An Instance-frequency-weighted Regularization Scheme for Non-negative Latent Factor Analysis on High Dimensional and Sparse Data

Xin Luo, *Senior Member*, *IEEE*, Zidong Wang, *Fellow*, *IEEE*, and Mingsheng Shang

***Abstract*—High dimensional and sparse (HiDS) data with non-negativity constraints are commonly seen in industrial applications like recommender systems. They can be modeled into an HiDS matrix, from which non-negative latent factor analysis (NLFA) is highly effective in extracting useful features. Preforming NLFA on an HiDS matrix is ill-posed, desiring an effective regularization scheme for avoiding overfitting. Current models mostly adopt a standard L_2 scheme, which does not consider the imbalanced distribution of known data in an HiDS matrix. From this point of view, this study proposes an instance-frequency-weighted regularization (IR) scheme for NLFA on HiDS data. It specifies the regularization effects on each LF with its relevant instance count, i.e., instance-frequency, which clearly describes the known data distribution of an HiDS matrix. By doing so, it achieves finely grained modeling of regularization effects. Experimental results on HiDS matrices from industrial applications demonstrate that compared with an L_2 scheme, an IR scheme enables a resultant model to achieve higher accuracy in missing data estimation of an HiDS matrix.**

***Index Terms*—Non-negative Latent Factor Analysis, Regularization, Instance-frequency, High Dimensional and Sparse Data, Recommender System, Industrial Application**

I. Introduction

IN THIS ERA of big data, learning system-based industrial applications usually involve numerous entities and their corresponding high dimensional relationships, e.g., users, items and user-item preferences in recommender systems [1-3], users, services and user-service QoS history in Web-service QoS analysis [4-6], users and user-user trust networks in social network-based services [7-10], and proteins and protein-protein interactome mappings in bioinformatics [11-13]. Due to the exploding numbers of involved entities, e.g., millions of users and items in a recommender system, it becomes impossible to observe the full relationship among them. As a result, High Dimensional and Sparse (HiDS) matrices are often adopted to describe such relationships in practice [1-13]. Note that in an HiDS matrix, most entries are "missing" rather than "zero" in conventional sparse matrices, e.g., an unobserved connection between a specified user-item tuple is "missing" rather than "zero" in recommender systems [1-3].

In spite of its sparsity, an HiDS matrix contains rich information regarding various desired patterns, e.g., user preferences in a recommender system [1-3] and community tendency in a social network service system [7-10]. How to extract such useful knowledge from an HiDS matrix becomes a vital yet thorny issue. Various knowledge acquisition models are proposed for such purposes [3, 8-10], among which latent factor analysis (LFA)-based models prove to be highly efficient [3, 14-18].

Similar to matrix factorization or low-rank embedding models [47-49], an LFA-based model maps entities corresponding to rows and columns of an HiDS matrix into a unique and low-dimensional feature space. Then it builds an objective function based on the target matrix's known entries and corresponding LFs. This objective is minimized with respect to desired LFs to form the output model [14-18]. Representative LFA-based models include the biased, regularized, incremental and simultaneous model [14], singular value decomposition plus-plus model [15, 16], probabilistic model [17], nonparametric model [18], weighted trace-norm regularization-based model [19], and non-parametric Bayesian-based model [20]. Different LFA-based models have different model design and training schemes, yet they share the same principle of focusing on known entries of an HiDS matrix to define the learning objective and train desired latent factors (LFs). These LFs are interpreted as the entity features hidden in an HiDS matrices and very useful in various data analysis tasks like missing data estimation [3, 14-18], community detection [9, 21], image tagging [22], video re-indexing [23], mobile-user tracking [24], and point of interests recommendation [50].

In spite of their efficiency, the aforementioned LFA-based models fail to fulfill the non-negativity constraints. Note that non-negative HiDS data are frequently encountered in

This research is supported in part by the National Natural Science Foundation of China under grants 61772493, 91646114 and 61602352, in part by the Chongqing Research Program of Technology Innovation and Application under grants cstc2017rgzn-zdyfX0020, cstc2017zdcy-zdyf0554 and cstc2017rgzn-zdyf0118, in part by the Chongqing Cultivation Program of Innovation and Entrepreneurship Demonstration Group under grant cstc2017kjrc-cxcytd0149, in part by the Chongqing Overseas Scholars Innovation Program under Grant cx2017012 and cx2018011, and in part by the Pioneer Hundred Talents Program of Chinese Academy of Sciences. Z. Wang and M. Shang are the corresponding authors.

X. Luo is with the School of Computer Science and Technology, Dongguan University of Technology, Dongguan, Guangdong 523808, China (e-mail: luoxin21@gmail.com).

Z. Wang is with the Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex UB8 3PH, U.K. (e-mail: Zidong.Wang@brunel.ac.uk)

M. Shang is with the Chongqing Engineering Research Center of Big Data Application for Smart Cities, and Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China (e-mail: msshang@cigit.ac.cn)

industrial applications like Web-service QoS analysis [4-6] and social-network services [7-10]. Naturally, non-negative LFs well depict the essential characteristic, i.e., non-negativity, of such non-negative data. Hence, it is highly significant to build non-negative latent factor analysis (NLFA)-based models on them. Given a full matrix, non-negative matrix factorization (NMF) models are well established [25-32]. Paatero and Tapper [25] propose to truncate the negative factors to zero in an alternating least squares (ALS)-based training process, thereby achieving non-negative LF matrices. Lee and Seung [26] derive the non-negatively multiplicative update to keep the non-negativity of the LF matrices if they are initially non-negative. Lin [27] proposes projected gradient decent to train the desired LFs with gradient descent and truncate the negative results to zero to implement an NMF model. Hoyer [28] proposes a sparse NMF model with the $L_{2,1}$ regularization adopting the non-negative and multiplicative update proposed in [25] for precisely clustering the involved entities. Kim and Park propose a sparse NMF model which also incorporates the $L_{2,1}$ regularization into the model but adopts the projected ALS proposed in [24] to train the desired LFs. Ding *et al.* [30] propose the kernel NMF model, which is especially effective in extracting meaningful patterns from images. These NMF models and their extensions [51-54] have proven to be highly effective on complete matrices. However, they cannot handle an HiDS matrix directly.

Great efforts haves been made for performing NLFA on an HiDS matrix. Existing models can be divided into two categories. The first kind of models adopt an intermediate full matrix to approximate an HiDS matrix, and then conduct the NMF process on this intermediate matrix to obtain non-negative LFs [33, 34], e.g., the weighted non-negative matrix factorization model by Zhang *et al*. [33] and the non-negative matrix completion model by Xu *et al*. [34]. Such a model can address HiDS matrices and is compatible with existing NMF algorithms [25-32], but suffers unacceptably high computational and storage costs. This is because an HiDS matrix's full approximation can be huge [3, 14-18]. For instance, the MovieLens 20M matrix [43] has 20,000,263 instances scattering in 138,493 rows and 26,744 columns. Its data density is 0.54% only, but the total number of its entries is more than 3.7 billion. To manipulate such a huge matrix is extremely expensive in practice if not impossible.

The second kind of models [35-39] design single-element-dependent and non-negative training schemes, which enable them to extract desired non-negative LFs from an HiDS matrix based on its known entries only. Luo *et al*. [35, 36] propose the single latent factor-dependent, non-negative and multiplicative update (SLF-NMU) learning scheme, which can guarantee the non-negativity of involved LFs if they are initially non-negative. They further integrate the principle of an alternating direction method into the training scheme, thereby achieving fast model convergence [37]. Although both models can perform efficient NLFA on an HiDS matrix, they mainly focus on the training scheme but ignoring the regularization design [33-39]. Performing NLFA on an HiDS matrix is in nature an ill-posed problem without a unique or globally-optimal solution. Due to the imbalanced data distribution in an HiDS matrix, the learning objective is also highly imbalanced, i.e., it depends heavily on LFs related to many instances and vice versa. When addressing such a problem, model regularization is vital in improving its generality [3, 14-18]. However, current NLFA-based models all adopt general regularization schemes like an L_2 scheme [14-18, 35-39]. It is designed for problems defined on full data, but HiDS data are far different in data sparsity and distribution. Note that characteristics of target data should be carefully modeled in a regularization scheme for making the resultant model well represent them. For addressing this issue, this study proposes an instance-frequency-weighted regularization (IR) scheme, which is especially designed for NFLA-based models defined on an HiDS matrix.

An IR scheme incorporates the information of imbalanced instance-frequency into the regularization terms, thereby implementing finely grained modeling of regularization effects. With it, this study aims at improving a resultant model's representativeness to HiDS data, and finally improving its performance like prediction accuracy for its missing data. The main contributions of this study include:

a) Instance-frequency-weighted regularization scheme, which is a novel regularization scheme especially designed for NLFA-based models on HiDS data;

b) Detailed algorithm design and analysis for a Non-negative latent factor analysis with Instance-frequency-weighted Regularization (NIR) model; and

c) Empirical studies on two HiDS matrices generated by industrial applications.

