Received March 3, 2016, accepted March 31, 2016, date of publication April 20, 2016, date of current version June 13, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2556680

A Novel Approach to Extracting Non-Negative Latent Factors From Non-Negative Big Sparse Matrices

XIN LUO[1,2], (Member, IEEE), MENGCHU ZHOU[3,4], (Fellow, IEEE), MINGSHENG SHANG[1,2], SHUAI LI[5], (Member, IEEE), AND YUNNI XIA[6], (Senior Member, IEEE)

[1]Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China
[2]Shenzhen Engineering Laboratory for Mobile Internet Application Middleware Technology of Shenzhen University, Shenzhen 518060, China
[3]Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China
[4]Helen and John C. Hartmann Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA
[5]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[6]Chongqing Key Laboratory of Software Theory and Technology, College of Computer Science, Chongqing University, Chongqing 400044, China

Corresponding authors: X. Luo (luoxin21@cigit.ac.cn) and M. Zhou (zhou@njit.edu)

This work was supported in part by the Fundo para o Desenvolvimento das Ciências e da Tecnologia under Grant 119/2014/A3, in part by the National Natural Science Foundation of China under Grant 61370150 and Grant 61433014, in part by the Young Scientist Foundation of Chongqing under Grant cstc2014kjrc-qnrc40005, in part by the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2015jcyjB0244, in part by the Post-Doctoral Science Funded Project of Chongqing under Grant Xm2014043, in part by the Fundamental Research Funds for the Central Universities under Grant 106112015CDJXY180005, and in part by the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20120191120030.

ABSTRACT An inherently non-negative latent factor model is proposed to extract non-negative latent factors from non-negative big sparse matrices efficiently and effectively. A single-element-dependent sigmoid function connects output latent factors with decision variables, such that non-negativity constraints on the output latent factors are always fulfilled and thus successfully separated from the training process with respect to the decision variables. Consequently, the proposed model can be easily and fast built with excellent prediction accuracy. Experimental results on an industrial size sparse matrix are given to verify its outstanding performance and suitability for industrial applications.

INDEX TERMS Latent factors, non-negativity, matrix factorization, non-negative big sparse matrix, big data, recommender system.

I. INTRODUCTION

Big sparse matrices are commonly seen in industrial applications, such as rating matrices in recommender systems [1], quality-of-service matrices in services computing [2], and protein interactome matrices in bioinformatics [3]. Latent factor (LF) models [4]–[6] are able to extract hidden LFs from them and can be used to address various important issues such as missing data estimation [1], [4]–[8], and community detection [9], [10]. However, the existing models mostly do not fulfill non-negativity constraints, i.e., obtained LFs might be negative. This is inconsistent with most application data, e.g., user-commodity preferences [1], [4]–[8], quality-of-service records [2], and social impacts [9], [10], are non-negative. Consequently, LFs extracted from these data are required to fulfill the non-negativity constraints in order to represent their actual meanings correctly and precisely [6]–[13].

Given a complete target matrix, several approaches are developed to extract non-negative LFs from it, e.g., the non-negative matrix factorization (NMF) based on non-negative alternating least squares [11], non-negative multiplicative updates [12] and projected gradient descent [13]. In spite of their popularity in many related issues [14]–[17], they are not applicable to LF models whose matrices are sparse [1]–[10]. If NMF models are used addressing their sparsity issue by introducing weight indicator matrices [9], [10] or interim approximation matrices [18], the problem of high complexity arises. Their computational and storage complexity is linear with the full size of a sparse matrix instead of the number of its known entries [1]–[8]. The latter is commonly a small fraction of the former, e.g., 1.34% in our later example [23].

To address non-negative big sparse matrices, a non-negative LF model is demanded to focus on their known

entry set when building desired LFs, and maintain the non-negativity constraints at the same time. One feasible approach is to combine the constraints into the training process through rescaling the hyper parameters according to the Karush-Kuhn-Tucker condition of a constrained optimization system [6], [8], [12]. Another is to integrate the principle of alternating-direction-method to implement a non-negative factorization process. But such specifically designed non-negative models are neither flexible nor extensible, e.g., commonly adopted gradient-based learning schemes and their extensions [1]–[5], [10] are not applicable to them [6]–[10], [12]. On the other hand, as discussed in [19], [20], non-negative LFs can be obtained by appending non-negative projection onto a generalized training scheme [13], [19], [20]. It is compatible with general training schemes and effective as proven in [19] and [20], but can result in many zero-valued LFs.

