This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

Convergence Analysis of Single Latent Factor-dependent, Non-negative and Multiplicative Update-based Non-negative Latent Factor Models

Zhigang Liu, Xin Luo, *Senior Member*, *IEEE*, and Zidong Wang, *Fellow*, *IEEE*

***Abstract*—**A single latent factor-dependent, non-negative and multiplicative update (SLF-NMU) learning algorithm is highly efficient in building a non-negative latent factor (NLF) model defined on an HiDS matrix. However, convergence characteristics of such NLF models are never justified in theory. To address this issue, this study conducts rigorous convergence analysis for an SLF-NMU-based NLF model. The main idea is two-fold, a) proving that its learning objective keeps non-increasing with its SLF-NMU-based learning rules via constructing specific auxiliary functions, and b) proving that it converges to a stable equilibrium point with its SLF-NMU-based learning rules via analyzing the Karush-Kuhn-Tucker (KKT) conditions of its learning objective. Experimental results on ten HiDS matrices from real applications provide numerical evidence that indicates the correctness of the achieved proof.

***Index Terms*—**Learning System, Single Latent Factor-dependent Non-negative and Multiplicative Update, Non-negative Latent Factor Analysis, Neural Networks, Convergence, Latent Factor Analysis, High-Dimensional and Sparse Matrix, Big Data

I. Introduction

A HIGH-DIMENSIONAL AND SPARSE (HiDS) matrix is commonly adopted to describe incomplete interactions among concerning objects in big data-related applications, e.g., user-service invocations in services computing [1], [3], user-item preferences in recommender systems [4], and protein interactomes in bioinformatics [7]. Despite its extreme sparsity, it contains tremendously useful knowledge such as potential links [11]-[13], community tendency [14]-[16], [54] and cluster [17], [35]. A latent factor (LF)-based approach [4], [5], [18], [19] can efficiently implement knowledge acquisition from an HiDS matrix. It works by 1) mapping concerned objects into a low-dimensional LF space, 2) defining a learning objective on the known data of an HiDS target with desired LFs, and 3) optimizing the objective to achieve LFs precisely outlining concerned objects, which are useful in addressing subsequent learning tasks like missing data recovering [9]-[17], [43].

Non-negativity is a common characteristic of most real data, e.g., social impacts from social network services applications [16], [27]. To well represent their non-negativity, an LF model should be restricted to be non-negative [3], [26], [28]. A non-negative matrix factorization (NMF) model [29]-[36], [48]-[52] is designed to address full matrices under the non-negative constraints, but it cannot address an HiDS one directly. Although such a defect can be overcome via specific model designs like weight indication [48] or intermediate matrix incorporation [49], it further suffers high costs in both computation and storage. This is because its costs are proportional to the target matrix's full size [3], [28]. When the target is HiDS, its full size can be much larger than the size of its known data. For instance, the MovieLens 20M [44] matrix's full size is 3.7 billion, while the density of its known data is 0.54%. Due to its extremely low density, it is very inefficient to make a model's costs linear with its full size.

For efficiently implementing non-negative latent factor (NLF) analysis on an HiDS matrix, Luo et al. [3], [28] present a single latent factor-dependent, non-negative and multiplicative update (SLF-NMU) algorithm. It is defined on an HiDS matrix's known data, thereby reducing the storage and computational costs to be linear with the size of its known data. Although it builds an NLF model with high efficiency, its convergence characteristics are never justified formally. For addressing this issue, this paper aims at theoretically proving that an NLF model converges to a Karush-Kuhn-Tucker (KKT) equilibrium point with an SLF-NMU learning algorithm. Main contributions of this work include

a) Rigorous proof demonstrating that an SLF-NMU-based NLF model's learning objective is non-increasing during the training process via building specific auxiliary functions corresponding to its parameter update rules;
b) Rigorous proof demonstrating that an SLF-NMU-based NLF model's parameter learning sequence converges to a KKT

This research is supported in part by the National Natural Science Foundation of China under grants 61772493 and 61933007, in part by the Natural Science Foundation of Chongqing (China) under grant cstc2019jcyjjqX0013, and in part by the Pioneer Hundred Talents Program of Chinese Academy of Sciences (*Corresponding Author: X. Luo*).

Z. Liu is with the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, and the Chongqing Engineering Research Center of Big Data Application for Smart Cities, and the 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: liuzhigangx@gmail.com).

X. Luo is with the Chongqing Engineering Research Center of Big Data Application for Smart Cities, and the Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, and the Hengrui (Chongqing) Artificial Intelligence Research Center, Department of Big Data Analyses Techniques, Cloudwalk, Chongqing 401331, China (e-mail: luoxin21@cigit.ac.cn).

Z. Wang is with the Department of Computer Science, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom (e-mail: Zidong.Wang@brunel.ac.uk).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

equilibrium point of its learning objective; and

c) Detailed empirical studies on ten HiDS matrices collected by big data-related applications.

Note that convergence analysis for a non-convex learning model is critical in the optimization community [2], [5], [6], [8], [21]-[25]. According to prior research [3], [11], [12], an SLF-NMU-based NLF model is a bi-linear and thus non-convex learning model. Therefore, its rigorous convergence proof is highly useful in providing insights into its learning ability with an HiDS matrix as its input. Moreover, as discussed in [66], [67], performing LF analysis on an HiDS matrix is actually equivalent to building a single-layered neural network whose inputs and outputs are both the target HiDS matrix. Note that in big data-related applications, HiDS matrices are far more frequently encountered than complete ones [3], [68]. It is becoming increasingly important to design a neural network-based learning system with incomplete inputs [3], [66]-[70]. Therefore, results achieved in this study can help people better understand the characteristics of a neural network defined on incomplete data.

The remainder of this paper is organized as follows. Section II gives the preliminaries. Section III presents rigorous proof regarding the convergence of an SLF-NMU-based NLF model. Section IV provides empirical studies. Section V discusses several critical points. Finally, Section VI concludes the paper.

II. Preliminaries

A. Problem Formulation

For LF analysis, we recall the definition of an HiDS matrix [3], [11], [12],

***Definition* 1.** Let M and N be two large entity sets, $Y^{|M|\times|N|}$ be a matrix whose element $y_{m,n}$ quantifies the interaction between $m\in M$ and $n\in N$, Λ and Γ be Y's known and unknown subsets. Y is an HiDS matrix if $|\Lambda|\ll|\Gamma|$.

Based on Λ, an LF model implements a rank-d estimation $\hat{Y}=AX^{\mathrm{T}}$ for Y with $A^{|M|\times d}$ and $X^{|N|\times d}$ being LF matrices. It should be pointed out that d is far less than the minimum of $|M|$ and $|N|$. For achieving A and X, we construct a loss function to measure the difference between Y and $\hat{Y}$ depending on Λ only. Based on the Euclidean distance, it is given by

$$\varepsilon\left(A,X\right)=\frac{1}{2}\sum_{y_{m,n}\in\Lambda}\left(y_{m,n}-\sum_{k=1}^{d}a_{m,k}x_{n,k}\right)^2, \tag{1}$$

where $a_{m,k}$, $x_{n,k}$ and $y_{m,n}$ denote specified elements in A, X, and Y. For correctly representing non-negative data [1], [3], [11]-[13], [16], [19], [27], (1) should fulfill the non-negativity constraints,

$$a_{m,k}\geq 0, x_{n,k}\geq 0,\ \forall m\in M,n\in N,k\in\left\{1,2,...,d\right\}. \tag{2}$$

Meanwhile, building an LF model on an HiDS matrix is ill-posed [3], [28], [42], making regularizations indispensable. Let $\hat{y}_{m,n}=\sum_{k=1}^{d}a_{m,k}x_{n,k}$, we reformulate (2) to incorporate L_2 norm-based regularizations into it,

$$\begin{aligned}&\varepsilon\left(A,X\right)=\frac{1}{2}\sum_{y_{m,n}\in\Lambda}\left(\left(y_{m,n}-\hat{y}_{m,n}\right)^2+\lambda_A\sum_{k=1}^{d}a_{m,k}^2+\lambda_X\sum_{k=1}^{d}x_{n,k}^2\right),\\ &s.t.\quad a_{m,k}\geq 0, x_{n,k}\geq 0,\ \forall m\in M,n\in N,k\in\left\{1,2,...,d\right\}.\end{aligned} \tag{3}$$

where λ_A and λ_X denote the regularization constants for A and X, respectively.

B. A Non-negative Latent Factor Model

An NLF model adopts an SLF-NMU-based learning algorithm to optimize A and X in (3). It initially applies additive gradient descent (AGD) to each LF:

$$\begin{aligned}&\arg\min_{A,X}\varepsilon\left(A,X\right)\overset{AGD}{\Rightarrow}\\ &\begin{cases}a_{m,k}\leftarrow a_{m,k}-\eta_{m,k}\sum_{n\in\Lambda(m)}\left(\lambda_A a_{m,k}-x_{n,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right),\\ x_{n,k}\leftarrow x_{n,k}-\eta_{n,k}\sum_{m\in\Lambda(n)}\left(\lambda_X x_{n,k}-a_{m,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right).\end{cases}\end{aligned} \tag{4}$$

where $\eta_{m,k}$ and $\eta_{n,k}$ are learning rates for $a_{m,k}$ and $x_{n,k}$, $\Lambda(m)$ and $\Lambda(n)$ are Λ's subsets related to m and n, respectively. From (4), we see that $a_{m,k}$ and $x_{n,k}$ can become negative due to $-\eta_{m,k}\sum_{n\in\Lambda(m)}\left(x_{n,k}\hat{y}_{m,n}+\lambda_A a_{m,k}\right)$ and $-\eta_{n,k}\sum_{m\in\Lambda(n)}\left(a_{m,k}\hat{y}_{m,n}+\lambda_X x_{n,k}\right)$. For canceling these negative terms to keep the non-negativity of A and X, an SLF-NMU algorithm manipulates $\eta_{m,k}$ and $\eta_{n,k}$,

$$\begin{aligned}\eta_{m,k}&=\frac{a_{m,k}}{\sum_{n\in\Lambda(m)}x_{n,k}\hat{y}_{m,n}+\lambda_A\left|\Lambda(m)\right|a_{m,k}},\\ \eta_{n,k}&=\frac{x_{n,k}}{\sum_{m\in\Lambda(n)}a_{m,k}\hat{y}_{m,n}+\lambda_X\left|\Lambda(n)\right|x_{n,k}}.\end{aligned} \tag{5}$$