The rest of this paper is organized as follows: Section II states an NLFA problem, Section III presents the IR scheme, Section IV gives and discusses the experimental results, and finally, Section V concludes this paper.

II. An NLFA Problem

In our context, an HiDS matrix is defined as [14-18, 35-39]:

***Definition* 1.** Let M and N denote two large entity sets, $T^{|M|\times|N|}$ denote a matrix whose entry $t_{m,n}$ quantifies some relationship between entities $m\in M$ and $n\in N$, and Λ and Γ denote the known and unknown entry sets of T; T is HiDS if $|\Lambda|\ll|\Gamma|$.

Given T, an LFA-based model is defined as [14-18]:

***Definition* 2.** Given T and Λ, an LFA-based model seeks for a rank-d approximation $\hat{T}=PQ^{\mathrm{T}}$ to T with $P^{|M|\times d}$ and $Q^{|N|\times d}$ being LF matrices and $d\ll\min\{|M|,|N|\}$.

Note that d is interpreted as the dimension of the LF space, P and Q consist of LFs reflecting the characteristics of M and N described by Λ, respectively. To obtain them, an objective measuring the difference between Λ and corresponding entries in $\hat{T}$ is required. With Euclidean distance [14-18, 35-39], such an objective function is formulated by:

$$\varepsilon(P,Q)=\frac{1}{2}\sum_{t_{m,n}\in\Lambda}\left(t_{m,n}-\sum_{k=1}^{d}p_{m,k}q_{n,k}\right)^2 \tag{1}$$

where $t_{m,n}$, $p_{m,k}$ and $q_{n,k}$ denote specified entries in T, P, and Q, respectively. Most industrial data are non-negative [25-34], i.e., for $\forall m\in M$, $n\in N$: $t_{m,n}\geq 0$. For correctly describing such

non-negative data, the objective (1) is extended into the following constrained form [35-39]:

$$\varepsilon(P,Q)=\frac{1}{2}\sum_{t_{m,n}\in\Lambda}\left(t_{m,n}-\sum_{k=1}^{d}p_{m,k}q_{n,k}\right)^2,$$
$$s.t.\ \ \forall m\in M,n\in N,k\in\{1,\ 2,\ ...,\ d\}:\ p_{m,k}\geq 0,\ q_{n,k}\geq 0; \tag{2}$$

which establishes the problem of NLFA on an HiDS matrix.

III. NLFA with Instance-Frequency-Weighted Regularization

A. L_2 Norm-regularized Problem

An NLFA problem is ill-posed: a unique and globally-optimal solution cannot be achieved, and the results depend heavily on its initial hypotheses [14-18, 35-39]. It desires regularization for preventing overfitting [3, 14-18]. Existing models commonly adopt general regularization schemes, where an L_2 scheme is frequently encountered [14-18, 35-39]. Note that with an L_2 scheme, the regularization term corresponding to (2) is formulated as follows:

$$L=\frac{\lambda}{2}\left(\|P\|_F^2+\|Q\|_F^2\right) \tag{3}$$

where $\|\cdot\|_F$ computes the Frobenius norm of the enclosed matrix, and λ denotes the regularization coefficient. With it, (2) is extended into:

$$\varepsilon(P,Q)=\frac{1}{2}\sum_{t_{m,n}\in\Lambda}\left(t_{m,n}-\sum_{k=1}^{d}p_{m,k}q_{n,k}\right)^2+L$$
$$=\frac{1}{2}\sum_{t_{m,n}\in\Lambda}\left(t_{m,n}-\sum_{k=1}^{d}p_{m,k}q_{n,k}\right)^2+\frac{\lambda}{2}\left(\sum_{m=1}^{|M|}\sum_{k=1}^{d}p_{m,k}^2+\sum_{n=1}^{|N|}\sum_{k=1}^{d}q_{n,k}^2\right),$$
$$s.t.\ \ \forall m\in M,n\in N,k\in\{1,\ 2,\ ...,\ d\}:\ p_{m,k}\geq 0,\ q_{n,k}\geq 0. \tag{4}$$

Note that (4) works by minimizing the generalized loss and the Frobenius norms of P and Q simultaneously, thereby avoiding overfitting [14-18, 35-39].

Regularization effects in (4) are controlled by λ solely. As discussed in prior studies [25-30, 33, 34, 40, 41], when the cost function is defined on a rectangular matrix with imbalanced row/column ratio, the regularization coefficients can be specified according to its row and column counts for controlling the regularization effects more precisely. Formally, we assign different regularization coefficients λ_P and λ_Q to LFs in P and Q, respectively. They are set as [25-30, 33, 34, 40, 41]:

$$\lambda_P/|N|=\lambda_Q/|M|=\gamma\Rightarrow\lambda_P=\gamma|N|,\lambda_Q=\gamma|M|, \tag{5}$$

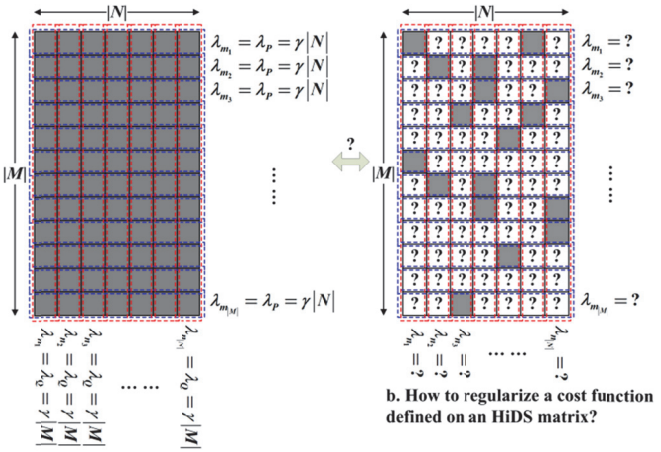
where the constant γ adjusts λ_P and λ_Q according to T's row and column counts. By combining (4-5) we achieve the following objective function:

$$\varepsilon(P,Q)=\frac{1}{2}\sum_{t_{m,n}\in\Lambda}\left(t_{m,n}-\sum_{k=1}^{d}p_{m,k}q_{n,k}\right)^2$$
$$+\frac{\gamma}{2}|N|\sum_{m=1}^{|M|}\sum_{k=1}^{d}p_{m,k}^2+\frac{\gamma}{2}|M|\sum_{n=1}^{|N|}\sum_{k=1}^{d}q_{n,k}^2, \tag{6}$$
$$s.t.\ \ \forall m\in M,n\in N,k\in\{1,\ 2,\ ...,\ d\}:\ p_{m,k}\geq 0,\ q_{n,k}\geq 0;$$

where the regularization term is actually given by:

$$L=\frac{\gamma}{2}\left(|N|\|P\|_F^2+|M|\|Q\|_F^2\right). \tag{7}$$

From (7) we clearly see that when $|N|<|M|$, heavier regularization effects are applied to LFs in Q rather than in P, and vice versa. In other words, an LF connected with many training instances suffer heavy regularization for preventing the entire model from overfitting, as in Fig. 1(a).



a. L_2 regularization for a cost function defined on a full matrix.

b. How to regularize a cost function defined on an HiDS matrix?

Fig. 1. Regularization design on a full matrix and an HiDS one.

B. An Instance-frequency-weighted Regularization Scheme

According to (7), an L_2 scheme considers the imbalanced row/column count in a full matrix. However, it cannot address the imbalanced data distribution of an HiDS matrix. As shown in Fig. 1(b), $\forall m\in M$ and $\forall n\in N$ correspond to specified entry counts in an HiDS matrix, which is actually the size of Λ's subsets related to them. Hence, regularization coefficients should be controlled with care by assigning a unique regularization coefficient to them, i.e., $\forall m\in M$ and $\forall n\in N$ we make λ_m and λ_n be their specified regularization coefficients. Thus, (4) is reformulated as follows:

$$\varepsilon(P,Q)=\frac{1}{2}\sum_{t_{m,n}\in\Lambda}\left(t_{m,n}-\sum_{k=1}^{d}p_{m,k}q_{n,k}\right)^2$$
$$+\frac{1}{2}\sum_{m=1}^{|M|}\lambda_m\sum_{k=1}^{d}p_{m,k}^2+\frac{1}{2}\sum_{n=1}^{|N|}\lambda_n\sum_{k=1}^{d}q_{n,k}^2 \tag{8}$$
$$s.t.\ \ \forall m\in M,n\in N,k\in\{1,\ 2,\ ...,\ d\}:\ p_{m,k}\geq 0,\ q_{n,k}\geq 0.$$