This paper proposes to solve this problem in a different way, i.e., transforming the constrained problem into an unconstrained one. To do so, we separate the non-negative constraints from the model training process by connecting output LFs and decision variables through a single-element-dependent sigmoid function, thereby achieving inherent non-negativity and a free optimization process. The resulting Inherently Non-negative Latent Factor model is highly efficient and effective. To be shown via the experimental results, it is able to achieve the best performance when addressing non-negative big sparse matrices.

II. PROBLEM OF A NON-NEGATIVE LATENT FACTOR MODEL

An LF model addresses sparse matrices that are defined as follows.

Definition 1: Given two entity sets U and S, a target matrix T is a $|U| \times |S|$ matrix where its element $t_{u,s}$ describes some relationship between entities $u \in U$ and $s \in S$; let Λ denote T's known entry set, then $|\Lambda| \ll |U| \times |S|$.

For ease of expression, throughout this paper by "(u, s) in T" we mean "T's entry's index pair (u, s)," i.e., $1 \leq u \leq |U|$ and $1 \leq s \leq |S|$, where T is a $|U| \times |S|$ matrix. Given T and Λ, an LF model is defined as follows,

Definition 2: Given matrix T, an LF model is its approximation $\hat{T} = PQ^T$, where P is $|U| \times d$, Q is $|S| \times d$ matrices, and $d \ll \min\{|U|, |S|\}$.

P and Q are actually LF matrices hidden in Λ. We obtain them by minimizing an objective function of the difference between Λ and corresponding entries in $\hat{T}$. With Euclidean distance [1]–[10], we formulate it as:

$$\arg\min_{P,Q} \varepsilon\,(P, Q) = \frac{1}{2} \sum_{(u,s)\in\Lambda} \left(t_{u,s} - \sum_{k=1}^{d} p_{u,k} q_{s,k} \right)^2 \quad (1)$$

where $\varepsilon(P, Q)$ denotes the generalized error regarding desired P and Q, $t_{u,s}$, $p_{u,k}$ and $q_{s,k}$ denote the corresponding entries in matrices T, P and Q, respectively. Note that for ease of expression, throughout this paper we slightly abuse the symbols by letting (u, s) denote the couple index where $1 \leq u \leq |U|$ and $1 \leq s \leq |S|$.

Many real applications require their data to be non-negative, i.e., we have $\forall(u, s)$ in T, $t_{u,s} \geq 0$. Hence, we have to constrain P and Q to be non-negative [1]–[8]. Thus, we have:

$$\begin{aligned} &\arg\min_{P,Q} \varepsilon\,(P, Q) = \frac{1}{2} \sum_{(u,s)\in\Lambda} \left(t_{u,s} - \sum_{k=1}^{d} p_{u,k} q_{s,k} \right)^2, \\ &s.t.\ \forall(u, s)\ in\ T, t_{u,s} \geq 0, \\ &\quad \forall(u, k)\ in\ P, p_{u,k} \geq 0, \\ &\quad \forall(s, k)\ in\ Q, q_{s,k} \geq 0; \end{aligned} \quad (2)$$

which is the problem of a non-negative LF model.

III. INHERENTLY NON-NEGATIVE LATENT FACTOR MODEL

The main barrier preventing us from solving (2) directly with general learning schemes is the possible violation of its non-negativity constraints. All LFs, i.e., $p_{u,k}$ and $q_{s,k}$ in (2), cannot be guaranteed to be non-negative after a training process. To present our method, we firstly introduce two matrices $X^{|U|\times d}$, and $Y^{|I|\times d}$. With them, we further introduce two single-element-dependent functions f and g, which map each element in X and Y to the corresponding entries in P and Q, respectively. Formally, we have

$$p_{u,k} = f\left(x_{u,k}\right), \quad q_{s,k} = g\left(y_{s,k}\right). \quad (3)$$

Note that if we carefully choose f and g to maintain the non-negativity, i.e., making the following inequalities hold,

$$\forall x_{u,k} \in \mathbb{R},\ p_{u,k} = f\left(x_{u,k}\right) \geq 0,\ \forall y_{s,k} \in \mathbb{R},\ q_{s,k} = g\left(y_{s,k}\right) \geq 0, \quad (4)$$

then optimizing the loss function with respect to the decision variables, i.e., X and Y, will no longer be constrained. Note that the mapping functions should not introduce additional local optima into the original system in order to keep its solution field, i.e.,

$$\forall x_{u,k} \in \mathbb{R},\ f'\left(x_{u,k}\right) \neq 0,\ \forall y_{s,k} \in \mathbb{R},\ g'\left(y_{s,k}\right) \neq 0. \quad (5)$$