By combining (5) and (4), we achieve the learning rules for an SLF-NMU-based NLF model [28]:

$$\begin{aligned}&\arg\min_{A,X}\varepsilon\left(A,X\right)\overset{SLF-NMU}{\Rightarrow}\\ &\begin{cases}a_{m,k}\leftarrow a_{m,k}\dfrac{\sum_{n\in\Lambda(m)}x_{n,k}y_{m,n}}{\sum_{n\in\Lambda(m)}x_{n,k}\hat{y}_{m,n}+\lambda_A\left|\Lambda(m)\right|a_{m,k}},\\ x_{n,k}\leftarrow x_{n,k}\dfrac{\sum_{m\in\Lambda(n)}a_{m,k}y_{m,n}}{\sum_{m\in\Lambda(n)}a_{m,k}\hat{y}_{m,n}+\lambda_X\left|\Lambda(n)\right|x_{n,k}}.\end{cases}\end{aligned} \tag{6}$$

C. Incorporation of Linear Biases

As discussed in previous studies [20], [22], incorporating linear biases makes an NLF model become more steadily during the training process, at the same time obtain higher accuracy for estimating missing entries of a target HiDS matrix. With linear bias vectors $B^{|M|}$ and $C^{|N|}$ for M and N, respectively, the objective function (3) is reformulated into

$$\begin{aligned}&\varepsilon\left(A,X,B,C\right)=\\ &\frac{1}{2}\sum_{y_{m,n}\in\Lambda}\left(\left(y_{m,n}-\hat{y}_{m,n}\right)^2+\lambda_A\sum_{k=1}^{d}a_{m,k}^2+\lambda_X\sum_{k=1}^{d}x_{n,k}^2+\lambda_B b_m^2+\lambda_C c_n^2\right),\\ &s.t.\quad a_{m,k}\geq 0,x_{n,k}\geq 0,b_m\geq 0,c_n\geq 0,\\ &\qquad\forall m\in M,n\in N,k\in\left\{1,2,\ldots,d\right\},\end{aligned} \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

where each approximation $\hat{y}_{m,n}$ to $y_{m,n}\in\Lambda$ is given by

$$\hat{y}_{m,n}=\sum_{k=1}^{d}a_{m,k}x_{n,k}+b_m+c_n. \tag{8}$$

Following the same principle of (6), we have the learning rules for a biased NLF (BNLF) model,

$$\begin{aligned}&\arg\min_{A,X,B,C}\varepsilon\left(A,X,B,C\right)\overset{SLF\text{-}NMU}{\Rightarrow}\\&\begin{cases}b_m\leftarrow b_m\sum\limits_{n\in\Lambda(m)}y_{m,n}\Big/\left(\sum\limits_{n\in\Lambda(m)}\hat{y}_{m,n}+\lambda_B\left|\Lambda\left(m\right)\right|b_m\right),\\c_n\leftarrow c_n\sum\limits_{m\in\Lambda(n)}y_{m,n}\Big/\left(\sum\limits_{m\in\Lambda(n)}\hat{y}_{m,n}+\lambda_C\left|\Lambda\left(n\right)\right|c_n\right),\\a_{m,k}\leftarrow a_{m,k}\sum\limits_{n\in\Lambda(m)}x_{n,k}y_{m,n}\Big/\left(\sum\limits_{n\in\Lambda(m)}x_{n,k}\hat{y}_{m,n}+\lambda_A\left|\Lambda\left(m\right)\right|a_{m,k}\right),\\x_{n,k}\leftarrow x_{n,k}\sum\limits_{m\in\Lambda(n)}a_{m,k}y_{m,n}\Big/\left(\sum\limits_{m\in\Lambda(n)}a_{m,k}\hat{y}_{m,n}+\lambda_X\left|\Lambda\left(n\right)\right|x_{n,k}\right).\end{cases}\end{aligned} \tag{9}$$

III. Convergence Analysis of the Non-negative Latent Factor Model

A. An Alternative Way to Achieve an SLF-NMU Algorithm

It is firstly necessary to discover the connections between an SLF-NMU algorithm and the KKT conditions [46], [52], [54] of the learning objective (6). Let $\mathrm{K}=[\kappa_{m,k}]$, $\Phi=[\phi_{n,k}]$ be Lagrangian multipliers for the non-negative constraints $a_{m,k}\geq 0$ and $x_{n,k}\geq 0$, respectively. Then the Lagrangian L for (3) is:

$$\begin{aligned}L&=\varepsilon\left(A,X\right)-Tr\left(\mathrm{K}A^{\mathrm{T}}\right)-Tr\left(\Phi X^{\mathrm{T}}\right)\\&=\varepsilon\left(A,X\right)-\sum_{m=1}^{|M|}\sum_{k=1}^{d}\kappa_{m,k}a_{m,k}-\sum_{n=1}^{|N|}\sum_{k=1}^{d}\phi_{n,k}x_{n,k},\end{aligned} \tag{10}$$

where $Tr(\cdot)$ calculates the trace of an enclosed matrix. Considering the partial derivatives of L with $a_{m,k}$ and $x_{n,k}$:

$$\begin{aligned}&\begin{cases}\dfrac{\partial L}{\partial a_{m,k}}=\sum\limits_{n\in\Lambda(m)}\left(\lambda_A a_{m,k}-x_{n,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right)-\kappa_{m,k}=0,\\\dfrac{\partial L}{\partial x_{n,k}}=\sum\limits_{m\in\Lambda(n)}\left(\lambda_X x_{n,k}-a_{m,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right)-\phi_{n,k}=0;\end{cases}\\&\Rightarrow\begin{cases}\kappa_{m,k}=\sum\limits_{n\in\Lambda(m)}\left(\lambda_A a_{m,k}-x_{n,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right),\\\phi_{n,k}=\sum\limits_{m\in\Lambda(n)}\left(\lambda_X x_{n,k}-a_{m,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right).\end{cases}\end{aligned} \tag{11}$$

Then considering the KKT conditions of (10), i.e., $\kappa_{m,k}a_{m,k}=0$, $\forall\kappa_{m,k}$, $a_{m,k}$, and $\phi_{n,k}x_{n,k}=0$, $\forall\phi_{n,k}$, $x_{n,k}$, we achieve that

$$\begin{cases}a_{m,k}\sum\limits_{n\in\Lambda(m)}\left(\lambda_A a_{m,k}-x_{n,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right)=0,\\x_{n,k}\sum\limits_{m\in\Lambda(n)}\left(\lambda_X x_{n,k}-a_{m,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right)=0.\end{cases} \tag{12}$$

With (12), we arrive at the following iterative equations which actually lead to (6),

$$\begin{aligned}&\begin{cases}a_{m,k}\sum\limits_{n\in\Lambda(m)}x_{n,k}y_{m,n}=a_{m,k}\left(\lambda_A\left|\Lambda\left(m\right)\right|a_{m,k}+\sum\limits_{n\in\Lambda(m)}x_{n,k}\hat{y}_{m,n}\right),\\x_{n,k}\sum\limits_{m\in\Lambda(n)}a_{m,k}y_{m,n}=x_{n,k}\left(\lambda_X\left|\Lambda\left(n\right)\right|x_{n,k}+\sum\limits_{m\in\Lambda(n)}a_{m,k}\hat{y}_{m,n}\right);\end{cases}\\&\Rightarrow\begin{cases}a_{m,k}\leftarrow a_{m,k}\dfrac{\sum\limits_{n\in\Lambda(m)}x_{n,k}y_{m,n}}{\lambda_A\left|\Lambda\left(m\right)\right|a_{m,k}+\sum\limits_{n\in\Lambda(m)}x_{n,k}\hat{y}_{m,n}},\\x_{n,k}\leftarrow x_{n,k}\dfrac{\sum\limits_{m\in\Lambda(n)}a_{m,k}y_{m,n}}{\lambda_X\left|\Lambda\left(n\right)\right|x_{n,k}+\sum\limits_{m\in\Lambda(n)}a_{m,k}\hat{y}_{m,n}}.\end{cases}\end{aligned} \tag{13}$$

From (10)-(13), we see that the SLF-NMU-based learning rules in an NLF model are closely connected to the KKT conditions of its learning objective.

B. Convergence Analysis of an NLF model

In this part, we theoretically analyze the convergence of an SLF-NMU learning algorithm in the following two steps.

1) Non-increasing learning objective with SLF-NMU.

In this step, we aim to prove that (3) is non-increasing with (6). To do so, we have

***Theorem* 1.** (3) is non-increasing with (6).

To prove Theorem 1, an auxiliary function [31], [53] is vital.

***Definition* 2.** $G(x, x')$ is an auxiliary function of $F(x)$ if

$$G\left(x,x'\right)\geq F\left(x\right),\ G\left(x,x\right)=F\left(x\right). \tag{14}$$

We further recall the following lemma [31], [53],

***Lemma* 1**. F keeps non-increasing with the following rule,

$$x^{t+1}=\arg\min_{x}G\left(x,x^{t}\right). \tag{15}$$

***Proof of Lemma* 1.** With Definition 2, we deduce that

$$F\left(x^{t}\right)=G\left(x^{t},x^{t}\right)\geq G\left(x^{t+1},x^{t}\right)\geq F\left(x^{t+1}\right). \tag{16}$$■

Note that we have $F(x^{t+1})=F(x^t)$ when x^t guarantees a local minimum of $G(x, x^t)$, Hence, $\nabla F(x^t)=0$ holds if F is differentiable around x^t. Hence, (16) can be extended into the following converging sequence to $x_{\min}=\arg\min_x F(x)$,

$$F\left(x_{\min}\right)\leq\cdots\leq F\left(x^{t+1}\right)\leq F\left(x^{t}\right)\leq\cdots\leq F\left(x_1\right)\leq F\left(x_0\right). \tag{17}$$

Next, we aim to achieve that (6) for is exactly consistent with that in (15) with a specifically designed G. Considering $x_{n,k}\in X$, let $F_{x_{n,k}}$ be the partial loss from $\varepsilon(A, X)$ related to $x_{n,k}$ only,

$$F_{x_{n,k}}=\frac{1}{2}\sum_{y_{m,n}\in\Lambda}\left(\left(y_{m,n}-\hat{y}_{m,n}\right)^2+\lambda_A\sum_{k=1}^{d}a_{m,k}^2+\lambda_X\sum_{k=1}^{d}x_{n,k}^2\right), \tag{18}$$

where $\hat{y}_{m,n}=a_{m,k}x_{n,k}+\sum\limits_{l=1,\neq k}^{d}a_{m,l}x_{n,l}$.