With (8), $\forall m\in M$ and $\forall n\in N$ we need to select λ_m and λ_n. When T is full as shown in Fig. 1(a), we actually have:

$$m_1,\ldots,m_{|M|}\in M:\left|\Lambda(m_1)\right|=\cdots=\left|\Lambda\left(m_{|M|}\right)\right|=|N|,$$
$$n_1,\ldots,n_{|N|}\in N:\left|\Lambda(n_1)\right|=\cdots=\left|\Lambda\left(n_{|N|}\right)\right|=|M|; \tag{9}$$

With (9), we rewrite the selection rule (5) as follows:

$$\lambda_{m_1}/|N|=\lambda_{m_1}/\left|\Lambda(m_1)\right|=\cdots=\lambda_{m_{|M|}}/|N|$$
$$=\lambda_{m_{|M|}}/\left|\Lambda\left(m_{|M|}\right)\right|=\lambda_P/|N|$$
$$=\lambda_{n_1}/|M|=\lambda_{n_1}/\left|\Lambda(n_1)\right|=\cdots=\lambda_{n_{|N|}}/|M| \tag{10}$$
$$=\lambda_{n_{|N|}}/\left|\Lambda\left(n_{|N|}\right)\right|=\lambda_Q/|M|=\gamma.$$

When T is HiDS, we naturally have $\forall m_1, m_2\in M$: $\Lambda(m_1)\leqslant|M|$,

$\Lambda(m_2)\leqslant|N|$ and we obviously cannot guarantee that $|\Lambda(m_1)|$ is equal to $|\Lambda(m_2)|$ (this condition is also applied to entities in N) as in (10). However, we can keep the ratio between an entity's related entry count and its regularization coefficient consistent in M and N, thereby achieving:

$$\begin{aligned}&\forall m\in M,\forall n\in N:\lambda_m\big/\left|\Lambda(m)\right|=\lambda_n\big/\left|\Lambda(n)\right|=\gamma\\&\Rightarrow\forall m\in M:\lambda_m=\gamma\left|\Lambda(m)\right|,\forall n\in N:\lambda_n=\gamma\left|\Lambda(n)\right|,\end{aligned}\tag{11}$$

where we call $|\Lambda(m)|$ and $|\Lambda(n)|$ instance-frequency weights related to m and n. In (11) we adopt the linear function $f(x)=x$ with respect to the instance-frequency weight related with each entity, i.e., we actually have:

$$\begin{aligned}&\forall m\in M:\lambda_m=\gamma\cdot f\left(\left|\Lambda(m)\right|\right),\forall n\in N:\lambda_n=\gamma\cdot f\left(\left|\Lambda(n)\right|\right);\\&\forall x\in\mathbb{R}^+:f(x)=x.\end{aligned}\tag{12}$$

Hence, its more generalized form is given by:

$$\forall m\in M:\lambda_m=\gamma\cdot f\left(\left|\Lambda(m)\right|\right),\forall n\in N:\lambda_n=\gamma\cdot f\left(\left|\Lambda(n)\right|\right),\tag{13}$$

where the function f is chosen to be monotonically increasing at its decision parameter for applying heavy regularization to LFs corresponding to high instance-frequency [25-30, 33, 34, 40, 41]. In this study, we make f be the power function, i.e., we let $f(x)=x^\beta$ where β is a positive constant, reformulating (13) into:

$$\forall m\in M:\lambda_m=\gamma\left|\Lambda(m)\right|^\beta,\forall n\in N:\lambda_n=\gamma\left|\Lambda(n)\right|^\beta.\tag{14}$$

Note that with such design, the regularization terms act as: a) increase slower than the instance-frequency weight when $0<\beta<1$, b) increase faster than the instance-frequency weight when $\beta>1$, and c) is linearly related to the instance-frequency weight when $\beta=1$.

By combining (8) and (14), we achieve the objective function for an NLFA-based model with an instance-frequency-weighted regularization (IR) scheme:

$$\begin{aligned}&\varepsilon(P,Q)=\frac{1}{2}\sum_{t_{m,n}\in\Lambda}\left(t_{m,n}-\sum_{k=1}^{d}p_{m,k}q_{n,k}\right)^2\\&+\frac{\gamma}{2}\left(\sum_{m=1}^{|M|}\left|\Lambda(m)\right|^\beta\sum_{k=1}^{d}p_{m,k}^2+\sum_{n=1}^{|N|}\left|\Lambda(n)\right|^\beta\sum_{k=1}^{d}q_{n,k}^2\right),\\&s.t.\ \forall m\in M,n\in N,k\in\{1,\ 2,\ ...,\ d\}:p_{m,k}\geq 0,\ q_{n,k}\geq 0.\end{aligned}\tag{15}$$

Note that the regularization term of (15) is formulated by:

$$L(P,Q)=\frac{\gamma}{2}\left(\sum_{m=1}^{|M|}\left|\Lambda(m)\right|^\beta\left\|P_{m,\cdot}\right\|_2^2+\sum_{n=1}^{|N|}\left|\Lambda(n)\right|^\beta\left\|Q_{n,\cdot}\right\|_2^2\right),\tag{16}$$

where $\|\cdot\|_2$ computes the L_2 norm of the enclosed vector, $P_{m,\cdot}$ and $Q_{n,\cdot}$ denote the mth row vector in P and nth row vector in Q, respectively. By comparing L_2 regularization terms in (7) with an IR terms in (16), we clearly see the differences between them: as shown in (7), an L_2 scheme only describes a small part of the information regarding the data distribution of a target HiDS matrix, which is reflected by its row/column ratio. In comparison, an IR scheme describes this information much more precisely by incorporating each entity's instance-frequency weight into the regularization coefficients. Moreover, it considers a tunable function with the instance-frequency weight, i.e., $f(x)=x^\beta$, to implement finely-grained control of the regularization effects. With it, we are able to precisely adjust the regularization on each desired LF during the training process, which is actually decided by the imbalanced data distribution of a HiDS matrix.

For solving the non-negativity-constrained problem (15), we adopt the single latent factor-dependent, non-negative and multiplicative update (SLF-NMU) scheme [36, 37]. Firstly, we apply the additive gradient descent to (13), yielding:

$$\begin{aligned}&\arg\min_{P,Q}\varepsilon(P,Q)\overset{AGD}{\Rightarrow}\forall m\in M,n\in N,k\in\{1,\ 2,\ ...,\ d\}:\\&\Rightarrow\begin{cases}p_{m,k}^c\leftarrow p_{m,k}^{c-1}-\eta_{m,k}\left(-\sum_{n\in\Lambda(m)}q_{n,k}^{c-1}\left(t_{m,n}-\hat{t}_{m,n}^{c-1}\right)+\gamma\left|\Lambda(m)\right|^\beta p_{m,k}^{c-1}\right),\\q_{n,k}^c\leftarrow q_{n,k}^{c-1}-\eta_{n,k}\left(-\sum_{m\in\Lambda(n)}p_{m,k}^{c-1}\left(t_{m,n}-\hat{t}_{m,n}^{c-1}\right)+\gamma\left|\Lambda(n)\right|^\beta q_{n,k}^{c-1}\right);\end{cases}\\&\Rightarrow\begin{cases}p_{m,k}^c\leftarrow p_{m,k}^{c-1}+\eta_{m,k}\sum_{n\in\Lambda(m)}q_{n,k}^{c-1}t_{m,n}-\eta_{m,k}\left(\sum_{n\in\Lambda(m)}q_{n,k}^{c-1}\hat{t}_{m,n}^{c-1}+\gamma\left|\Lambda(m)\right|^\beta p_{m,k}^{c-1}\right),\\q_{n,k}^c\leftarrow q_{n,k}^{c-1}+\eta_{n,k}\sum_{m\in\Lambda(n)}p_{m,k}^{c-1}t_{m,n}-\eta_{n,k}\left(\sum_{m\in\Lambda(n)}p_{m,k}^{c-1}\hat{t}_{m,n}^{c-1}+\gamma\left|\Lambda(n)\right|^\beta q_{n,k}^{c-1}\right);\end{cases}\end{aligned}\tag{17}$$

where we adopt c-1 and c for the status of LFs in P and Q after the cth and (c-1)th training iterations, and $\hat{t}_{m,n}^{c-1}=\sum_{k=1}^{d}p_{m,k}^{c-1}q_{n,k}^{c-1}$ for the status of entries in $\hat{T}$ relying on P^{c-1} and Q^{c-1}, respectively.