To meet (4) and (5) simultaneously, we make f and g be single-element-dependent sigmoid functions, which are frequently adopted for slacking constrained optimization problems in the optimization community [24]. More specifically, let $\phi(\alpha)$ denote a sigmoid function, i.e., $\phi(\alpha) = (1 + e^{-\alpha})^{-1}$ where e denotes a mathematical constant. Then we map $x_{u,k}/y_{s,k}$ to $p_{u,k}/q_{s,k}$ as follows,

$$p_{u,k} = \phi\left(x_{u,k}\right), \quad q_{s,k} = \phi\left(y_{s,k}\right); \quad (6)$$

Then we transform the original problem (2) into:

$$\arg\min_{X,Y} \varepsilon\left(X,Y\right)=\frac{1}{2}\sum_{(u,s)\in\Lambda}\left(t_{u,s}-\sum_{k=1}^{d}\phi\left(x_{u,k}\right)\cdot\phi\left(y_{s,k}\right)\right)^2, \tag{7}$$

which possesses the inherent non-negativity, i.e.,

$$\begin{aligned}&\forall(u,k)\ in\ P,\quad \forall x_{u,k}\in\mathbb{R}, p_{u,k}=\phi\left(x_{u,k}\right)>0,\\&\forall(s,k)\ in\ Q,\quad \forall y_{s,k}\in\mathbb{R}, q_{s,k}=\phi\left(y_{s,k}\right)>0,\\&\forall(u,s)\ in\ T,\quad \hat{t}_{u,s}=\sum_{k=1}^{f}p_{u,k}q_{s,k}>0.\end{aligned} \tag{8}$$

Note that (7) constrains entries in P and Q to be in the range of (0, 1). However, it can be proven that (7) is able to represent (2), since a) with a linear transformation, we can rescale Λ and any solution to (2) to be in the range of [0, 1) without loss of generality; in such cases, by noting that α in $\phi(\alpha)$ is defined on P, we can approximate any solution to (2) at any desired precision with another solution to (7); b) both (2) and (7) seek for the same objective; and c) the output LFs of (7), i.e., P and Q, fulfill the non-negativity constraints inherently. Note that (7) and kernel NMF [14], [25], which is a variant of NMF, possess similar principles of slacking the non-negative constraints through constantly positive non-linear mapping functions (kernels). However, (7) considers the problem on positive big, sparse matrices, and the corresponding modeling is taken in a single-element-dependent way to fit a sparse target matrix.

Moreover, as proven in prior research [1]–[8], it is necessary to integrate the l_2 norm of the output LFs, i.e., P and Q, into (7) for regularization. Hence, we introduce a Tikhonov regularization term into (7), thus extending it into:

$$\begin{aligned}&\arg\min_{X,Y}\varepsilon\left(X,Y\right)\\&=\frac{1}{2}\sum_{(u,s)\in\Lambda}\left(\left(t_{u,s}-\sum_{k=1}^{d}\phi\left(x_{u,k}\right)\cdot\phi\left(y_{s,k}\right)\right)^2\right.\\&\qquad\left.+\lambda\sum_{k=1}^{d}\left(\phi^2\left(x_{u,k}\right)+\phi^2\left(y_{s,k}\right)\right)\right),\end{aligned} \tag{9}$$

which tries to balance between minimizing the generalized error and minimizing the l_2 norm of the output LFs related to Λ, i.e., $\forall p_{u,k}=\phi(x_{u,k})$ in P and $\forall q_{s,k}=\phi(y_{s,k})$ in Q. Note that such regularization design takes the instance frequency related to each desired LF into consideration, i.e., heavy regularization effect is assigned to LFs corresponding to many known instances, and vice versa. The positive effect of such design can be found in [21]. In practice, the parameter λ should be tuned on training data for the best performance. Its empirical scale is (0.001, 0.1) according to our experience.