So we have the first-order and second-order derivatives of $F_{x_{n,k}}$ with respect to $x_{n,k}$,

$$F'_{x_{n,k}}=\frac{\partial\varepsilon}{\partial x_{n,k}}=\lambda_X\left|\Lambda\left(n\right)\right|x_{n,k}+\sum_{m\in\Lambda(n)}\left(-a_{m,k}\left(y_{m,n}-\hat{y}_{m,n}\right)\right), \tag{19}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

$$F''_{x_{n,k}} = \frac{\partial^2 \varepsilon}{\partial \left(x_{n,k}\right)^2} = \lambda_X \left|\Lambda(n)\right| + \sum_{m\in\Lambda(n)} \left(a_{m,k}\right)^2. \quad (20)$$

Based on (18)-(20), we achieve the following proposition.

***Proposition* 1.** The following function

$$G\left(x, x_{n,k}^{(t)}\right) = F_{x_{n,k}}\left(x_{n,k}^{(t)}\right) + F'_{x_{n,k}}\left(x_{n,k}^{(t)}\right)\left(x - x_{n,k}^{(t)}\right) + \frac{1}{2}\left(\lambda_X \left|\Lambda(n)\right| x_{n,k}^{(t)} / x_{n,k}^{(t)} + \sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n}\right)\left(x - x_{n,k}^{(t)}\right)^2 \quad (21)$$

is an auxiliary function of $F_{x_{n,k}}$.

***Proof of Proposition* 1.** With (21), $G(x,x) = F_{x_{n,k}}(x)$ holds. Then we aim to prove $G\left(x, x_{n,k}^{(t)}\right) \geq F_{x_{n,k}}(x)$. To do so, we first derive the quadratic approximation to $F_{x_{n,k}}$ at $x_{n,k}^{(t)}$.

$$F_{x_{n,k}}(x) = F_{x_{n,k}}\left(x_{n,k}^{(t)}\right) + F'_{x_{n,k}}\left(x_{n,k}^{(t)}\right)\left(x - x_{n,k}^{(t)}\right) + \frac{1}{2}F''_{x_{n,k}}\left(x_{n,k}^{(t)}\right)\left(x - x_{n,k}^{(t)}\right)^2. \quad (22)$$

By combining (20-22), we see that $G\left(x, x_{n,k}^{(t)}\right)$ is an auxiliary function of $F_{x_{n,k}}$ if the following inequality holds,

$$\frac{\lambda_X \left|\Lambda(n)\right| x_{n,k}^{(t)} + \sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n}}{x_{n,k}^{(t)}} \geq \lambda_X \left|\Lambda(n)\right| + \sum_{m\in\Lambda(n)} \left(a_{m,k}\right)^2. \quad (23)$$

Note that we have $y_{m,n}\geq 0$ according to Y's non-negativity, and $a_{m,k}\geq 0$, and $x_{n,k}\geq 0$ with SLF-NMU. So (23) is equal to

$$\sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n} \geq x_{n,k}^{(t)} \sum_{m\in\Lambda(n)} \left(a_{m,k}\right)^2. \quad (24)$$

Then we reformulate the left term of (24) as follows:

$$\begin{aligned} &\sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n} = \sum_{m\in\Lambda(n)} a_{m,k}\left(a_{m,k}x_{n,k}^{(t)} + \sum_{l=1,l\neq k}^{d} a_{m,l}x_{n,l}\right) \\ &= x_{n,k}^{(t)} \sum_{m\in\Lambda(n)} \left(a_{m,k}\right)^2 + \sum_{m\in\Lambda(n)} \left(a_{m,k} \sum_{l=1,l\neq k}^{d} a_{m,l}x_{n,l}\right) \\ &\geq x_{n,k}^{(t)} \sum_{m\in\Lambda(n)} \left(a_{m,k}\right)^2. \end{aligned} \quad (25)$$

Note that (23) holds with (25), making $G\left(x, x_{n,k}^{(t)}\right)$ be an auxiliary function of $F_{x_{n,k}}$. ■

Based on Proposition 1, we achieve the following proof.

***Proof of Theorem* 1.** Based on (15), (19) and (21), we achieve

$$\begin{aligned} &x_{n,k}^{(t+1)} = \arg\min_x G\left(x, x_{n,k}^{(t)}\right) \\ &\Rightarrow F'_{x_{n,k}}\left(x_{n,k}^{(t)}\right) + \frac{\sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n} + \lambda_X \left|\Lambda(n)\right| x_{n,k}^{(t)}}{x_{n,k}^{(t)}}\left(x - x_{n,k}^{(t)}\right) = 0 \quad (26) \\ &\Rightarrow x_{n,k}^{(t+1)} \leftarrow x_{n,k}^{(t)} \frac{\sum_{m\in\Lambda(n)} a_{m,k}y_{m,n}}{\sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n} + \lambda_X \left|\Lambda(n)\right| x_{n,k}^{(t)}}. \end{aligned}$$

Based on (26), it is clear that $F_{x_{n,k}}$ is non-increasing with (6). Naturally, (26) holds $\forall n\in N$, $m\in M$, $k\in\{1, 2, \ldots, d\}$. Hence, Theorem 1 holds. ■

Following Theorem 1, with a positive initialization, i.e., $\forall m\in M$, $n\in N$ and $k\in\{1, 2, \ldots, d\}$: $a_{m,k}^{(0)}$, $x_{n,k}^{(0)} > 0$, (3) is non-increasing when training A and X with (6). From this point of view, we have the following recursion,

$$F\left(x_{n,k}^{(0)}\right) \geq F\left(x_{n,k}^{(t)}\right) \geq G\left(x_{n,k}^{(t+1)}, x_{n,k}^{(t)}\right) \geq F\left(x_{n,k}^{(t+1)}\right) \geq 0, \quad (27)$$

which indicates that a sequence $\left\{F\left(x_{n,k}^{(t)}\right)\right\}$ is monotonically non-increasing and bounded. Hence, we deduce that

$$\lim_{t\to+\infty}\left(F\left(x_{n,k}^{(t+1)}\right) - F\left(x_{n,k}^{(t)}\right)\right) = 0. \quad (28)$$

With (28), we further have the following inference,

$$\lim_{t\to+\infty}\left|x_{n,k}^{(t+1)} - x_{n,k}^{(t)}\right| = 0. \quad (29)$$

Therefore, a sequence $\left\{x_{n,k}^{(t)}\right\}$ is bounded and convergent. Similarly, a sequence $\left\{a_{m,k}^{(t)}\right\}$ is also bounded and convergent.

2) Sequences $\left\{a_{m,k}^{(t)}\right\}$ and $\left\{x_{n,k}^{(t)}\right\}$ by SLF-NMU converge to a KKT equilibrium point of (3).

To prove it, we have

***Theorem* 2**. Sequences $\left\{a_{m,k}^{(t)}\right\}$ and $\left\{x_{n,k}^{(t)}\right\}$ by (6) converge to an equilibrium point ($a_{m,k}^{(*)}$, $x_{n,k}^{(*)}$) of $\varepsilon(A, X)$ in (3).

***Proof of Theorem* 2.** Based on (29), a sequence $\left\{x_{n,k}^{(t)}\right\}$ converges with (6). Let $X^{(*)}$ denote a stationary point of X, i.e., $0 \leq x_{n,k}^{(*)} = \lim_{t\to+\infty} x_{n,k}^{(t)} < +\infty$, $\forall n\in N$, $k\in\{1, 2, \ldots, d\}$. Thus, the following KKT conditions of (3) regarding X should be satisfied, if $X^{(*)}$ is one of its equilibrium points.

$$\begin{aligned} &\forall n\in N, m\in M, k\in\{1,\ldots,d\}: \\ &\text{(a)}\ \left.\frac{\partial L}{\partial x_{n,k}}\right|_{x_{n,k}=x_{n,k}^{(*)}} = \sum_{m\in\Lambda(n)}\left(\lambda_X x_{n,k}^{(*)} - a_{m,k}\left(y_{m,n} - \hat{y}_{m,n}\right)\right) - \phi_{n,k}^{(*)} = 0, \\ &\text{(b)}\ \phi_{n,k}^{(*)} \cdot x_{n,k}^{(*)} = 0, \\ &\text{(c)}\ x_{n,k}^{(*)} \geq 0, \\ &\text{(d)}\ \phi_{n,k}^{(*)} \geq 0. \end{aligned} \quad (30)$$

Note that following (10)-(13), Condition (a) is naturally fulfilled with (6) and (13). Thus, we actually have

$$\phi_{n,k}^{(*)} = \sum_{m\in\Lambda(n)}\left(\lambda_X x_{n,k}^{(*)} - a_{m,k}\left(y_{m,n} - \hat{y}_{m,n}\right)\right). \quad (31)$$

Thus, we focus on Conditions (c)-(d). We start with constructing $\theta_{n,k}^{(t)}$,

$$\theta_{n,k}^{(t)} = \frac{\sum_{m\in\Lambda(n)} a_{m,k}y_{m,n}}{\lambda_X \left|\Lambda(n)\right| x_{n,k}^{(t)} + \sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n}}. \quad (32)$$

Obviously, (32) is bounded by non-negative $a_{m,k}$ and $y_{m,n}$:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

$$0 \le \theta_{n,k}^{(*)} = \lim_{t\to+\infty} \theta_{n,k}^{(t)} = \frac{\sum_{m\in\Lambda(n)} a_{m,k} y_{m,n}}{\lambda_X \left|\Lambda(n)\right| x_{n,k}^{(*)} + \sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n}}. \quad (33)$$

Hence, the update rule of $x_{n,k}$ can be rewritten with SLF-NMU,

$$x_{n,k}^{(t+1)} = x_{n,k}^{(t)}\theta_{n,k}^{(t)}. \quad (34)$$

By combining (29) and (34), we have

$$\lim_{t\to+\infty}\left|x_{n,k}^{(t+1)} - x_{n,k}^{(t)}\right| = 0 \Rightarrow x_{n,k}^{(*)}\theta_{n,k}^{(*)} - x_{n,k}^{(*)} = 0. \quad (35)$$

Note that following the update rule (13), $x_{n,k}^{(*)}$ is greater than or equal to zero with a non-negatively initial hypothesis. Hence, we have the following inferences.