In update rule (17) $-\eta_{m,k}\left(\sum_{n\in\Lambda(m)}q_{n,k}^{c-1}\hat{t}_{m,n}^{c-1}+\gamma\left|\Lambda(m)\right|^\beta p_{m,k}^{c-1}\right)$ and $-\eta_{n,k}\left(\sum_{m\in\Lambda(n)}p_{m,k}^{c-1}\hat{t}_{m,n}^{c-1}+\gamma\left|\Lambda(n)\right|^\beta q_{n,k}^{c-1}\right)$ are the negative terms. Following the principle of SLF-NMU, we manipulate $\eta_{m,k}$ and $\eta_{n,k}$ to cancel them for achieving a non-negative training process. More specifically, we set them as

$$\begin{cases}\eta_{m,k}=p_{m,k}^{c-1}\Big/\left(\sum_{n\in\Lambda(m)}q_{n,k}^{c-1}\hat{t}_{m,n}^{c-1}+\gamma\left|\Lambda(m)\right|^\beta p_{m,k}^{c-1}\right)\\\eta_{n,k}=q_{n,k}^{c-1}\Big/\left(\sum_{m\in\Lambda(n)}p_{m,k}^{c-1}\hat{t}_{m,n}^{c-1}+\gamma\left|\Lambda(n)\right|^\beta q_{n,k}^{c-1}\right)\end{cases}\tag{18}$$

to achieve the following parameter update rule:

$$\begin{aligned}&\arg\min_{P,Q}\varepsilon(P,Q)\overset{SLF-NMU}{\Rightarrow}\ \forall m\in M,n\in N,k\in\{1,\ 2,\ ...,\ d\}:\\&\begin{cases}p_{m,k}^c\leftarrow p_{m,k}^{c-1}\dfrac{\sum_{n\in\Lambda(m)}q_{n,k}^{c-1}t_{m,n}}{\sum_{n\in\Lambda(m)}q_{n,k}^{c-1}\hat{t}_{m,n}^{c-1}+\gamma\left|\Lambda(m)\right|^\beta p_{m,k}^{c-1}},\\q_{n,k}^c\leftarrow q_{n,k}^{c-1}\dfrac{\sum_{m\in\Lambda(n)}p_{m,k}^{c-1}t_{m,n}}{\sum_{m\in\Lambda(n)}p_{m,k}^{c-1}\hat{t}_{m,n}^{c-1}+\gamma\left|\Lambda(n)\right|^\beta q_{n,k}^{c-1}}.\end{cases}\end{aligned}\tag{19}$$

With (19), we achieve the learning scheme of a Non-negative latent factor analysis with Instance-frequency-weighted Regularization (NIR) model. Next we present its algorithm design and analysis.

C. Algorithm Design and Analysis

Based on the inferences above, we deign the algorithm for an NIR model, as shown in Algorithm NIR. In this algorithm, we adopt four auxiliary matrices, i.e., $A^{|M|\times d}$, $B^{|M|\times d}$ $X^{|N|\times d}$ and $Y^{|N|\times d}$

for a highly efficient training process. Note that A and B are designed for caching the training increment brought by each instance on the numerator and denominator of the update rule for each LF in P, and X and Y are for the same purpose for each LF in Q, respectively.

As depicted in Step 1 of Algorithm NIR, we pre-compute and cache the instance-frequency-weight for each involved LF, i.e., $|\Lambda(m)|^{\beta}:\forall m\in M$ and $|\Lambda(n)|^{\beta}:\forall n\in N$ to avoid redundant operations during the iterations. Then we train P and Q iteratively, conducting each iteration as follows: a) traverse on Λ to compute the training increment brought by each training instance to the numerator and denominator of the update rule for each LF in P and Q, as shown in Step 3 of Algorithm NIR; and b) traverse on M and N to update P and Q by integrating corresponding IR terms into their learning rule.

ALGORITHM NIR

Input: Λ, M, N	
Operation	**Complexity**
1. Initialize $P^{\|M\|\times d}$, $A^{\|M\|\times d}$, $B^{\|M\|\times d}$, $Q^{\|N\|\times d}$, $X^{\|N\|\times d}$, $Y^{\|N\|\times d}$	$\Theta((\|M\|+\|N\|)\times d)$
Initialize d, γ, β, i, *Max_Iteration_Count*	$\Theta(1)$
traverse on Λ **get** $\|\Lambda(m)\|:\forall m\in M$, $\|\Lambda(n)\|:\forall n\in N$	$\Theta(\|\Lambda\|)^{*}$
traverse on M and N **get** $\|\Lambda(m)\|^{\beta}:\forall m\in M$, $\|\Lambda(n)\|^{\beta}:\forall n\in N$	$\Theta(\|M\|+\|N\|)^{**}$
2. **while not** converge **and** i<*Max_Iteration_Count* **do**	$\times i$
Set A, B, X, $Y = 0$	$\Theta((\|M\|+\|N\|)\times d)$
3. **traverse** on Λ **get** each $t_{m,n}$	$\times\|\Lambda\|$
comput $\hat{t}_{m,n}$	$\Theta(d)$
for k=1~d	$\times d$
$a_{m,k}=a_{m,k}+q_{n,k}\times t_{m,n}$	$\Theta(1)$
$b_{m,k}=b_{m,k}+q_{n,k}\times \hat{t}_{m,n}$	$\Theta(1)$
$x_{n,k}=x_{n,k}+p_{m,k}\times t_{m,n}$	$\Theta(1)$
$y_{n,k}=y_{n,k}+p_{m,k}\times \hat{t}_{m,n}$	$\Theta(1)$
end for	--
end for	--
4. **for** $m\in M$	$\|M\|$
for k=1~d	$\times d$
$b_{m,k}=b_{m,k}+\gamma\times\|\Lambda(m)\|^{\beta}\times p_{m,k}$	$\Theta(1)$
$p_{m,k}=a_{m,k}/b_{m,k}$	$\Theta(1)$
end for	--
end for	--
5. **for** $n\in N$	$\times\|N\|$
for k=1~d	$\times d$
$y_{n,k}=y_{n,k}+\gamma\times\|\Lambda(n)\|^{\beta}\times q_{n,k}$	$\Theta(1)$
$q_{n,k}=x_{n,k}/y_{n,k}$	$\Theta(1)$
end for	--
end for	--
end while	--
Output: ***P***, ***Q***	

*Such low cost can be achieved with the help of a HashMap-like data structure.
**Note that to raise an arbitrary number to the power of β can be done in constant time in most industrial language.

Based on Algorithm NIR, we formulate the time cost of an NIR model as follows:

$$C_{Time}=\Theta\left(\left(|M|+|N|\right)\times d+i\times\left(\left(|M|+|N|\right)\times d+|\Lambda|\times d\right)\right)\approx\Theta\left(i\times|\Lambda|\times d\right) \quad (20)$$

Note that the last step of (20) is achieved based on the common condition that $\max\{|M|,|N|\}\ll|\Lambda|$ in most HiDS matrices, and also by ignoring the lower-order-terms and constant coefficients. Considering the storage complexity of Algorithm NIR, we formulate it as follows:

$$C_{Storage}=\Theta\left(\left(|M|+|N|\right)\times d+|\Lambda|\right)\approx\max\left\{\left(|M|+|N|\right)\times d,|\Lambda|\right\}, \quad (21)$$

which is either linear with the entity count or instance count of an HiDS matrix T.

IV. EXPERIMENTS

A. General Settings

Evaluation Protocol. Given an HiDS matrix T, one major task is to estimate its unknown entries in Γ based on its known ones in Λ due to the great need to recover the full relationship mapping between M and N [1-13]. Hence, this study adopts the task of missing data estimation as the evaluation protocol. More specifically, given Λ, such a task makes a tested model predict unobserved data in Γ. The outcome is measured on a validation set Ψ disjoint with Λ. For validating the accuracy of a tested model, we choose the mean absolute error (MAE), which is widely adopted for validating the statistical accuracy of an LFA-based model when predicting missing data in an HiDS matrix [1-3, 14-20, 35-39, 42]. It is given by

$$MAE=\left(\sum_{t_{m,n}\in\Psi}\left|t_{m,n}-\hat{t}_{m,n}\right|_{abs}\right)\Big/|\Psi|,$$