Note that (9) guarantees the non-negativity of P and Q via f and g as in (8). So no constraints are taken into consideration when minimizing ε with respect to the decision variables, i.e., X and Y. We choose stochastic gradient descent (SGD) that is very popular in building an LF model [1]–[5] as the solver. With it we minimize (9) with respect to X and Y through the stochastic approximation to the gradient on each training instance, i.e., with the following training rule,

$$\begin{aligned}&\forall(u,s)\in\Lambda,\quad k=1\sim d\\&\Rightarrow\begin{cases}x_{u,k}\leftarrow x_{u,k}-\eta\dfrac{\partial\varepsilon_{(u,s)}}{\partial p_{u,k}}\cdot\dfrac{dp_{u,k}}{dx_{u,k}},\\ y_{s,k}\leftarrow y_{s,k}-\eta\dfrac{\partial\varepsilon_{(u,s)}}{\partial q_{s,k}}\cdot\dfrac{dq_{s,k}}{dy_{s,k}};\end{cases}\end{aligned} \tag{10}$$

where $\varepsilon_{(u,s)}$ denotes the error on instance $t_{u,s}\in\Lambda$, and η denotes the learning rate. By applying (10) to (9) we obtain:

$$\begin{cases}\dfrac{\partial\varepsilon_{(u,s)}}{\partial p_{u,k}}=-\phi\left(y_{e,k}\right)\left(t_{u,s}-\sum_{k=1}^{d}\phi\left(x_{u,k}\right)\cdot\phi\left(y_{s,k}\right)\right)\\ \qquad\qquad+\lambda\phi\left(x_{u,k}\right),\\ \dfrac{\partial\varepsilon_{(u,s)}}{\partial q_{s,k}}=-\phi\left(x_{u,k}\right)\left(t_{u,s}-\sum_{k=1}^{d}\phi\left(x_{u,k}\right)\cdot\phi\left(y_{s,k}\right)\right)\\ \qquad\qquad+\lambda\phi\left(y_{s,k}\right).\end{cases} \tag{11}$$

According to the characteristics of $\phi(x)$ we have:

$$\begin{aligned}\phi'\left(x_{u,k}\right)&=\phi\left(x_{u,k}\right)\left(1-\phi\left(x_{u,k}\right)\right),\\ \phi'\left(y_{s,k}\right)&=\phi\left(y_{s,k}\right)\left(1-\phi\left(y_{s,k}\right)\right).\end{aligned} \tag{12}$$

By combining (10)-(12) we obtain the following update rule,

$$\begin{aligned}&\forall(u,s)\in\Lambda,\quad for\ k=1\sim d,\\&\Rightarrow\begin{cases}x_{u,k}\leftarrow x_{u,k}+\eta\phi\left(x_{u,k}\right)\left(1-\phi\left(x_{u,k}\right)\right)\\ \qquad\times\left(\phi\left(y_{s,k}\right)err_{u,s}-\lambda\phi\left(x_{u,k}\right)\right),\\ y_{s,k}\leftarrow y_{s,k}+\eta\phi\left(y_{s,k}\right)\left(1-\phi\left(y_{s,k}\right)\right)\\ \qquad\times\left(\phi\left(x_{u,k}\right)err_{u,s}-\lambda\phi\left(y_{s,k}\right)\right);\end{cases}\end{aligned} \tag{13}$$

where $err_{u,s}=t_{u,s}-\sum_{k=1}^{d}\phi\left(x_{u,k}\right)\cdot\phi\left(y_{s,k}\right)$. Note that the frequently used $\phi(\alpha)$ in (13) requires us to raise the mathematical constant e to the power of an arbitrary real number α. It is time consuming in some programming language, and we can use $n=2^{10}=1024$ in the following expression

$$e^{\alpha}=\lim_{n\to+\infty}\left(1+\frac{\alpha}{n}\right)^n \tag{14}$$

to obtain the highly accurate approximation of e^{α} with constant time, i.e., $\Theta(1)$. The constant n in (14) can be any positive real number. We choose to set $n=2^{10}$ because a) the resulted approximation can be very close to e^{α}; and b) such a formula can be resolved in a bottom-up way easily. When the training process ends, the desired non-negative LFs P and Q are obtained by:

$$\begin{aligned}&\forall(u,k)\ in\ P,\quad p_{u,k}=\phi\left(x_{u,k}\right),\\&\forall(s,k)\ in\ Q,\quad q_{s,k}=\phi\left(y_{s,k}\right).\end{aligned} \tag{15}$$

Based on the inference above, we obtain the Inherently Non-Negative Latent Factor (INLF) model for non-negative sparse matrices. Next we conduct experiments on such a matrix of industrial size to shown its effectiveness and efficiency.

