombination-Based Speciation Differential Evolution for Multimodal Optimization," *IEEE Trans. on Cybernetics*, vol. 46, no. 1, pp. 64-74, Jan., 2016.
- [Kim and Park, 2007] Hyunsoo Kim and Haesun Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495-1502, 2007.
- [Koren et al., 2009] Yehuda Koren, Robert Bell, and Chris Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE Computer*, vol. 42, no. 8, pp. 30-37, Aug., 2009.
- [Lee and Seung, 1999] Daniel Lee and Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788-791, 1999.
- [Liao et al., 2013] Yongjun Liao, Wei Du, Pierre Geurts, and Guy Leduc, "DMFSGD: A Decentralized Matrix Factorization Algorithm for Network Distance Prediction," *IEEE/ACM Trans. on Networking*, vol. 21, no. 5, pp. 1511-1524, May., 2013.
- [Luo et al., 2014] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu, "An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative

- Filtering for Recommender Systems,” *IEEE Trans. on Industrial Informatics*, vol. 10, no. 2, pp. 1273 - 1284, May., 2014.
- [Luo et al., 2015] Xin Luo, Mengchu Zhou, Shuai Li, Zhuhong You, Yunni Xia, Qingsheng Zhu, and Hareton Leung, “An Efficient Second-order Approach to Factorizing Sparse Matrices in Recommender Systems,” *IEEE Trans. on Industrial Informatics*, vol. 11, no. 4, pp. 946 - 956, Aug., 2015.
- [Luo et al., 2016] Xin Luo, Mengchu Zhou, Yunni Xia, Qingsheng Zhu, Ahmed Ammari, and Ahmed Alabdulwahab, “Generating Highly Accurate Predictions for Missing QoS Data via Aggregating Nonnegative Latent Factor Models ” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 524-537 Mar., 2016.
- [Nguyen and Shin, 2016] Thu Nguyen and Yoan Shin, “Matrix Completion Optimization for Localization in Wireless Sensor Networks for Intelligent IoT,” *Sensors*, vol. 16, no. 5, pp. 722-733, 2016.
- [Paul and Chen, 2016] Subhadeep Paul and Yuguo Chen, “Orthogonal symmetric non-negative matrix factorization under the stochastic block model,” arXiv:1605.05349, 2016.
- [Piao et al., 2014] Xinglin Piao, Yongli Hu, Yanfeng Sun, Baocai Yin, and Junbin Gao, “Correlated Spatio-Temporal Data Collection in Wireless Sensor Networks Based on Low Rank Matrix Approximation and Optimized Node Sampling,” *Sensors*, vol. 14, no. 12, pp. 23137-23158, 2014.
- [Szklarczyk et al., 2015] Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi Tsafou, Michael Kuhn, Peer Bork, Lars Jensen, and Christian von Mering, “STRING v10: protein–protein interaction networks, integrated over the tree of life,” *Nucleic Acids Research*, vol. 43, no. 1, pp. D447-D452, 2015.
- [Wang et al., 2008] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding, “Multi-Document Summarization via Sentence-Level Semantic Analysis and Symmetric Matrix Factorization,” in *Proc. of the 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Singapore, 2008, pp. 307-314.
- [Yang et al., 2015] Liang Yang, Xiaochun Cao, Di Jin, Xiao Wang, and Dan Meng, “A Unified Semi-Supervised Community Detection Framework Using Latent Space Graph Regularization,” *IEEE Trans. on Cybernetics*, vol. 45, no. 11, pp. 2585-2598, 2015.
- [You et al., 2010] Zhuhong You, Yingke Lei, Jie Gui, Deshuang Huang, and Xiaobo Zhou, “Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data,” *Bioinformatics*, vol. 26, no. 21, pp. 2744-2751, Nov, 2010.