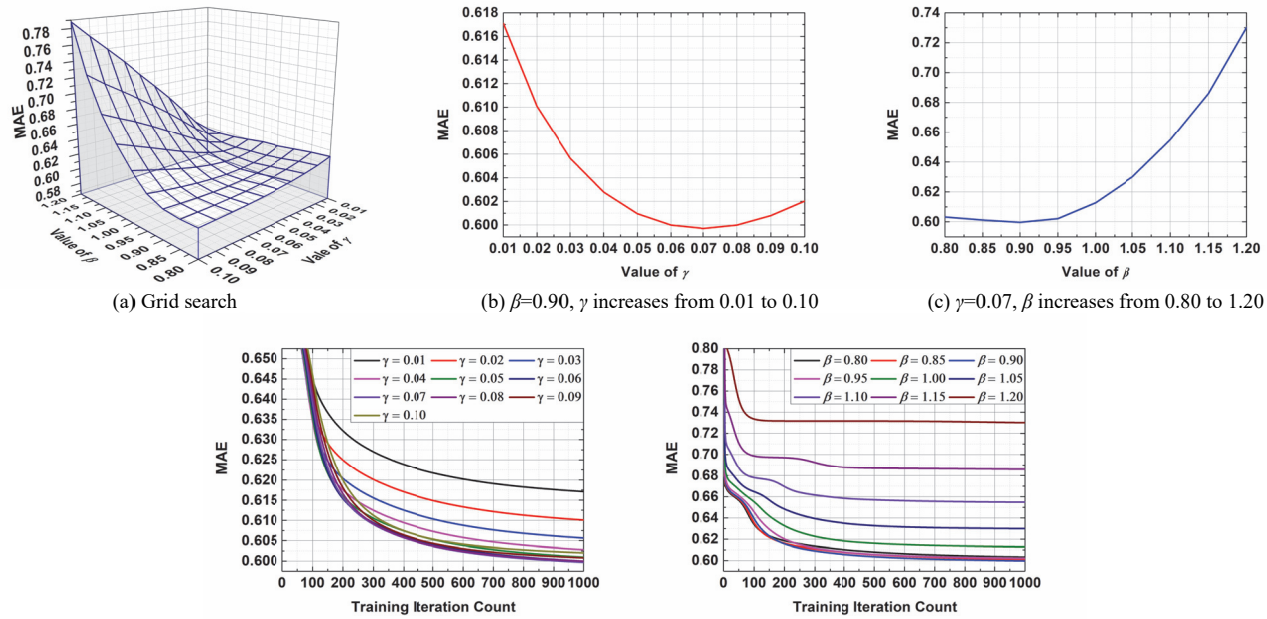
where $|\cdot|_{abs}$ denotes the absolute value of a given number. Note that All experiments are conducted on a PC with a 2.5 GHz i7 CPU and 32 GB RAM. All models are implemented in JAVA SE 7U60 to check their suitability for industrial usage.

Experimental Datasets. Two HiDS datasets are adopted
a) D1: MovieLens 20M. It is collected by the MovieLens system [43] and maintained by the GroupLens research team. It contains 20,000,263 known entries in the scale of [0.5, 5], by 138,493 users on 26,744 movies. Its data density is 0.54% only;
b) D2: Douban. It is collected by the Chinese largest online book, movie and music reviewing application Douban [44], and contains 16,830,839 ratings in the scale of [1, 5] from 129,490 users on 58,541 movies. Its data density is 0.22% only.
Note that both datasets are a) high-dimensional, b) extremely sparse, and c) collected by industrial applications currently in use. Hence, results on them are highly representative.

In our experiments, either dataset is randomly split into five equally-sized, disjoint subsets. On both datasets, we adopt the 80%-20% train-test settings and five-fold cross-validations, i.e., each time we select four subsets as Λ to train a model predicting the remaining one subset as Ψ. This process is sequentially repeated for five times to achieve the final results. During the experiments, the training process of a tested model terminates if a) the number of consumed iterations reaches a preset threshold, i.e., 1000, and b) the model converges, i.e., its error arises, or the difference in the training error of two consecutive iterations is smaller than 10^{-5}.

B. Parameter Sensitivity Tests

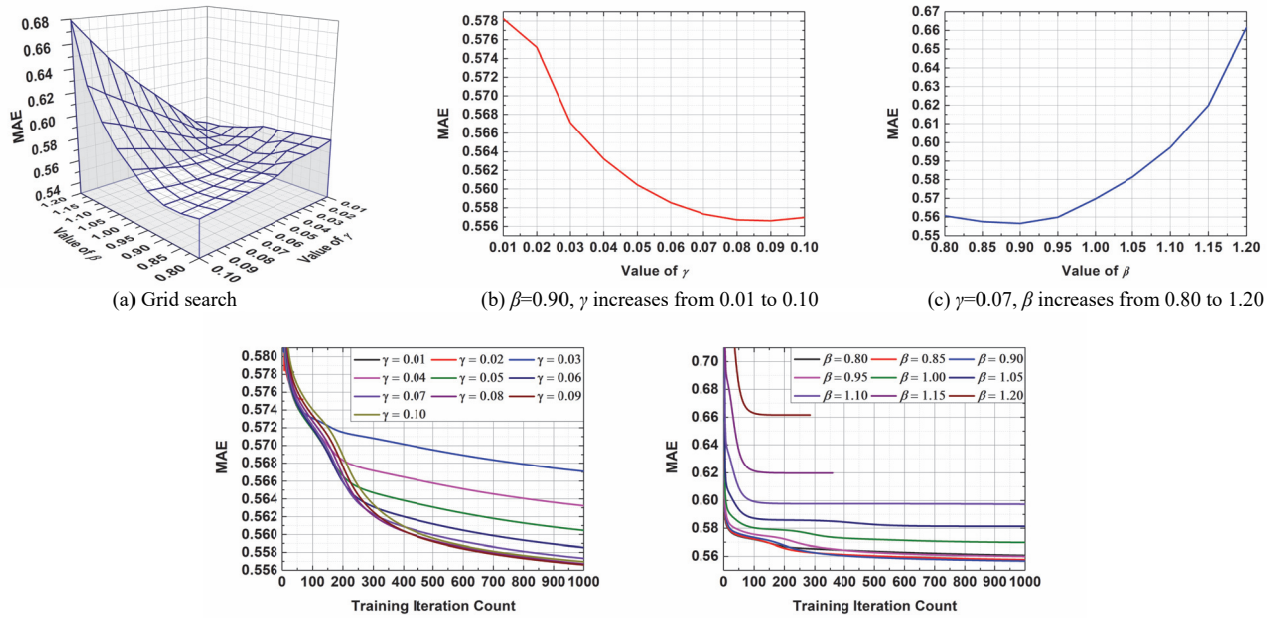
From (19) we see that an NIR model depends on γ and β. So we conduct parameter sensitivity tests with them. Considering β, we aim to validate different cases when the regularization effects increase a) slower than, b) faster than, and c) linearly with corresponding instance-frequency weight. Hence, its testing scale is set as (0.8, 1.2). In terms of γ, it tunes the regularization effects linearly and uniformly across a whole model. So we make it grow from 0.01 to 0.10 linearly.



(a) Grid search (b) β=0.90, γ increases from 0.01 to 0.10 (c) γ=0.07, β increases from 0.80 to 1.20

(d) Training curves with γ in [0.01, 0.10] and β=0.90 (e) Training curves with β in [0.80, 1.20] and γ=0.07.

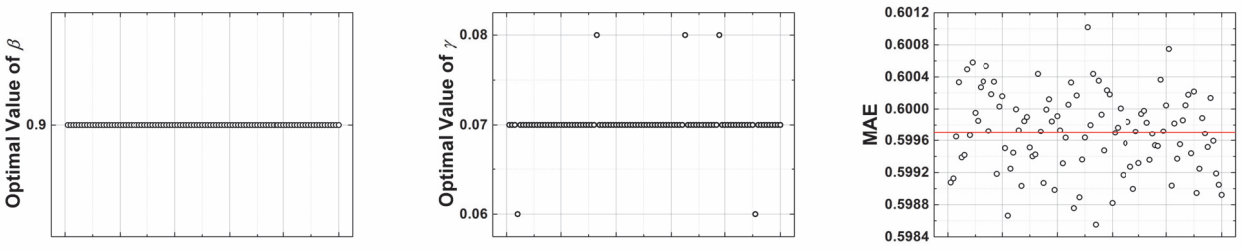
Fig. 2. Parameter sensitivity tests in an NIR model on D1.



(a) Grid search (b) β=0.90, γ increases from 0.01 to 0.10 (c) γ=0.07, β increases from 0.80 to 1.20

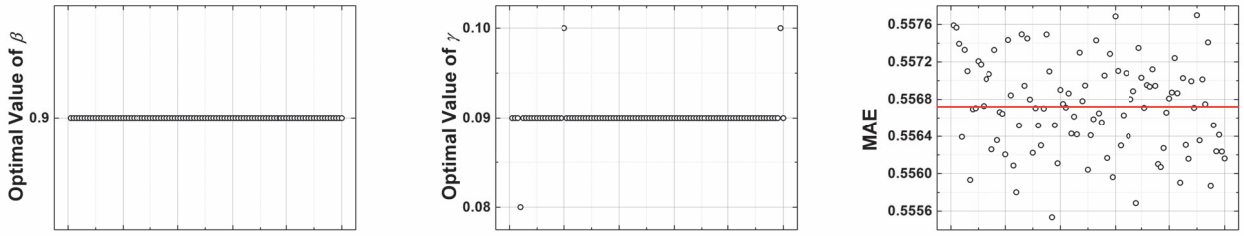
(d) Training curves with γ in [0.01, 0.10] and β=0.90 (e) Training curves with β in [0.80, 1.20] and γ=0.07.

Fig. 3. Parameter sensitivity tests in an NIR model on D2.



(a) Optimal value of β (b) Optimal value of γ (c) MAE of resultant models

Fig. 4. Performance of an NIR model: optimal hyper parameters and MAE with different initial hypotheses for P and Q in 100 independent tests on D1.



(a) Optimal value of β (b) Optimal value of γ (c) MAE of resultant models

Fig. 5. Performance of an NIR model: optimal hyper parameters and MAE with different initial hypotheses for P and Q in 100 independent tests on D2.