IV. EXPERIMENTS AND RESULTS

A. EVALUATION METRICS

To evaluate the accuracy of an LF-based model, we choose the root mean squared error (RMSE) and mean absolute error (MAE) [22], both of which enjoy their popularity in evaluating the statistical accuracy of LF models:

$$RMSE = \sqrt{\left(\sum_{(u,s)\in\Gamma}\left(t_{u,s}-\hat{t}_{u,s}\right)^2\right)\Big/|\Gamma|},$$
$$MAE = \left(\sum_{(u,s)\in\Gamma}\left|t_{u,s}-\hat{t}_{u,s}\right|_{abs}\right)\Big/|\Gamma|; \quad (16)$$

where Γ denotes the validation set and naturally Λ and Γ are disjoint, and $|\cdot|_{abs}$ denotes the absolute value of a given number.

To measure its computational efficiency, we have to find a) the converging training round count, and b) the average time per round. All experiments are conducted on a PC with a 2.5 GHz i5 CPU and 32 GB RAM. All models are implemented in JAVA SE 7U60 to test their suitability for industrial usage.

B. DATASET

The experiments are conducted on the MovieLens 10M dataset [23]. It is a well-known and widely used non-negative big sparse matrix in the community of recommender systems [1]–[8], [23]. It contains 10,000,054 known entries in the range of [0.5, 5], by 69,878 users on 10,677 movies. Its data density is 1.34% only.

The experimental dataset is randomly split into five equally-sized, disjoint subsets. 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 to predict the remaining subset as Γ for performance test. This process is sequentially repeated for five times to obtain the final results. In our experiments, the training process of a tested model terminates if a) the iteration number reaches a preset threshold, i.e., 1000, and b) the model converges, i.e., its training error rises after achieving the minimum, or the difference in the training error of two consecutive rounds is smaller than 10^{-5}. When a model seems to converge, 50 more rounds are taken to avoid the misjudgment caused by error fluctuations.

C. COMPARED MODELS

Besides the proposed INLF, four rival models are selected: a) the Weighted Non-negative Matrix Factorization (WNMF) model [9], a widely adopted NMF model for incomplete matrices [9], [10]; b) the Non-negative Matrix Completion (NMC) model [18], which non-negatively factorizes an incomplete non-negative matrix through manipulating its full approximation; c) the proximal alternating linearized minimization-based NMF (PALM-NMF) model [19], [20], a highly competitive model for non-convex and non-smooth optimization problems; and d) the projected stochastic gradient descent-based NMF model (PSGD-NMF), which also adopts SGD for extracting non-negative LFs. Note that both PALM-NMF and PSGD-NMF ensure the non-negativity of output LFs through projecting them onto the non-negative field of real numbers during the training process. Moreover, PALM and PSGD are compatible with specified optimization problems, and thus we make PALM-NMF and PSGD-NMF depend on the regularized form of (2) to guarantee a fair comparison.

D. MODEL SETTINGS

Note that the performance of an LF model is affected by its initial states and the LF dimension. To obtain objective results, the LF dimension is set as $d = 20$ uniformly. To minimize the impact caused by an initial guess, the LFs of each model are initialized with the same randomly-generated arrays, whose entries are scattered in the interval of (0, 0.01).

For hyper parameters, a) WNMF has no hyper parameters [7]; b) NMC depends on the learning rate γ, and augmentation terms α and β. Following [18], we set $\gamma = 1.618$, and $\beta = \alpha|U|/|S|$. We tune α on one fold and adopt the same value on the other four folds according to [18]; c) for INLF, PALM-NMF and PSGD-NMF, the regularizing coefficient λ is set at 0.02 uniformly; d) PALM-NMF's learning rate is self-adaptive with back-tracking line search; e) PSGD-NMF's learning rate is set at 0.01; and f) INLF's learning rate η is set at 0.1.

E. RESULTS

Tables 1 records their RMSE and MAE along with the consumed iterations. Tables 2 records the average time cost by each iteration, along with the total time to make them converge. Fig. 1 depicts their typical training process. From these results, we have the following findings:

TABLE 1. RMSE, MAE and Converging Round Count of Compared Models.

Model	RMSE/Rounds	MAE/Rounds
WNMF	0.8597/1000	0.6496/1000
NMC	0.8693/13.6	0.6745/17.4
PALM-NMF	0.8094/1000	0.6233/1000
PSGD-NMF	0.8128/86.8	0.6281/97.2
INLF	**0.7935/62.8**	**0.6109/67.0**

TABLE 2. Consumed Time of Compared Models.