a) **When** $x_{n,k}^{(*)}>0$. Based on (32) and (35), we have:

$$\begin{aligned} &x_{n,k}^{(*)}\theta_{n,k}^{(*)} - x_{n,k}^{(*)} = 0, x_{n,k}^{(*)} > 0 \Rightarrow \theta_{n,k}^{(*)} = 1 \\ &\Rightarrow \lambda_X \left|\Lambda(n)\right| x_{n,k}^{(*)} + \sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n} - \sum_{m\in\Lambda(n)} a_{m,k} y_{m,n} = 0. \end{aligned} \quad (36)$$

Combining (31) and (36), we obtain the Condition (b) in (30),

$$\begin{aligned} &\phi_{n,k}^{(*)} = \lambda_X \left|\Lambda(n)\right| x_{n,k}^{(*)} + \sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n} - \sum_{m\in\Lambda(n)} a_{m,k} y_{m,n} = 0 \\ &\Rightarrow \phi_{n,k}^{(*)} \cdot x_{n,k}^{(*)} = 0. \end{aligned} \quad (37)$$

Meanwhile, when $\phi_{n,k}^{(*)}=0$ and $x_{n,k}^{(*)}>0$, Conditions (c) and (d) are naturally fulfilled. Hence, (30) is fulfilled when $x_{n,k}^{(*)}>0$.

b) **When** $x_{n,k}^{(*)}=0$. Note that conditions (b) and (c) in (30) are naturally fulfilled in this case. Hence, we want to justify Condition (d). To do so, we reformulate $x_{n,k}^{(*)}$ into

$$x_{n,k}^{(*)} = x_{n,k}^{(0)} \lim_{t\to+\infty} \prod_{r=1}^{t} \theta_{n,k}^{(r)}. \quad (38)$$

Based on (38) we further have the following inferences,

$$\begin{aligned} &x_{n,k}^{(0)} > 0, x_{n,k}^{(0)} \lim_{t\to+\infty} \prod_{r=1}^{t} \theta_{n,k}^{(r)} = x_{n,k}^{(*)} = 0 \Rightarrow \lim_{t\to+\infty} \prod_{r=1}^{t} \theta_{n,k}^{(r)} = 0 \\ &\Rightarrow \lim_{t\to+\infty} \theta_{n,k}^{(t)} = \theta_{n,k}^{(*)} = \frac{\sum_{m\in\Lambda(n)} a_{m,k} y_{m,n}}{\lambda_X \left|\Lambda(n)\right| x_{n,k}^{(*)} + \sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n}} \le 1 \\ &\Rightarrow \phi_{n,k}^{(*)} = \lambda_X \left|\Lambda(n)\right| x_{n,k}^{(*)} + \sum_{m\in\Lambda(n)} a_{m,k}\hat{y}_{m,n} - \sum_{m\in\Lambda(n)} a_{m,k} y_{m,n} \ge 0. \end{aligned} \quad (39)$$

Hence, (30) is also fulfilled when $x_{n,k}^{(*)}>0$. Analogously, we can prove that sequence $\left\{a_{m,k}^{(t)}\right\}$ also converges to an stable equilibrium point of (3). Thus, Theorem 2 stands.■

By combining Theorems 1 and 2, we have proven that an SLF-NMU-based NLF model converges to a KKT equilibrium point of its objective. It should be pointed out a positive initialization of LFs is helpful in achieving an optimal solution, which is consistent with [55].

C. Convergence Analysis of a BNLF model

1) Non-increasing learning objective with biased SLF-NMU.

We firstly present the following theorem.

Theorem 3. (7) is non-increasing with (9).

Note that for Theorem 3, we only have to analyze the partial loss with B and C since conditions of A and X are highly similar with that in an NLF model as shown in Theorem 1. Considering $b_m \in B$, let $\tilde{F}_{b_m}$ be the partial loss of $\varepsilon(A, X, B, C)$ related to b_m only. Then we derive the first-order and second-order derivatives of $\tilde{F}_{b_m}$ with respect to b_m,

$$\tilde{F}'_{b_m} = \lambda_B \left|\Lambda(m)\right| b_m - \sum_{n\in\Lambda(m)} \left(y_{m,n} - \hat{y}_{m,n}\right). \quad (40)$$

$$\tilde{F}''_{b_m} = \left(1+\lambda_B\right)\left|\Lambda(m)\right|. \quad (41)$$

Based on (7), (40) and (41), we present Proposition 2.

Proposition 2. The following function

$$\begin{aligned} \tilde{G}\left(b, b_m^{(t)}\right) = &\tilde{F}_{b_m}\left(b_m^{(t)}\right) + \tilde{F}'_{b_m}\left(b_m^{(t)}\right)\left(b - b_m^{(t)}\right) \\ &+ \frac{1}{2}\left(\sum_{n\in\Lambda(m)} \left(\hat{y}_{m,n} + \lambda_B b_m^{(t)}\right) \Big/ b_m^{(t)}\right)\left(b - b_m^{(t)}\right)^2, \end{aligned} \quad (42)$$

is an auxiliary function of $\tilde{F}_{b_m}$.

Note that the proof of Proposition 2 is provided in the supplementary file of this paper. With it we prove Theorem 3, whose proof is also provided in the supplementary file.

Following Theorems 1 and 3, with a positive initialization, i.e., $\forall m\in M$, $n\in N$, and $k\in\{1, 2, \ldots, d\}$: $a_{m,k}^{(0)}$, $x_{n,k}^{(0)}$, $b_m^{(0)}$ and $c_n^{(0)}>0$, (7) is non-increasing when training A, X, B and C with (9). From this point of view, we have the following recursion:

$$\tilde{F}\left(b_m^{(0)}\right) \ge \tilde{F}\left(b_m^{(t)}\right) \ge \tilde{G}\left(b_m^{(t+1)}, b_m^{(t)}\right) \ge \tilde{F}\left(b_m^{(t+1)}\right) \ge 0, \quad (43)$$

which indicates that a sequence $\left\{\tilde{F}\left(b_m^{(t)}\right)\right\}$ is non-increasing and bounded. Hence, we deduce that

$$\lim_{t\to+\infty}\left|\tilde{F}\left(b_m^{(t+1)}\right) - \tilde{F}\left(b_m^{(t)}\right)\right| = 0. \quad (44)$$

Based on (49), we further yield the following inference:

$$\lim_{t\to+\infty}\left|b_m^{(t+1)} - b_m^{(t)}\right| = 0. \quad (45)$$

Therefore, a sequence $\left\{b_m^{(t)}\right\}$ is convergent and bounded. Similarly, a sequence $\left\{c_n^{(t)}\right\}$ is also convergent and bounded.

2) Sequences $\left\{a_{m,k}^{(t)}\right\}$, $\left\{x_{n,k}^{(t)}\right\}$, $\left\{b_m^{(t)}\right\}$ and $\left\{c_n^{(t)}\right\}$ by a biased SLF-NMU converge to a KKT equilibrium point of (7).

To prove it, we have the following theorem.

Theorem 4. Sequences $\left\{a_{m,k}^{(t)}\right\}$, $\left\{x_{n,k}^{(t)}\right\}$, $\left\{b_m^{(t)}\right\}$ and $\left\{c_n^{(t)}\right\}$ by (9) converge to an equilibrium point of (7).

Note that its proof is provided in the supplementary file. By analogy, we prove that sequence $\left\{c_n^{(t)}\right\}$ also converges to an equilibrium point of (7).

Hence, by combining Theorems 1-4, we arrive at the conclusion that a BNLF model converges to a KKT equilibrium point of its objective with a biased SLF-NMU algorithm.

IV. Experimental Results and Analysis

A. General Settings

Evaluation Metrics. Considering the need for recovering

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

full links among involved entities [1]-[12] when addressing an HiDS matrix in real applications, the main purpose is to estimate its missing entries. Hence, we adopt the root mean squared error (RMSE) as an evaluation metric to test the reconstruction ability of tested models on an HiDS matrix [3], [11], [12], [28], [47],

$$\text{RMSE}=\sqrt{\frac{1}{|\Psi|}\left(\sum_{r_{u,v}\in\Psi}\left(r_{u,v}-\hat{r}_{u,v}\right)^2\right)}, \tag{46}$$

where $|\cdot|$ calculates a set's cardinality, Ψ denotes the validation set that is disjoint with the given set Λ, and $\hat{r}_{u,v}$ stands for an estimation for $r_{u,v}\in\Psi$ by a tested model, respectively.

Thus, the convergence behaviors of a tested model are reflected by three factors, i.e., a) its training RMSE, b) its absolute reduction of the learning objective (2), and c) its LF distributions during/after the learning process. All factors are expected to be stable when a model converges. All empirical tests are conducted on a server with a 2.7 GHz E5-2680 V4 CPU and 256 GB RAM, and all models are implemented in JAVA SE 8U131.

Tested Models. As shown in Table I, three models, i.e., NLF [3], BNLF [28], and WNMF [48] are tested and compared.

TABLE I
TESTED MODELS IN OUR EXPERIMENTS

No.	Name	Description
M1	NLF	Original NLF model relying on the SLF-NMU algorithm proposed in [28].
M2	BNLF	Biased NLF model by SLF-NMU proposed in [3].
M3	WNMF	A weighted NMF model proposed in [48]. It can address an HiDS matrix.

TABLE II
DETAILS OF EXPERIMENTAL DATASETS

No.	Name	$\|\Lambda\|+\|\Psi\|$	$\|M\|$	$\|N\|$	Source
D1	ML20M	20,000,263	138,493	26,744	MovieLens [44]
D2	NetFlix	100,480,507	2,649,429	17,770	NetFlix [45]
D3	Douban	16,830,839	129, 490	58, 541	Douban [56]
D4	Flixter	8,196,077	147,612	48,794	Flixter [61]
D5	Epinion	13,668,320	120,492	755,760	Truslet website [62]
D6	Dating	17,359,346	135,359	168,791	LibimSeTi [63]
D7	EM	2,811,718	72,916	1,628	EachMovie
D8	Network	175,180	134,007	134,007	Authors' affiliation
D9	PICE	5,778,268	15,752	15,752	STRING [64]
D10	PIPA	1.559.616	5,565	5,565	STRING [64]

Datasets. As recorded in Table II, ten real HiDS matrices collected from industrial applications are adopted. We employ the following settings for objective results.

a) Each dataset is divided into five disjoint and equally-sized subsets randomly. We adopt the 80%-20% given-validation settings and five-fold cross-validations in all the experiments, i.e., each time we choose four subsets as the given set Λ to build a model to predict missing data and the remaining one as the validation set Ψ to achieve its validation RMSE. Then we sequentially repeat this process five times to conduct five-fold cross-validation for the final results.

b) When building a model, the given set Λ is further split into two disjoint training and testing subsets with the ratio of 90%: 10%. The model is trained on the training set and tested on the testing set to determine if its training process should be terminated. The training of a model ends if it converges, i.e., its error difference between two successive epochs is less than 10^{-5}, or its iteration count exceeds a preset threshold, i.e., 1000.