Results. The parameter sensitivity results are depicted in Figs. 2 and 3. From them, we find that:

a) γ and β decide an NIR model's prediction accuracy for missing data. As depicted in Fig. 2(b), when fixing β=0.9 on D1, the MAE of an NLFA-based model is 0.5997 with γ=0.07, and 0.6171 with γ=0.01, respectively. The numerical gap of MAE is 2.82%. When fixing γ=0.07, the MAE of an NLFA-based model is 0.5997 with β=0.9, and 0.7301 with β=1.2, respectively. The numerical gap of MAE is 17.86%. Similar situations can be found on D2, as shown in Fig. 3(b). When fixing β=0.9, the MAE is 0.5566 with γ=0.09, and 0.5782 with γ=0.01, respectively. The numerical gap of MAE is 3.74%. With γ=0.09, the MAE is 0.5566 with β=0.9, and 0.6615 with β=1.2, respectively. The numerical gap of MAE is 15.86%. From these results, we see that the prediction accuracy of NIR is closely connected with β and γ: they should be chosen with care to achieve the lowest MAE.

b) Optimal β and γ are data-dependent. For instance, on D1 NIR achieves the lowest MAE at 0.5997 with β=0.9 and γ=0.07, as shown in Fig. 2(a). On D2, it achieves the lowest MAE at 0.5566 with β=0.9 and γ=0.09, as shown in Fig. 3(a). On both datasets, the optimal β is 0.9. As given in (19), β controls the instance-frequency-related regularization on each individual LF. It monotonically increases with instance-frequency weight, but can grows slower when β<1. It appears necessary to carefully control the regularization increment at individual LF-related instance-frequency-weight to achieve a highly accurate NIR model. On the other hand, as given in (19), γ controls the regularization effect from a model perspective. Its optimal value is different on D1 and D2. Note they differ from each other vastly in data distribution, entity count, instance count, and data density. These factors are closely connected with regularization effects [14-20, 35, 36].

c) β and γ can affect NIR's convergence rate. As depicted in Figs. 2(d), 2(e), 3(d) and 3(e), in most testing cases NIR's training iteration count reaches the upper bound, i.e., 1,000 as β and γ vary. However, on some extreme cases on D2, i.e., with β>1.1, an NLFA-based model is tracked by some local saddle points after much less iterations, as given in Fig. 3(e). Nonetheless, this phenomenon is explainable: according to (19), as β grows over certain threshold, regularization effects in IR terms can be over amplified. Under such circumstances, a NIR model can suffer under fitting instead of over fitting, making it suffer accuracy loss.

d) Selection of β and γ does not depend on initial hypotheses. We conduct 100 independent tests with randomly generated, different initial values of P and Q on both datasets, whose results are given in Figs. 4 and 5. From them, we see that optimal β and γ are rarely affected by the initial hypotheses. For β, its optimal value keeps on 0.9 on both datasets, as depicted in Figs. 4(a) and 5(a). For γ, its optimal value may change with different initial hypotheses; however, such situations are seldom encountered. On D1, the optimal value of γ is 0.07 on most cases (95 out of 100); on only a few cases it may become 0.06 (2 out of 100) or 0.08 (3 out of 100), as given in Fig. 4(b). Similar situations are also encountered on D2, as depicted in Fig. 5(b). Nonetheless, according to Figs. 4(c) and 5(c), an NIR model's MAE may fluctuate in the scale of 0.5% with different initial hypotheses. Such situation is consistent with those encountered in prior LFA-based models [14-20, 35-39]. As discussed in [36], such randomness can help to build more accurate NLFA ensemble. Further investigations into this issue are in our future plan.

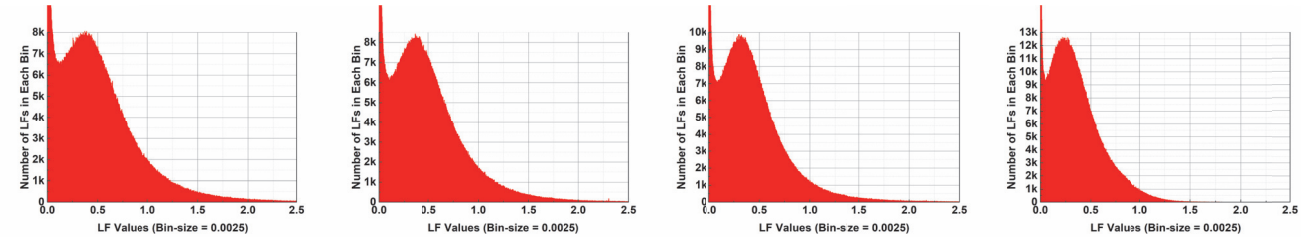
Based on the above findings, we summarize that an NIR model's prediction accuracy for missing data depends heavily on its hyper parameters, i.e., β and γ in (15). Considering the parameter selection, β is more stable, whose optimal value is at 0.9 on both datasets. The optimal value of γ is more data-dependent, scattering in the (0.07, 0.09) interval. However, selection of β and γ is rarely affected by an NIR model's initial hypothesis.

C. Comparison against other Regularization Schemes

TABLE I COMPARED REGULARIZATION SCHEMES

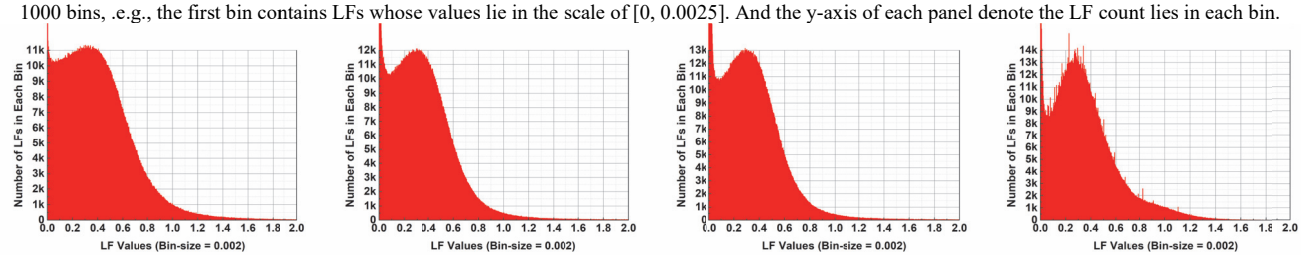
No.	Regularization terms
M1	None.
M2	$L=\frac{\gamma\left(\lvert M\rvert+\lvert N\rvert\right)}{2}\left(\lVert P\rVert_F^2+\lVert Q\rVert_F^2\right)$
M3	$L=\frac{\gamma}{2}\left(\lvert N\rvert\lVert P\rVert_F^2+\lvert M\rvert\lVert Q\rVert_F^2\right)$
NIR	$L(P,Q)=\frac{\gamma}{2}\left(\sum_{m=1}^{\lvert M\rvert}\lvert\Lambda(m)\rvert^{\beta}\lVert P_{m*}\rVert_2^2+\sum_{n=1}^{\lvert N\rvert}\lvert\Lambda(n)\rvert^{\beta}\lVert Q_{n*}\rVert_2^2\right)$

Four models with different regularization schemes are compared in this part of experiments, whose details are summarized in Table I. To achieve unbiased results, we set the experiments as follows: a) d is fixed at 20 for all models; b) P and Q are initialized with the same arrays whose elements are randomly selected from the (0, 0.01) interval following a uniform distribution to eliminate the impact by different initial hypotheses; c) for M1 and M2, we tune the value of γ to achieve



(a) M1 (b) M2 (c) M3 (d) NIR

Fig. 6. LF distribution of involved models on D1. Note that the x-axis of each panel denote the value of acquired LFs by a model, which is sequentially split into 1000 bins, .e.g., the first bin contains LFs whose values lie in the scale of [0, 0.0025]. And the y-axis of each panel denote the LF count lies in each bin.



(a) M1 (b) M2 (c) M3 (d) NIR

Fig. 7. LF distribution of involved models on D2. Note that the x-axis of each panel denote the value of acquired LFs by a model, which is sequentially split into 1000 bins, .e.g., the first bin contains LFs whose values lie in the scale of [0, 0.002]. And the y-axis of each panel denote the LF count in each bin.

the lowest prediction error on one fold of either dataset, and then adopt the tuned γ on the other four; and d) for NIR, we set β=0.9 and γ=0.07 on D1, and β=0.9 and γ=0.09 on D2 according to results in Section IV(B).