Model	Average Time (ms)	Total Time (seconds)
WNMF	14,850	14,850/14,850
NMC	3,112,127	42,324/54,151
PALM-NMF	9,695	9,695/9,695
PSGD-NMF	1,928	167/187
INLF	**4,036**	**253/270**

a) Among all compared models, INLF is able to achieve the highest prediction accuracy, as recorded in Table 1 and Fig. 1. On D1, its RMSE is 0.7935, compared with 0.8597

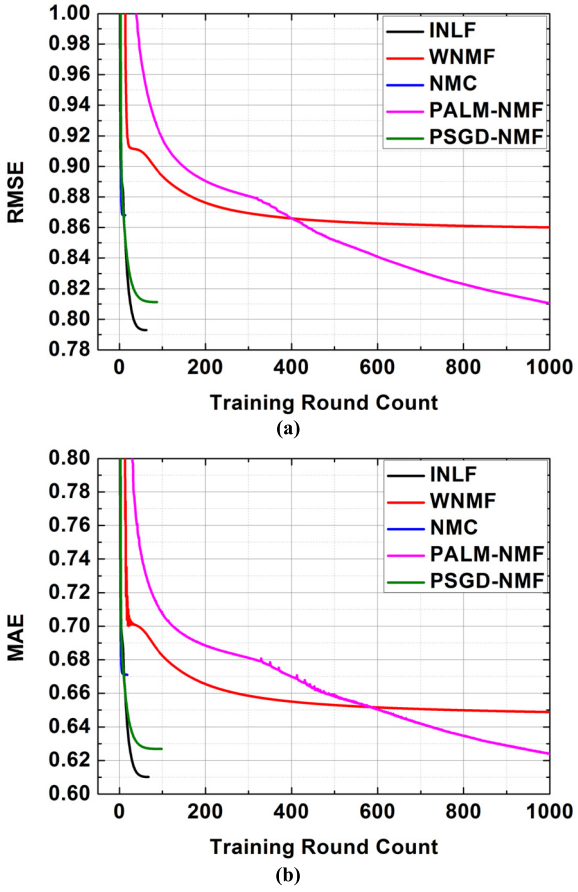


FIGURE 1. Typical training process of all models in RMSE and MAE. (a) RMSE. (b) MAE.

by WNMF, 0.8693 by NMC, 0.8094 by PALM-NMF and 0.8128 by PSGD-NMF, thus leading to 7.70%, 8.72%, 1.96% and 2.37% improvement, respectively. In terms of MAE, INLF achieves 0.6109, which is 5.96%, 9.43%, 1.99% and 2.74% lower than 0.6496, 0.6745, 0.6233 and 0.6281 by WNMF, NMC, PALM-NMF and PSGD-NMF respectively.

Note that PALM-NMF and PSGD-NMF are also designed for the non-convex problem (2) on non-negative big sparse matrices. However, their prediction accuracy for corresponding missing data is lower than INLF. The reason for this phenomenon is partly reflected by the distributions of non-negative LFs extracted by these three models, as depicted in Fig. 2.

As shown in Fig. 2, we see that PALM-NMF and PSGD-NMF projects many LFs to be zero (more than 70,000 for PALM-NMF, and more than 13,000 for PSGD-NMF; given $f = 20$ the total number of LFs on ML20M dataset is 1,611,100, so about 4.3% and 0.8% LFs by PALM-NMF and PSGD-NMF are zero, respectively). This indicates that with a general training scheme like that in PALM-NMF or PSGD-NMF, LFs will probably turn negative. Although we can keep their non-negativity through non-negative projection without harming the model convergence, the LF distribution is bent to some extent, thereby resulting in accuracy loss. We also want to study the effect of such phenomenon in