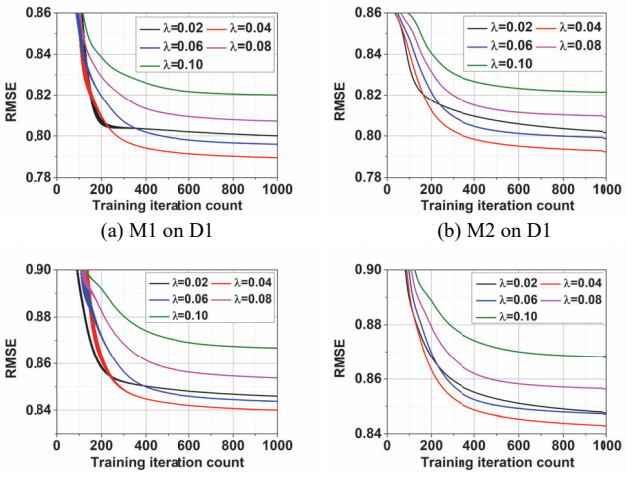
c) Making $\lambda_A=\lambda_X=\lambda_B=\lambda_C=\lambda$ for M1-2 following [3], [28], [47] (it should be pointed out that tuning them separately can slightly improve the accuracy of M1-2 but is very time-consuming).

d) Note that according to [3]-[5], [7]-[12], [28], [47], an NLF model's performance can be affected by its initial hypothesis. Hence, on each testing case, we initialize M1-3 with the same randomly-generated and non-negative arrays for eliminating the influence of an initial hypothesis.

By doing so, we make all involved models built on the given set Λ only. For M1-2, the hyper-parameter λ is tuned on Λ. For M1-3, the training termination is guaranteed on Λ. Each model's reconstruction ability is validated on Ψ separately, whose information is never referred by its training process.

B. *Impact of λ*

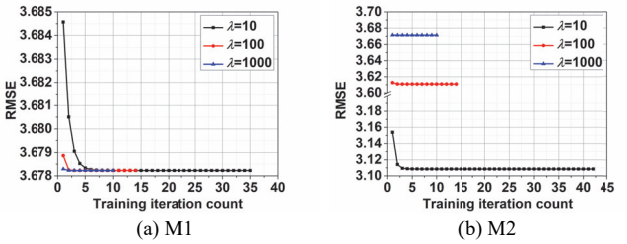
As discussed in prior studies [28], [47], M1-2's performance can be improved significantly via incorporating the Tikhonov regularization. Note that the hyper-parameter λ actually controls the regularization effects in their objective function. They may suffer overfitting as λ becomes too small and underfitting as λ becomes too large. These phenomena are connected with the convergence behaviors of M1-2 and should be carefully validated. We have set λ in the scale of [0.01, 0.20] for demonstrating the former, and set λ=10, 100, and 1000 for demonstrating the latter.



(a) M1 on D1 (b) M2 on D1

(c) M1on D2 (d) M2 on D2

Fig. 1. RMSE decreasing curves of M1-2 on D1-2 as λ in [0.02, 0.10].



(a) M1 (b) M2

Fig. 2. RMSE decreasing curves of M1-2 on D1 as λ=10, 100, 1000.

Fig. 1 depicts the RMSE decreasing curves for M1-2 on D1-2 as λ in [0.02, 0.10]. Fig. 3 depicts the RMSE decreasing curves for M1-2 on D1 as λ=10, 100 and 1000. Similar phenomena are also encountered on D3-10. Moreover, the optimal λ for M1-2

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

on D1-10 are summarized in Table III. Note that these results are achieved on Λ of D1-10 only, following the settings mentioned in Section IV(A). From them, we find that:

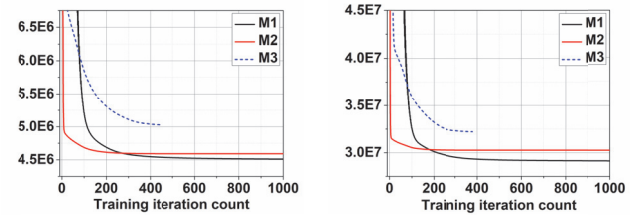
a) **M1-2 suffers overfitting as λ becomes too small, which is reflected by their accuracy loss.** From Figs. 1-2, we see that M1-2's RMSE as λ=0.02 is obviously higher than that achieved as λ=0.04. More specifically, M1's RMSE is 0.8001 when λ=0.02, and 0.7895 when λ=0.04. Thus, inappropriately small λ makes it suffer an accuracy loss at 1.34%. Considering M2, its RMSE is 0.8021 as λ=0.02 and 0.7924 as λ=0.04. It turns out that too small λ makes it suffer overfitting, leading to an accuracy loss at 1.20%.

b) **M1-2 suffer underfitting as λ becomes too large, which is also reflected by their accuracy loss.** When λ increases over 0.04, M1-2 suffer accuracy loss on D1-2, as shown in Figs. 1-2, which is caused by underfitting. From Fig. 2, we clearly see that as λ increases to 10-1000, M1-2 can still guarantee the convergence to a stable equilibrium point of the objective function. However, they converge to mostly minimize the regularization terms instead of minimizing the generalized error to fit the training data, thereby suffering significant accuracy loss. For example, as depicted in Fig. 2(b), M2's RMSE at 3.6714 when λ=1000, which is about 4.65 times that of 0.7895 when λ=0.04.

TABLE III
OPTIMAL HYPER-PARAMETER λ FOR M1-2 ON D1-10.

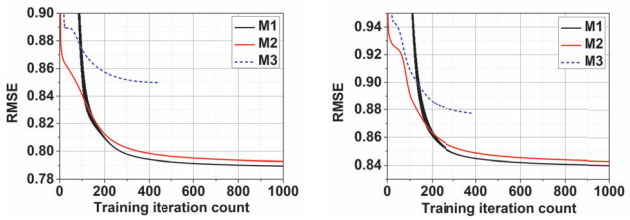
Dataset / Model	D1	D2	D3	D4	D5
M1	0.04	0.04	0.06	0.06	0.10
M2	0.04	0.04	0.06	0.06	0.08

Dataset / Model	D6	D7	D8	D9	D10
M1	0.18	0.02	0.12	0.01	0.02
M2	0.18	0.015	0.16	0.01	0.02



(a) D1 (b) D2

Fig. 3. Absolute reduction of learning objective (2) by M1-3 on D1-2.



(a) D1 (b) D2

Fig. 4. RMSE decreasing curves of M1-3 on D1-2.

C. Convergence Behaviors

In this part of the experiments, we validate the convergence behaviors of M1-2. In addition, we include M3 for comparison. On Λ, each tested model is expected to stabilize its a) Training RMSE, b) absolute reduction of learning objective (2), and c) LF distribution.

Fig. 3 depicts the absolute reduction of learning objective (2) by M1-3 on D1-2. The RMSE decreasing curves of M1-3 on D1-2 are depicted in Fig. 4. Figs. S1-2 in the Supplementary File of this paper depict LF distributions of M1-2 after the initialization, 1st, 10th, 50th, 100th, 200th, 400th, 600th, 800th, and last iteration during the training process on D1. Note that we present these results only for a concise report. However, highly similar phenomena are encountered in other testing cases. From these results, we have the following findings:

a) **M1-2 converge better than M3 does on an HiDS matrix.** From Figs. 3-4, we see that M3 cannot reach a good local optimum, and commonly ends with fewer iterations and higher RMSE when compared with M1-2. For instance, according to Fig. 4(a), on D1 it consumes 441 iterations to obtain the RMSE at 0.8489. In comparison, M1 consumes 1000 iterations with the RMSE at 0.7895, and M2 also consumes 1000 iterations with the RMSE at 0.7924. Both NLF and BNLF models converge with much lower RMSE than a WNMF does. Mover, according to Fig. 3(a), we see that the absolute reduction of learning objective (2) achieved by M3 is also much less than that achieved by M1-2 on D1, indicating that M1-2 can better approximate the learning objective than M3 does. The same situations are also found on D2-10.

b) **M1-2 enable their LF-distributions to converge to a steady-state.** As shown in Fig. S1-2, M1-2's LFs initially distribute uniformly in the range of [0, 0.005] with a random hypothesis. However, their LF values and distributions change sharply during the first 200 iterations, and then change very slow during subsequent iterations, and finally converge to an equilibrium state. Such a phenomenon again supports our convergence proof.

D. Reconstruction Ability

In this part of the experiments, we validate M1-3's RMSE on Ψ of each dataset to see their reconstruction ability on an HiDS matrix. RMSEs of M1-3 on D1-10 are recorded in Table IV. Then we conduct the Friedman test [65] on these results to fully understand their significance.

The average ranks with each model's RMSE in table IV are computed in Table V. Let r_i^j denote the rank of the jth one of p tested models on the ith one of S testing cases, a Friedman test compares the average rank of each tested model calculated by $H_j = \sum_{j=1}^{S} r_i^j / S$. Then the Friedman value is computed as

$$\chi_F^2 = \frac{12S}{p(p+1)}\left[\sum_j H_j^2 - \frac{p(p+1)^2}{4}\right]. \tag{47}$$

Based on (47), the testing score is given by

$$F_F = \frac{(S-1)\chi_F^2}{S(p-1)-\chi_F^2}, \tag{48}$$

which is distributed according to the F-distribution with $p-1$ and $(p-1)(S-1)$ degrees of freedom [65]. Note that if F_F is larger than a given critical value α, the null hypothesis which claims the equivalence of all tested models can be rejected.

In our experiments, three models are tested on ten datasets. Hence, we have p=3 and S=10. Thus, F_F has (2, 18) degrees of freedom and the corresponding critical value is 3.55 for α=0.05.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

Hence, the null hypothesis can be rejected if the testing score is higher than 3.55. By substituting the average ranks of M1-3 in Table V into (47) and (48), we achieve that F_F=27.0, which is much higher than 3.55. Hence, M1-3 are significantly different with confidence at 95%.

TABLE IV
RMSE OF M1-3 ON Ψ OF D1-10.