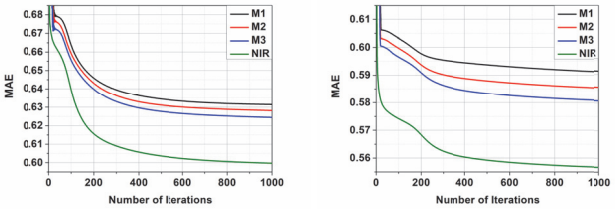
TABLE II LOWEST MAE ACHIEVED BY EACH MODEL

Model	M1	M2	M3	NIR
D1	0.6314	0.6281	0.6244	**0.5997**
D2	0.5912	0.5853	0.5808	**0.5567**

Figs. 6 and 7 depict the LF distribution of NLFA-based models with different regularization schemes. Fig. 8 depict the training process of NLFA-based models with different regularization schemes on D1 and D2. Table II depicts the lowest MAE achieved by compared models. From these results, we have the following findings:

a) An NIR model outperforms its peers inters of prediction accuracy for missing data owing to its incorporation of an IR scheme. As depicted in Table II and Fig. 8(a), the MAE of M1, M2 and M3 is 0.6314, 0.6281 and 0.6244, respectively. In comparison, an NIR model's MAE is 0.5997. Compared to M1-3, it achieves accuracy gain of 5.02%, 4.52% and 3.96%, respectively. Similar situations can be found on D2, as shown in Table II and Fig. 8(b). M1-3 and NIR achieve the MAE at 0.5912, 0.5853, 0.5808 and 0.5567, respectively. Compare with M1-3, an NIR model achieves its accuracy gain at 5.83%, 4.88% and 4.14%, respectively.

b) An NIR model achieves more specific LF distribution on an HiDS matrix. As depicted in Figs. 6 and 7, with different regularization schemes, resultant models achieve different LF distributions. From Figs. 6 and 7, we observe an interesting phenomenon that with regularization schemes precisely describing the characteristics of the target HiDS matrix, an NLFA-based model's LF distribution tends to be more specific. From Figs. 6 and 7, we clearly see that an NIR model's LF distribution is the most specific out of involved models. More specifically, LF distributions are becoming more centralized as well as their shape becomes more specific with a more advanced regularization scheme. This phenomenon indicates that with a well-designed regularization scheme enhances an NLFA-based model's representativeness to a target HiDS matrix, thereby guaranteeing its high prediction accuracy for its missing data.



(a) D1 (b) D2

Fig. 8. Involved models' training process on both datasets.

c) An NLFA-based model's convergence rate will merely be affected by a tested regularization scheme. In our experiments, all models consume 1,000 training iterations to achieve the highest prediction accuracy for missing data of an HiDS matrix, which is the upper bound in our experiments.

D. Summary

Based on the experimental results, we summarize that:

a) The effects by IR are closely related to the hyper parameters γ and β. β controls the regularization effects from a global perspective of the whole model, while γ controls the regularization effects applied to individual LFs. In general, the selection of γ is more data-dependent than β;

b) An IR scheme can help an NLFA-based model better represent an HiDS matrix, thereby achieving high prediction accuracy for its missing data.

V. CONCLUSIONS

To perform NLFA on an HiDS matrix is ill-posed, making an NLFA-based model eager for a precise regularization scheme to avoid overfitting [14-24, 35-39]. Commonly adopted L_2 regularization scheme does not consider an HiDS matrix's sparsity or imbalanced data distribution, which can weaken a resultant NLFA-based model's representative ability to it. This study proposes an IR scheme to address this problem. It considers instance-frequency weight corresponding to each LF,

which actually describes the sparsity and imbalanced data distribution of an HiDS matrix. Thus, it precisely controls regularization effects applied to each individual LF, thereby achieving a model with fine representativeness to an HiDS matrix. Based on the experimental results, an NIR model can achieve significantly higher prediction accuracy for missing data of an HiDS matrix compared with NLFA-based models with L_2 schemes. An IR scheme proves to be very effective in improving the generality of an NLFA-based model defined on an HiDS matrix.

Future extensions of IR. In the future, we plan to extend an IR scheme in the following aspects:

1) This study takes an L_2 regularization scheme as the baseline. Although an L_2 scheme has proven to be very effective in improving the generality of an NLFA-based model when addressing the task of missing data estimation [14-18, 35-39], other regularization schemes like an L_1 scheme or elastic-net scheme [45, 46] work well in other data-analysis tasks like community detection [8, 9, 21]. Note that the principle of IR is also compatible with them, making it highly interesting to fully investigate the performance of IR series in data analysis tasks.

2) It is significant to investigate the effects of different mapping functions in an IR scheme according to (13). Meanwhile, an IR scheme relies on a single mapping function, i.e., a power function as presented in Section III. As a matter of fact, several mapping functions can be adopted to form compound effects, i.e., making the regularization term $R=f(x)+g(x)$ with f and g being different mapping functions. This issue should be also addressed to achieve more effective regularization schemes.

3) As indicated in Section IV, performance of an NIR model depends on its hyper parameters β and γ. How to make them self-adaptive remains an open issue, which is highly worthy of investigations to achieve more practical models.

4) It is desired to validate the compatibility between an IR scheme and other learning objectives like β-distance or L_p norm [14-24].

We plan to address the above issues in the future.

REFERENCES

[1] J. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative-filtering," in *Proc. of the 14th Int. Conf. on Uncertainty in Artificial Intelligence*, San Francisco, California, USA, 1998, pp. 43-52.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item Based Collaborative-filtering Recommendation Algorithms," in *Proc. of the 10th Int. Conf. on World Wide Web*, Hong Kong, PRC, 2001, pp. 285-295.

[3] G. Adomavicius and A. 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, pp. 734-749, 2005.

[4] Z.-B. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-Aware Web Service Recommendation by Collaborative Filtering," *IEEE Trans. on Services Computing,* vol. 4, pp. 140-152, 2011.

[5] J. Wu, L. Chen, Y.-P. Feng, Z.-B. Zheng, M.-C. Zhou, and Z. Wu, "Predicting Quality of Service for Selection by Neighborhood-Based Collaborative-filtering," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 43, pp. 428-439, 2013.

[6] X. Luo, M. Zhou, Z. Wang, Y. Xia, and Q. Zhu, "An Effective QoS Estimating Scheme via Alternating Direction Method-based Matrix Factorization," *IEEE Trans. on Services Computing*, DOI 10.1109/TSC.2016.2597829.

[7] R. Narayanam, and Y. Narahari, "A Shapley Value-Based Approach to Discover Influential Nodes in Social Networks," *IEEE Trans. on Automation Science and Engineering,* vol. 8, no. 1, pp. 130-147, Jan, 2011.

[8] C.-L. Liu, J. Liu, and Z.-Z. Jiang, "A Multiobjective Evolutionary Algorithm Based on Similarity for Community Detection From Signed Social Networks," *IEEE Trans. on Cybernetics,* vol. 44, no. 12, pp. 2274-2287, Dec., 2014.

[9] L. Yang, X. Cao, D. Jin, X. Wang, and D. 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.

[10] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized Recommendation Combining User Interest and Social Circle," *IEEE Trans. on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1763-1777, 2014.

[11] H. N. Chua, and L. Wong, "Increasing the Reliability of Protein Interactomes," *Drug Discovery Today,* vol. 13, no. 15-16, pp. 652-658, Aug, 2008.

[12] Z.-H. You, Y.-K. Lei, J. Gui, D.-S. Huang, and X.-B. 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.

[13] Z.-H. You, M.-C. Zhou, X. Luo, and S. Li, "Highly Efficient Framework for Predicting Interactions Between Proteins," *IEEE Trans. on Cybernetics,* vol. 47, no. 3, pp. 721-733, 2017.

[14] G. Takács, I. Pilászy, Bottyán Németh, and D. Tikky, "Scalable Collaborative Filtering Approaches for Large Recommender Systems," *Journal of Machine Learning Research,* vol. 10, pp. 623-656, 2009.

[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix-factorization Techniques for Recommender Systems," *IEEE Computer,* vol. 42, pp. 30-37, 2009.

[16] Y. Koren and R. Bell, "Advances in Collaborative Filtering," in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., ed New York: Springer US, 2011, pp. 145-186.

[17] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix-factorization," *Advances in Neural Information Processing Systems,* vol. 20, pp. 1257–1264, 2008.

[18] K. Yu, S. Zhu, J. Lafferty, and Y. Gong, "Fast Nonparametric Matrix-factorization for Large-scale Collaborative-filtering," in *Proc. of the 32nd ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*, Boston, Massachusetts, USA, 2009, pp. 211-218.

[19] N. Srebro, and R. Salakhutdinov, "Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm," in *Advances in Neural Information Processing Systems 23 (NIPS)*, Vancouver, British Columbia, Canada, 2010, pp. 2056-2064.

[20] S. Chatzis, "Nonparametric Bayesian multitask collaborative filtering," in *Proc. of the 22nd ACM Int. Conf. on Information and Knowledge Management*, San Francisco, California, USA, 2013, pp. 2149-2158.

[21] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, "MetaFac: Community Discovery via Relational Hypergraph Factorization," in *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Paris, France, 2009.