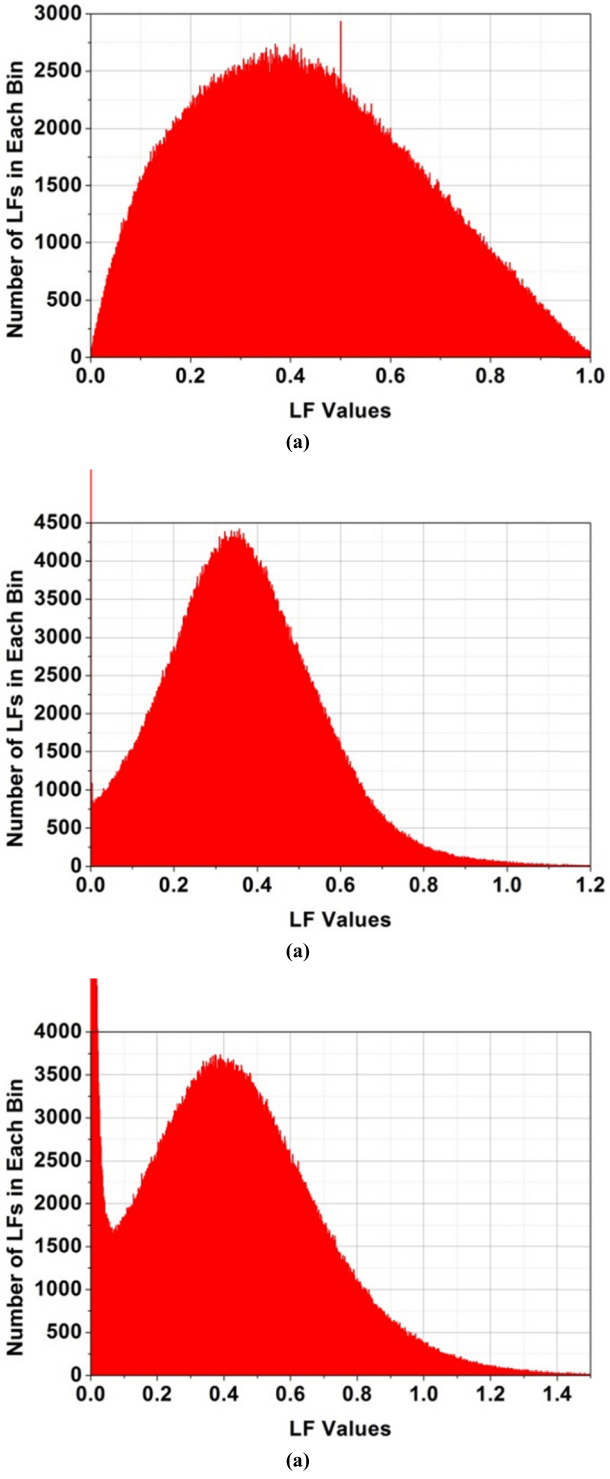


FIGURE 2. Distributions of LFs extracted by INLF, PALM-NMF and PSGD-NMF.

other machine learning tasks like clustering. This issue is left as our future work.

b) In the test, two SGD-based models, i.e., PSGD-NMF and INLF, are able to achieve higher computational efficiency than their peers. The consumed time per round in PSGD-NMF is 1,928 milliseconds, followed

by 4,036 milliseconds in INLF, as recorded in Table 1. Note that INLF relies on an expensive operation, i.e., e^{α}. Although the computational burden can be alleviated through some approximation skills as discussed in Section III, INLF still costs more time than PSGD-NMF does since the latter directly applies SGD on desired LFs.

PALM-NMF conducts back-tracking line search to tune the step size in each iteration. It computes the generalized loss frequently. Thus its consumed time per round is higher than that of PSGD-NMF and INLF, but still much lower than 14,850 milliseconds of WNMF, and 3,112,127 milliseconds of NMC. This phenomenon is reasonable. As described by (14), INLF only requires traversing on Λ to update involved decision variables per round. The cost of this process is linear with $|\Lambda|$. PSGD-NMF and PALM-NMF rely on the regularized form of (2), and their training costs are also linear with respect to $|\Lambda|$. However, the cost per round in WNMF and NMC is linear with the target matrix size, i.e., $|U| \times |S|$.

c) As depicted in Fig. 1, PSGD-NMF and INLF converges very fast. On the ML10M dataset, they require only dozens of training rounds to converge, with high accuracy and fulfillment of the non-negativity constraints. Hence, their total time consumption is low. On ML10M dataset, PSGD-NMF consumes 167 and 187 seconds to converge respectively in RMSE and MAE, followed by INLF with 253 and 270 seconds.

PALM-NMF and WNMF's execution rounds reach the predefined threshold, and their total time cost is higher than that of PSGD-NMF and INLF. Although NMC requires even fewer rounds to converge than PSGD-NMF and INLF do, its time cost is the highest among all models due to its high cost per round.