Dataset / Model	D1	D2	D3	D4	D5
M1	**0.7901**	**0.8389**	0.7258	0.9133	0.6101
M2	0.7930	0.8417	**0.7177**	**0.9114**	**0.5982**
M3	0.8498	0.8773	0.7655	0.9708	0.6481

Dataset / Model	D6	D7	D8	D9	D10
M1	1.8517	**0.2316**	2.1359	**0.1081**	**0.1409**
M2	**1.8442**	0.2335	**2.0847**	0.1126	0.1470
M3	2.0781	0.2611	3.2824	0.1308	0.1667

TABLE V
AVERAGE RANKS OF M1-3 W.R.T RMSE.

Models	M1	M2	M3
Average Rank	1.5	1.5	3.0

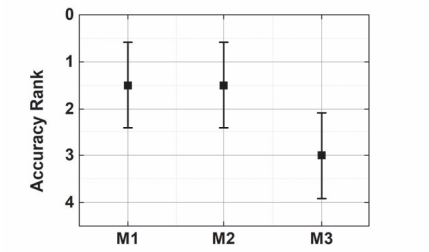


Fig. 5. Results of Nemenyi analysis

To further identify M1-3's performance, we adopt the Nemenyi test [65]. Two models are significantly different if the difference between their performance ranks is larger than the critical difference value [65] given by

$$CD = q_\alpha \sqrt{\frac{p(p+1)}{6S}}. \quad (49)$$

Note that q_α in (49) is 2.052 with the critical level α=0.1 as p=3 in our case [65]. By substituting p=3, S=10, and $q_{0.1}$=2.052 into (49), we obtain that CD=0.918. Hence, two models are significantly different in reconstruction ability on an HiDS matrix with the confidence of 90%, when their rank different value is larger than 0.918. The Nemenyi analysis results according to Tables IV and V are shown in Fig. 5. From them, we see that both M1 and M2, i.e., NLF and BNLF models, significantly outperform M3, i.e., a WNMF model, in terms of reconstruction ability on an HiDS matrix. In addition, M1 and M2 do not have a significance difference in the level of RMSE from each other.

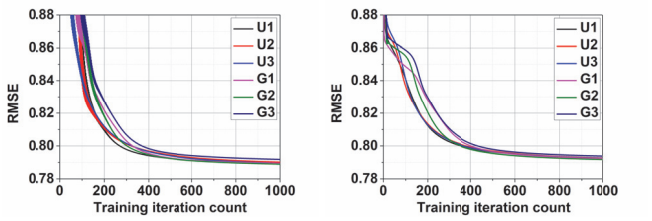
E. Initialization Strategies

As revealed before [7]-[12], [26]-[31], [58], the performance of an NLF model is sensitive to its initial hypothesis. However, how to select an appropriate initial hypothesis for it remains an open issue. According to prior research [7]-[12], [26]-[31], [58], it is commonly accepted to randomly generate an initial hypothesis in the non-negative field of real numbers. Hence, it is the distribution of these randomly generated initializations that affects the performance of a resultant model. For validating its effects, we have conducted tests with six different initialization distributions for M1-2. The details of tested initialization distributions are summarized in Table VI.

TABLE VI
DETAILS OF TESTED DISTRIBUTIONS FOR GENERATING INITIAL HYPOTHESES OF M1-2. NOTE THAT F DENOTES ALL OF THEIR DESIRED LFS. FOR M1, F=$\{A, X\}$, FOR M2, F=$\{B, C, A, X\}$.

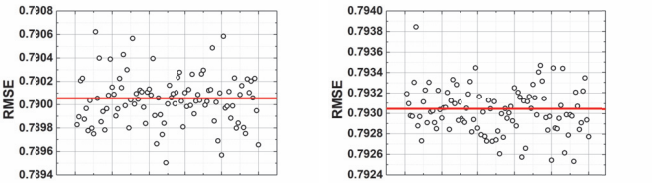
No.	Distribution	No.	Distribution
U1	F~U(0, 0.005)	G1	F~N(0.005, 1e-7)
U2	F~U(0, 0.1)	G2	F~N(0.1, 1e-5)
U3	F~U(0, 1.0)	G3	F~N(1.0, 1e-4)

The RMSE decreasing curves of M1-2 with different initialization distributions on D1 is depicted in Fig. 6. The RMSE of M1-2 when F~U(0, 0.005) in 100 independent tests on D1 is depicted in Fig. 7. Note that similar results can be observed on D2-10. Table VII summarizes the RMSE of M1-2 on D1-3 with different initial hypotheses. From them, we see that:



(a) M1 (b) M2

Fig. 6. RMSE decreasing curves of M1 and M2 on D1 with different initial hypotheses.



(a) M1 (b) M2

Fig. 7. RMSE of M1-2 when F~U(0, 0.005) in 100 independent tests on D1.

a) **The prediction accuracy of an SLF-NMU-based NLF model is slightly affected by its initial hypothesis.** From Fig. 6, we find that the convergence curves of M1-2 on D1 varies as their initialization distribution changes. However, they will always converge to an equilibrium point regardless of the initialization strategies. Moreover, from Fig. 7 we can only observe slight accuracy fluctuation of M1-2 by different initial hypotheses randomly generated from the same distribution. The fluctuation ratio is limited in the range of [-0.25%, 0.25%] according to our results.

b) **Appropriate initialization strategies can enable a more accurate model.** If we aim at achieving performance gain via improving M1-2's initial hypothesis, the following two strategies can be considered: 1) choosing the initialization distribution wisely. According to Table VII, an NLF model with a Gaussian hypothesis mostly outperforms one with a Uniform hypothesis, which demonstrates that a Gaussian initialization distribution can probably improve an NLF model's prediction accuracy. But in practice, it is still necessary to adopt warming up tests with initialization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

distributions on a probe set; and 2) building an ensemble. Since different initialization distributions can achieve diversified NLF models as shown in Fig. 6, they can form an efficient ensemble [57], [58] which can outperform any of its base models to achieve stably high accuracy.

TABLE VII
RMSE OF M1-2 ON D1-3 WITH DIFFERENT INITIAL HYPOTHESES

		Uniform Hypothesis			Gaussian Hypothesis		
		U1	U2	U3	G1	G2	G3
D1	M1	0.7901	0.7905	**0.7894**	0.7901	0.7899	0. 7930
	M2	0.7930	0.7930	0.7925	**0.7923**	0.7925	0. 7947
D2	M1	0.8389	0.8389	0.8392	**0.8387**	**0. 8387**	0. 8392
	M2	0.8417	0.8419	0.8416	0.8414	**0. 8413**	0. 8417
D3	M1	0.7258	0.7226	**0.7089**	0.7423	0.7193	0.7359
	M2	0.7177	0.7158	**0.7083**	0.7177	0. 7135	0. 7283

F. Summary

We summarize from the experimental results:

a) An SLF-NMU-based NLF/BNLF model converges to a stable equilibrium point on an HiDS matrix;
b) An SLF-NMU-based NLF/BNLF model converges better than an NMF model on an HiDS matrix. Its reconstruction ability for an HiDS matrix's unknown data is also better than that of an NMF Model; and
c) An SLF-NMU-based NLF/BNLF model's regularization coefficient and initialization hypotheses should be chosen wisely to improve its performance.

V. DISCUSSIONS

A. Connections between an NLF and an NMF Model

An NMF model is designed for full matrices, which are mostly seen as images in the area of computer vision. Initially, Lee and Seung [30], [31] adopt it to extract local features from a non-negative matrix representing an image. Subsequently, many researchers investigate it and propose various NMF extensions [29]-[36], [48]-[52]. To achieve the desired non-negative feature matrices, a non-negative multiplicative update (NMU) algorithm is commonly adopted. With it, if $Y^{|M|\times|N|}$ is a full matrix, then it extracts features $A^{|M|\times d}$ and $X^{|N|\times d}$ as

$$a_{m,k} \leftarrow a_{m,k}\frac{(YX)_{m,k}}{\left(AX^{\mathrm{T}}X\right)_{m,k}}, x_{n,k} \leftarrow x_{n,k}\frac{\left(Y^{\mathrm{T}}A\right)_{n,k}}{\left(XA^{\mathrm{T}}A\right)_{n,k}}. \quad (50)$$

As a matter of fact, other algorithms like the projected gradient descent [59], projected alternating least squares [60] are also proposed to train an NMF model. Nonetheless, an NMU algorithm does not make truncations during the training process, making it more suitable to represent the natural structures such as node clusters hiding in the original matrix. Meanwhile, it is implemented through making the learning rate adaptive to achieve the multiplicative form, i.e., its learning rate is self-adaptive. Hence, an NMF model frequently adopts an NMU algorithm. The processing flow of such a model is shown in Fig. S3 in the Supplementary File. According to (50), its computational complexity is $\Theta(|M|\times|N|\times k\times t)$ with t being training iteration count, and storage complexity is $\Theta(|M|\times|N|)$.

However, an NMF model cannot handle an HiDS matrix directly since the products YX and $Y^{\mathrm{T}}A$ are intractable with Y being incomplete. As discussed in [28], [48], an NMF model can be adjusted to fit an incomplete Y by integrating an indicator matrix $W^{|M|\times|N|}$ into (50) to achieve a weighted NMF (WNMF) model. In it, the indicator matrix's element is set at one if the corresponding element in Y is known, and zero otherwise. Thus, WNMF can extract non-negative features from an incomplete Y with an NMU algorithm:

$$a_{m,k} \leftarrow a_{m,k}\frac{\left((W\circ Y)X\right)_{m,k}}{\left(AX^{\mathrm{T}}X\right)_{m,k}}, x_{n,k} \leftarrow x_{n,k}\frac{\left((W\circ Y)^{\mathrm{T}}A\right)_{n,k}}{\left(XA^{\mathrm{T}}A\right)_{n,k}}; \quad (51)$$

where the operator $\circ$ calculates the Hardamad product of two matrices. Thus, the products $(W\circ Y)X$ and $(W\circ Y)^{\mathrm{T}}A$ are solvable when Y is HiDS. However, the computational and storage costs of WNMF are respectively $\Theta(|M|\times|N|\times k\times t)$ and $\Theta(|M|\times|N|)$, which are the same as those of NMF. It is compatible with existing NMF training algorithms like an NMF algorithm, however, it suffers unnecessarily high costs on an HiDS matrix. Its processing flow is in Fig. S4 in the Supplementary File.