[22] Z. Ning, W. K. Cheung, Q. Guoping, and X. Xiangyang, "A Hybrid Probabilistic Model for Unified Collaborative and Content-Based Image Tagging," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 33, no. 7, pp. 1281-1294, 2011.

[23] M.-F. Weng, and Y.-Y. Chuang, "Collaborative Video Reindexing via Matrix Factorization," *ACM Trans. on Multimedia Computing, Communications, and Applications,* vol. 8, no. 2, pp. 1-20, 2012.

[24] J. J. Pan, S. J. Pan, Y. Jie, L. M. Ni, and Y. Qiang, "Tracking Mobile Users in Wireless Networks via Semi-Supervised Colocalization," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 34, no. 3, pp. 587-600, 2012.

[25] P. Paatero, and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics,* vol. 5, no. 2, pp. 111-126, 1994.

[26] D. D. Lee and S. H. Seung, "Learning the Parts of Objects by Non-negative Matrix-factorization," *Nature,* vol. 401, pp. 788-791, 1999.

[27] C.-J. Lin, "Projected Gradient Methods for Nonnegative Matrix Factorization," *Neural Computation*, vol. 19, pp. 2756-2779, 2007.

[28] P. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *Journal of Machine Learning Research,* vol. 5, pp. 1457-1469, 2004.

[29] H. Kim, and H. 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.

[30] C. Ding, L. Tao, and M. I. Jordan, "Convex and Semi-nonnegative Matrix-factorizations," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 45-55, 2010.

[31] L.-X. Li, L. Wu, H.-S. Zhang, and F.-X. Wu, "A Fast Algorithm for Nonnegative Matrix Factorization and Its Convergence," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 10, pp. 1855-1863, 2014.

[32] M. Ye, Y. Qian, and J. Zhou, "Multitask Sparse Nonnegative Matrix Factorization for Joint Spectral-Spatial Hyperspectral Imagery Denoising," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2621-2639, 2015.

[33] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from Incomplete Ratings Using Non-negative Matrix-factorization," in *Proc. of the SIAM Int. Conf. on Data Mining*, Bethesda, Maryland, USA, 2006, pp. 549-553.

[34] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An Alternating Direction Algorithm for

Matrix Completion with Nonnegative Factors," *Frontiers of Mathematics in China,* vol. 7, no. 2, pp. 365-384, 2012.
[35] X. Luo, M.-C. Zhou, Y.-N. Xia, and Q.-S. 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, 2014.
[36] X. Luo, M.-C. Zhou, Y.-N. Xia, Q.-S. Zhu, A. C. Ammari, and A. 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, 2016.
[37] X. Luo, M.-C. Zhou, S. Li, Z.-H. You, Y.-N. Xia, and Q.-S. Zhu, "A Nonnegative Latent Factor Model for Large-Scale Sparse Matrices in Recommender Systems via Alternating Direction Method," *IEEE Trans. on Neural Networks and Learning Systems,* vol. 27, no. 3, pp. 579-592, 2016.
[38] X. Luo, M. Shang, and S. Li, "Efficient Extraction of Non-negative Latent Factors from High-dimensional and Sparse Matrices in Industrial Applications," in *Proc. of the IEEE Int. Conf. on Data Mining*, Barcelona, Spain, 2016, pp. 311-319.
[39] X. Luo, M. Zhou, M. Shang, S. Li, and Y. Xia, "A Novel Approach to Extracting Non-Negative Latent Factors From Non-Negative Big Sparse Matrices," *IEEE Access,* vol. 4, pp. 2649-2655, 2016.
[40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning,* vol. 3, no. 1, pp. 1-122, 2011.
[41] S. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge: Cambridge University Press, 2009.
[42] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. on Information Systems,* vol. 22, no. 1, pp. 5-53, 2004.
[43] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to Usenet news," *Communications of the ACM,* vol. 40, no. 3, pp. 77-87, 1997.
[44] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender Systems with Social Regularization," in *Proc. of the 4th ACM Int. Conf. on Web Search and Data Mining*, Hong Kong, China, 2011, pp. 287-296.
[45] Q. Li, B. Xie, J. You, W. Bian, and D. Tao, "Correlated Logistic Model With Elastic Net Regularization for Multilabel Image Classification," *IEEE Trans. on Image Processing,* vol. 25, no. 8, pp. 3801-3813, 2016.
[46] Y. Feng, S. G. Lv, H. Hang, and J. A. K. Suykens, "Kernelized Elastic Net Regularization: Generalization Bounds, and Sparse Recovery," *Neural Computation,* vol. 28, no. 3, pp. 525-562, 2016.
[47] G. Cao, L. R. Bachega, and C. A. Bouman, "The Sparse Matrix Transform for Covariance Estimation and Analysis of High Dimensional Signals," *IEEE Trans. on Image Processing,* vol. 20, no. 3, pp. 625-640, 2011.
[48] Z. Zhang, W. Jiang, F. Li, M. Zhao, B. Li, and L. Zhang, "Structured Latent Label Consistent Dictionary Learning for Salient Machine Faults Representation-Based Robust Classification," *IEEE Trans. on Industrial Informatics,* vol. 13, no. 2, pp. 644-656, 2017.
[49] Z. Zhang, T. W. S. Chow, and M. Zhao, "M-Isomap: Orthogonal Constrained Marginal Isomap for Nonlinear Dimensionality Reduction," *IEEE Trans. on Cybernetics,* vol. 43, no. 1, pp. 180-191, 2013.
[50] W. J. Luan, G. J. Liu, C. J. Jiang, and L. Qi, "Partition-based Collaborative Tensor Factorization for POI Recommendation," *IEEE/CAA Journal of Automatica Sinica,* vol. 4, no. 3, pp. 437-446, Jul, 2017.
[51] H. Li, K. L. Li, J. W. Peng, J. Y. Hu, and K. Q. Li, "An Efficient Parallelization Approach for Large-scale Sparse Non-negative Matrix Factorization Using Kullback-Leibler Divergence on Multi-GPU," in *Proc. of the 15th IEEE Int. Symp. on Parallel and Distributed Processing With Applications and 16th IEEE Int. Conf. on Ubiquitous Computing and Communications*, pp. 511-518, 2017.
[52] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, "A Deep Matrix Factorization Method for Learning Attribute Representations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 417-429, Feb, 2017.
[53] M. G. Gong, X. M. Jiang, H. Li, and K. C. Tan, "Multi-objective Sparse Non-Negative Matrix Factorization," *IEEE Trans. on Cybernetics*, vol. 49, no. 8, pp. 2941-2954, Aug, 2019.
[54] H. Xiong, and D. G. Kong, "Elastic Nonnegative Matrix Factorization," *Pattern Recognition,* vol. 90, pp. 464-475, Jun, 2019.

Xin Luo (M'14–SM'17) received the B.S. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2005 and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2011. In 2016, he joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China, as a Professor of computer science and engineering. He is currently also a Distinguished Professor of computer science with the Dongguan University of Technology, Dongguan, China. His current research interests include big data analysis and intelligent control. He has published over 100 papers (including over 30 IEEE TRANSACTIONS papers) in the above areas. Dr. Luo was a recipient of the Hong Kong Scholar Program jointly by the Society of Hong Kong Scholars and China Post-Doctoral Science Foundation in 2014, the Pioneer Hundred Talents Program of Chinese Academy of Sciences in 2016, and the Advanced Support of the Pioneer Hundred Talents Program of Chinese Academy of Sciences in 2018. He is currently serving as an Associate Editor for the IEEE/CAA JOURNAL OF AUTOMATICA SINICA, IEEE ACCESS, and *Neurocomputing*. He has received the Outstanding Associate Editor reward of IEEE ACCESS in 2018. He has also served as the Program Committee Member for over 20 international conferences.

Zidong Wang (SM'03-F'14) was born in Jiangsu, China, in 1966. He received the B.Sc. degree in mathematics from Soochow University, Suzhou, China, and the M.Sc. degree in applied mathematics and the Ph.D. degree in electrical engineering from Nanjing University of Science and Technology, Nanjing, China, in 1986, 1990, and 1994, respectively.

He is currently a Professor of dynamical systems and computing in the Department of Information Systems and Computing, Brunel University London, Middlesex, U.K. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany, and the U.K. He has published more than 400 papers in refereed international journals. He is a holder of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, and the William Mong Visiting Research Fellowship of Hong Kong. His research interests include dynamical systems, signal processing, bioinformatics, and control theory and applications.

Prof. Wang serves (or has served) as the Editor-in-Chief for Neurocomputing and an Associate Editor for 12 international journals, including the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS--Part C. He is a Fellow of the Royal Statistical Society and a member of program committee for many international conferences.

Mingsheng Shang received his B.E. degree in Management in Sichuan Normal University in Chengdu, China in 1995, and Ph.D. degree in Computer Science from University of Electronic Science and Technology of China in Chengdu, China in 2007. He is currently a professor at the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China. His research interests are in complex network analysis and big data applications.