In comparison, an NLF model is specifically designed for an HiDS matrix. With an SLF-NMU algorithm, its computational and storage costs are respectively $\Theta(|\Lambda|\times k\times t)$ and $\Theta(\max\{|\Lambda|, (|M|+|N|)\times d\})$. Thus, we have the following inferences:

$$\frac{T_{NLF}}{T_{NMF}} = \frac{|\Lambda|\times k\times t_{NLF}}{|M|\times|N|\times k\times t_{NMF}} \approx \frac{|\Lambda|}{|M|\times|N|}$$
$$\frac{S_{NLF}}{S_{NMF}} = \frac{max\left\{|\Lambda|,\ (|M|+|N|)\times d\right\}}{|M|\times|N|} = max\left\{\frac{|\Lambda|}{|M|\times|N|},\ d\left(\frac{1}{|M|}+\frac{1}{|N|}\right)\right\} \quad (52)$$

Given that $|\Lambda|\ll|M|\times|N|$, i.e., the known data of an HiDS matrix are far less than its unknown ones (e.g., the known data of the NetFlix matrix take 0.21% of its all entries only), an SLF-NMU-based NLF model's computational and storage costs are much lower than those of an NMF model in theory. Hence, an NLF model is far more efficient than an NMF-type model on an HiDS matrix. Its processing flow is depicted in Fig. S5 in the Supplementary File. From this point of view, it is much suitable to address an HiDS matrix than an NMF model does on the premise that it converges well, which is guaranteed by the convergence analysis given in this paper.

On the other hand, it is well known that in the area of recommender systems or social network services, an HiDS matrix is far more frequently encountered than a full one. Therefore, we'd prefer to take an NMF model as a special case for an NLF model owing to the following reasons:

a) In our concerned big data-related Web-applications like a recommender system, the observed data are commonly incomplete, making an HiDS matrix be far more frequently encountered than a full matrix. Actually, a full matrix can be considered as an HiDS matrix whose data density is 100% in such a scenario, indicating that the interactions among users and items are fully observed (or quantized by the system, which is also an ultimate objective for a recommender system); and
b) Note that owing to its data-density-oriented learning objective and algorithm, i.e., an SLF-NMU algorithm, an NLF model can address an HiDS matrix whose data density can be an arbitrary number in the (0%, 100%] interval. In other words, it can address a matrix with/without missing data. In comparison, an NMF model is defined on a full matrix. Thus, it can only handle a matrix without missing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNNLS.2020.2990990, IEEE Transactions on Neural Networks and Learning Systems

data (or transmit a HiDS target into a full one like in a WNMF model).

Nonetheless, in other fields like computer vision, full matrices representing normal images are much more than incomplete ones. Under such circumstances, an NLF model can be considered as a special case of an NMF model. From this point of view, we clearly see that although these two kinds of models are closely connected, they are targeting at different problems. It is important to make an appropriate choice for a specific problem. From the theoretical and empirical studies of this paper, it is efficient to adopt NLF on an HiDS matrix.

B. Theoretical Achievements of This Study

Since both NMF and NLF models are bi-linear, an auxiliary function-based method can be adopted to show that both NMU for NMF and SLF-NMU for NLF make their loss functions non-increasing. Nonetheless, the theoretical achievements of this work stand in two aspects: 1) an SLF-NMU is single LF-dependent, making its mathematical expression very different from that of an NMU algorithm. Meanwhile, an HiDS matrix's data density is low and distribution is highly imbalanced. The convergence analysis in this paper innovatively shows that an SLF-NMU learning algorithm can enable an NLF model to converge on an arbitrary matrix in spite of its sparsity and imbalanced data distribution; 2) its ability to converge to a KKT equilibrium point is also proved on an HiDS matrix, which is also not seen in prior studies.

C. Local or Global Convergence

It should be mentioend that the objective function of an NLF/BNLF model is constrained and non-convex. Meanwhile, according to (6), an SLF-NMU-based learning algorithm actually implements an additive gradient descent-based training process with carefully-selected learning rates to guarantee the non-negativity of a resultant model. In other words, its achieved stationary point (whose properties are verified according to the previously provided theoretical studies) is a first-order one on a non-convex problem. According to a prior study [25], such a first-order stationary point can be a global optimum, local optimum, or saddle point. How to identify its more specific properties is very challenging (which still remains unveiled in the optimization community according to [25]). We plan to address this problem in our future work.

VI. Conclusions

In this paper, we rigorously prove that an NLF/BNLF model defined on an HiDS matrix converges to a KKT equilibrium point of its objective with an SLF-NMU-based learning algorithm. Note that the proof consists of two steps, a) proving that its learning objective goes non-increasing with SLF-NMU by constructing a specifically-designed auxiliary function; and b) proving that its parameter learning sequences finally converge to a stable equilibrium point with SLF-NMU by analyzing the KKT conditions of its learning objective.

Note that this study conducts the convergence analysis on an objective relying on the Euclidean distance. However, the same principle also applies equally to an NLF model depending on other kinds of loss functions. With it, an SLF-NMU-based NLF model's convergence characteristics are theoretically justified, which are also supported by the empirical results on ten HiDS matrices from real applications.

As revealed in previous studies [3], the parallelization of an SLF-NMU algorithm can be implemented via a properly-designed distributed computing framework. In this case, can it still converge to a stable equilibrium point? We will answer this question in the future.

References

[1] 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, Apr. 2011.

[2] H. Karimi, J. Nutini, and M. Schmidt, "Linear Convergence of Gradient and Proximal-Gradient Methods under the Polyak-Łojasiewicz Condition," in *Proc. of the 2016 Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, Springer, Riva del Garda, Italy, Sept. 2016, pp. 795-811.

[3] 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, Mar. 2016.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE Computer,* vol. 42, no. 8, pp. 30-37, Aug. 2009.

[5] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient Descent Only Converges to Minimizers," *JMLR Workshop and Conference Proceedings*, vol. 49, pp. 1-12, Aug. 2016.

[6] R. Sun and Z. Q. Luo, "Guaranteed Matrix Completion via Non-convex Factorization," *IEEE Trans. on Information Theory*, vol. 62, no. 11, pp. 6535–6579, Nov. 2016.

[7] X. Luo, Z. H. You, M. C. Zhou, S. Li, H. Leung, Y. N. Xia, and Q. S. Zhu, "A Highly Efficient Approach to Protein Interactome Mapping Based on Collaborative Filtering Framework," *Scientific Reports*, vol. 5, no. 7702, pp. 1-10, Jan. 2015.

[8] Q. Zheng and J. Lafferty, "Convergence Analysis for Rectangular Matrix Completion Using Burer-Monteiro Factorization and Gradient Descent," *arXiv preprint arXiv:1605.07051*, Nov. 2016.

[9] Z. Ghahramani, "Probabilistic Machine Learning and Artificial Intelligence," *Nature,* vol. 521, no. 7553, pp. 452-459, May. 2015.

[10] M. Hofree, J. P. Shen, H. Carter, A. Gross, and T. Ideker, "Network-based Stratification of Tumor Mutations," *Nature Methods,* vol. 10, no. 11, pp. 1108-1115, Jan. 2013.

[11] X. Luo, M. C. Zhou, H. Leung, Y. N. Xia, Q.-S. Zhu, Z. H. You, and S. Li, "An Incremental-and-Static-Combined Scheme for Matrix-Factorization-Based Collaborative Filtering," *IEEE Trans. on Automation Science and Engineering,* vol. 13, no. 1, pp. 333-343, Jan. 2016.

[12] X. Luo, M. C. Zhou, S. Li, Z. H. You, Y. N. Xia, Q.-S. Zhu, and H. 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.

[13] J. Wu, L. Chen, Y. P. Feng, Z. B. Zheng, M. C. Zhou, and Z. H. Wu, "Predicting Quality of Service for Selection by Neighborhood-Based Collaborative Filtering," *IEEE Trans. on Systems, Man, and Cybernetics: Systems,* vol. 43, no. 2, pp. 428-439, Mar. 2013.

[14] L. Yang, X. C. 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, Dec. 2015.

[15] T. He, L. Hu, K. C. C. Chan, and P. Hu, "Learning Latent Factors for Community Identification and Summarization," *IEEE Access*, vol. 6, no. 99, pp. 30137-30148, Jun. 2018.

[16] J. Cao, Z. Bu, G. L. Gao, and H. C. Tao, "Weighted Modularity Optimization for Crisp and Fuzzy Community Detection in Large-Scale Networks," *Physica A: Statistical Mechanics and its Applications*, vol. 462, pp. 386-395, May. 2016.

[17] J. Cao, Z. A. Wu, J. J. Wu, and H. Xiong, "SAIL: Summation-based Incremental Learning for Information-Theoretic Text Clustering," *IEEE Trans. on Cybernetics*, vol. 43, no. 2, pp. 570-584, Apr. 2013.

[18] Y. Wang and Y. Zhang, "Nonnegative Matrix Factorization: A Comprehensive Review," *IEEE Trans. on Knowledge and Data*

Engineering, vol. 25, no. 6, pp. 1336-1353, Jun. 2013.

[19] D. Rafailidis, and A. Nanopoulos, "Modeling Users Preference Dynamics and Side Information in Recommender Systems," *IEEE Trans. on Systems, Man, and Cybernetics: Systems,* vol. 46, no. 6, pp. 782-792, Jun. 2016.

[20] A. Naman, A. Z. Zeyuan, B. Brian, H. Elad, and T. Ma, "Finding Approximate Local Minima for Nonconvex Optimization in Linear Time," *arXiv preprint arXiv:1611.01146*, Nov. 2016.

[21] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, "Accelerated Methods for Nonconvex Optimization," *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 1751-1772, Jan. 2018.

[22] B. Srinadh, N. Behnam, and S. Nathan, "Global Optimality of Local Search for Low Rank Matrix Recovery," *arXiv preprint arXiv:1605.07221*, May. 2016.

[23] J. Zeng, K. Ma, and Y. Yao, "On Global Linear Convergence in Stochastic Nonconvex Optimization for Semidefinite Programming," *IEEE Trans. on Signal Processing*, vol. 67, no. 16, pp. 4261-4275, Aug. 2019.

[24] Duc N. Tran, S. Rüffer Björn, and M. Kellett Christopher, "Convergence Properties for Discrete-Time Nonlinear Systems," *IEEE Trans. on Automatic Control*, vol. 64, no. 8, pp. 3415-3422, Aug. 2019.

[25] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to Escape Saddle Points Efficiently," in *Proc. of the 34th Int. Conf. on Machine Learning*, Sydney, Australia, Mar. 2017, vol. 70, pp. 1724-1732.

[26] X. Luo, J. P. Sun, Z. Wang, S. Li, and M. Shang, "Symmetric and Non-negative Latent Factor Models for Undirected, High Dimensional and Sparse Networks in Industrial Applications," *IEEE Trans. on Industrial Informatics*, vol. 13, no. 6, pp. 3098-3107, Jul. 2017.

[27] W. Wang, and Y. Jiang. "Community-aware Task Allocation for Social Networked Multiagent Systems," *IEEE Trans. on Cybernetics, vol.* 44, no. 9, pp. 1529-1543, Sept. 2014.

[28] 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, May. 2014.

[29] 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, Jun. 1994.

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

[31] D. D. Lee, and S. H. Seung, "Algorithms for Non-negative Matrix Factorization," *Advances in Neural Information Processing Systems,* vol. 13, no. 2000, pp. 556-562, Jan. 2000.

[32] C. J. Lin, "Projected Gradient Methods for Nonnegative Matrix Factorization," *Neural Computation,* vol. 19, no. 10, pp. 2756-2779, Oct. 2007.

[33] 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, Jun. 2007.

[34] K. Huang, N. D. Sidiropoulos, A. P. Liavas, "A Flexible and Efficient Algorithmic Framework for Constrained Matrix and Tensor Factorization," *IEEE Trans. on Signal Processing*, vol. 64, no. 19, pp. 5052-5065, Oct. 2016.

[35] M. Yin, J. B. Gao, Z. C. Lin, Q. F. Shi, and Y. Guo, "Dual Graph Regularized Latent Low-Rank Representation for Subspace Clustering," *IEEE Trans. on Image Processing*, vol. 24, no. 12, pp. 4918-4933, Dec. 2015.

[36] 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, Mar. 2017.

[37] P. Wu, A. Che, F. Chu, and M. Zhou, "An Improved Exact ε-Constraint and Cut-and-Solve Combined Method for Biobjective Robust Lane Reservation," *IEEE Trans. on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1479-1492, Jun. 2015.

[38] L. Feng, and B. Bhanu, "Semantic Concept Co-Occurrence Patterns for Image Annotation and Retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 785-799, Jan. 2016.

[39] A. Che, P. Wu, F. Chu, and M. Zhou, "Improved Quantum-Inspired Evolutionary Algorithm for Large-Size Lane Reservation," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 12, pp. 1535-1548, Dec. 2015.

[40] Q. Kang, J. Wang, M. Zhou, and A. C. Ammari, "Centralized Charging Strategy and Scheduling Algorithm for Electric Vehicles Under a Battery Swapping Scenario," *IEEE Trans. on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 659-669, Mar. 2016.

[41] I. Meganem, Y. Deville, S. Hosseini, P. Deliot, and X. Briottet, "Linear-Quadratic Blind Source Separation Using NMF to Unmix Urban Hyperspectral Images," *IEEE Trans. on Signal Processing,* vol. 62, no. 7, pp. 1822-1833, Apr. 2014.

[42] S. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge: Cambridge University Press, 2009.

[43] C. X. Ou, and R. M. Davison, "Technical Opinion: Why eBay Lost to TaoBao in China: the Glocal advantage," *Communications of the ACM,* vol. 52, no. 1, pp. 145-148, Jun. 2009.

[44] 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, Mar. 1997.

[45] Y. Zhuang, W. S. Chin, Y. C. Juan, and C. J. Lin, "A Fast Parallel SGD for Matrix Factorization in Shared Memory Systems," in *Proc. of the 7th ACM Conf. on Recommender Systems, ACM*, Oct. 2013, pp. 249-256.

[46] L. X. Li, L. Wu, H.-S. Zhang, 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, Oct. 2014.

[47] X. Luo, Z. G. Liu, S. Li, M. S. Shang, and Z. D. Wang, "A Fast Non-negative Latent Factor Model based on Generalized Momentum Method," *IEEE Trans. on System, Man, and Cybernetics: Systems*, DOI: 10.1109/TSMC.2018.2875452, Nov. 2018.

[48] R. Zhao and V. Y. F. Tan, "A Unified Convergence Analysis of the Multiplicative Update Algorithm for Regularized Nonnegative Matrix Factorization," *IEEE Trans. on Signal Processing*, vol. 66, no. 1, pp. 129-138, Jan. 2018.

[49] S. Yang, Z. Yi, M. Ye and X. He, "Convergence Analysis of Graph Regularized Non-negative Matrix Factorization," *IEEE Trans. on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2151-2165, Sept. 2014.

[50] D. Cai, X. F. He, J. W. Han, and T. S. Huang, "Graph Regularized Nonnegative Matrix Factorization for Data Representation," *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548-1560, Sept. 2011.

[51] N. Takahashi and R. Hibi, "Global Convergence of Modified Multiplicative Updates for Nonnegative Matrix Factorization," *Computational Optimization and Applications*, vol. 57, no. 2, pp. 417-440, Mar. 2014.

[52] W. Xu, X. Liu, and Y. H. Gong, "Document Clustering Based On Non-negative Matrix Factorization," In *Proc. of the 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, ACM*, New York, NY, USA, Jul. 2003, pp. 267-273.

[53] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the Em Algorithm," *J. Royal Statistical Soc. Series B (Methodological)*, vol. 39, no. 1, pp. 1-38, 1977.

[54] F. Wang, T. Li, X. Wang, S. H. Zhu, and C. Ding, "Community Discovery Using Nonnegative Matrix Factorization," *Data Mining and Knowledge Discovery*, vol. 22, no. 3, pp. 493-521, May. 2011.

[55] Z. He, S. Xie, Z. Rafal, G. Zhou and C. Andrzej, "Symmetric Nonnegative Matrix Factorization: Algorithms and Applications to Probabilistic Clustering," *IEEE Trans. on Neural Networks*, vol. 22, no.12, pp. 2117-2131, Dec. 2011.

[56] H. Ma, I. King, and M. R. Lyu, "Learning to Recommend with Social Trust Ensemble," in *Proc. Proc. of the 32nd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Boston, MA, USA, Jul. 2009, pp. 203-210.

[57] X. Luo, D. Wang, M. C. Zhou, and H. Yuan, "Latent Factor-based Recommenders Relying on Extended Stochastic Gradient Descent Algorithms," *IEEE Trans. on System Man Cybernetics: Systems*, DOI 10.1109/TSMC.2018.2884191, Jan. 2019.

[58] X. Luo, Z. Wang, and M. Shang, "An Instance-Frequency-Weighted Regularization Scheme for Non-negative Latent Factor Analysis on High-Dimensional and Sparse Data," *IEEE Trans. on System Man Cybernetics: Systems*, DOI:10.1109/TSMC.2019.2930525, Aug. 2019.

[59] A. Cichocki and Z. Rafal, "Multilayer Nonnegative Matrix Factorization Using Projected Gradient Approaches," *International Journal of Neural Systems*, vol. 17, no, 06, pp. 431-446, Dec. 2007.

[60] X. Ding, J. Lee, and S. Lee, "A Constrained Alternating Least Squares Nonnegative Matrix Factorization Algorithm Enhances Task-related Neuronal Activity Detection from Single Subject's fMRI Data," In *Proc.*

of the 2011 IEEE Int. Conf. on Machine Learning and Cybernetics, Jul. 2011, pp. 338-343.

[61] J. Mohsen and M. Ester, "A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks," in *Proc. of the 4th ACM Conf. on Recommender Systems*, Barcelona, Spain, Sept. 2010, pp. 135-142.

[62] P. Massa, and P. Avesani, "Trust-aware Recommender Systems," in *Proc. of the 1st ACM Conf. on Recommender Systems*, Minneapolis, MN, USA, Oct. 2007, pp. 17-24.

[63] N. Hurley and M. Zhang, "Novelty and Diversity in top-N Recommendation: Analysis and Evaluation," *ACM Trans. on Internet Technology*, vol. 10, no. 4, pp. 1-30, Mar. 2011.

[64] D. Szklarczyk et al., "STRING v10: Protein-protein Interaction Networks, Integrated Over the Tree of Life," *Nucleic Acids Research*, vol. 43, no. D1, pp. D447-D452, Jan. 2015.

[65] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1-30, Dec. 2006.

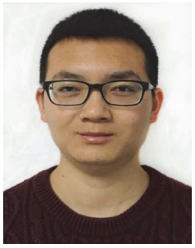
[66] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising autoencoder," In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ACM, Oct. 2015, pp. 811-820.

[67] M. Shang, X. Luo, Z. Liu, J. Chen, and Y. Yuan, "Randomized Latent Factor Model for High-dimensional and Sparse Matrices from Industrial Applications," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, pp. 131-141, Jan. 2019.

[68] W. Yang, Y. Shi, Y. Gao, and M. Yang, "Incomplete-data oriented multiview dimension reduction via sparse low-rank representation," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 6276-6291, May. 2018.

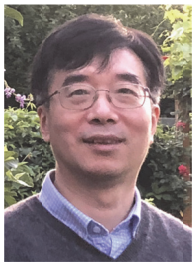
[69] A. Balavoine, J. Romberg, and C. J. Rozell, "Convergence and Rate Analysis of Neural Networks for Sparse Approximation," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 23, no. 9, pp. 1377-1389, Sept. 2012.

[70] M. Gong, J. Liu, H. Li, Q. Cai, and L. Sun, "A Multiobjective Sparse Feature Learning Model for Deep Neural Networks," *Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3263-3277, Dec. 2015.



Zhigang Liu received the B.S. degree in geographical information system from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2013, and his M.E. degree in computer science and technology from Chongqing University, Chongqing, China. He is currently pursueing his Ph.D. degree jointy at Chongqing University of Posts and Telecommunications, and Chongqing College of University of Chinese Academy of Sciences, Chongqing, China. His research interests include big data analysis and algorithm design.

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 40 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.



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

He is currently a Professor of Dynamical Systems and Computing in the Department of Computer Science, Brunel University London, U.K. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany and the UK. Prof. Wang's research interests include dynamical systems, signal processing, bioinformatics, control theory and applications. He has published more than 500 papers in refereed international journals. He is a holder of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, William Mong Visiting Research Fellowship of Hong Kong.

Prof. Wang serves (or has served) as the Editor-in-Chief for Neurocomputing, the Deputy Editor-in-Chief for International Journal of Systems Science, and an Associate Editor for 12 international journals, including IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks, IEEE Transactions on Signal Processing, and IEEE Transactions on Systems, Man, and Cybernetics-Part C. He is a Fellow of the IEEE, a Fellow of the Royal Statistical Society and a member of program committee for many international conferences.