Note that as recorded in Table 2, NMC consumes even much more time than WNMF does, given that they possess the same theoretical computational cost. WNMF introduces the binary weight matrix to implement the matrix multiplications on sparse matrices [9], [10]. Such operation can be accelerated by ignoring the operations corresponding to zero-entries. On the other hand, NMC employs a full matrix corresponding to the target matrix to implement the MF process. Hence, its each operation is constantly performed on this $69{,}878 \times 10{,}677$ matrix, thereby requiring much time.

d) Based on the above observations, we conclude that INLF is highly efficient and effective on big sparse matrices. It achieves very high prediction accuracy with low computational burden which is comparable with the most efficient non-negative models. Meanwhile, it possesses the inherent characteristic of non-negativity, i.e., the non-negativity constraints are always met in its training process.

V. CONCLUSIONS

This work proposes to extract latent factors from non-negative big sparse matrices via an inherently non-negative latent factor model. Its main idea is to map the decision variables to output latent factors through a single-element-dependent sigmoid function, thereby making the output latent factors constantly non-negative no matter how the decision variables are trained to minimize the objective function. Therefore, the training process with respect to the decision variables can be taken freely depending on general learning schemes. The efficiency and effectiveness of the proposed model are proven by experiments conducted on an industrial size non-negative sparse matrix. Next we plan to investigate its further extension and application in the area of complex network analysis [26]–[31].

REFERENCES

[1] 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. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[2] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. C. Zhou, and Z. Wu, "Predicting quality of service for selection by neighborhood-based collaborative filtering," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 2, pp. 428–439, Mar. 2013.

[3] X. Luo *et al.*, "A highly efficient approach to protein interactome mapping based on collaborative filtering framework," *Sci. Rep.*, vol. 5, p. 7702, Jan. 2015.

[4] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems," *J. Mach. Learn. Res.*, vol. 10, pp. 623–656, Dec. 2009.

[5] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[6] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1273–1284, May 2014.

[7] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q.-S. Zhu, "A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 579–592, Mar. 2016.

[8] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. C. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, Mar. 2016.

[9] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proc. 6th SIAM Int. Conf. Data Mining*, Bethesda, MD, USA, Apr. 2006, pp. 549–553.

[10] Q. Gu, J. Zhou, and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *Proc. 10th SIAM Int. Conf. Data Mining*, Columbus, OH, USA, Apr. 2010, pp. 199–210.

[11] 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.

[12] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[13] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, 2007.

[14] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[15] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1087–1099, Jul. 2012.

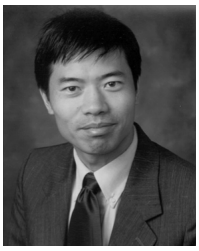
[16] A. Gogna and A. Majumdar, "A comprehensive recommender system model: Improving accuracy for both warm and cold start users," *IEEE Access*, vol. 3, pp. 2803–2813, Dec. 2015.

[17] H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele, and F. Xia, "Context-based collaborative filtering for citation recommendation," *IEEE Access*, vol. 3, pp. 1695–1703, Oct. 2015.

[18] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Frontiers Math. China*, vol. 7, no. 2, pp. 365–384, 2012.

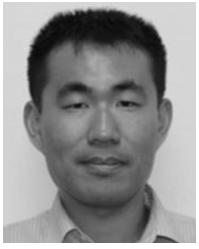
[19] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Math. Program.*, vol. 137, no. 1, pp. 91–129, Feb. 2013.
[20] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, nos. 1–2, pp. 459–494, 2014.
[21] N. Srebro and R. R. Salakhutdinov, "Collaborative filtering in a non-uniform world: Learning with the weighted trace norm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23. 2010, pp. 2056–2064.
[22] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.
[23] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: Applying collaborative filtering to usenet news," *Commun. ACM*, vol. 40, no. 3, pp. 77–87, 1997.
[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[25] D.-Q. Zhang, Z.-H. Zhou, and S. Chen, "Non-negative matrix factorization on kernels," in *Proc. 9th Pacific Rim Int. Conf. Artif. Intell. Trends Artif. Intell. (PRICAI)*, 2006, pp. 404–412.
[26] M. Dotoli, M. P. Fanti, C. Meloni, and M. Zhou, "A multi-level approach for network design of integrated supply chains," *Int. J. Prod. Res.*, vol. 43, no. 20, pp. 4267–4287, Oct. 2005.
[27] C. Jiang, H. Sun, Z. Ding, P. Wang, and M. Zhou, "An indexing network: Model and applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 12, pp. 1633–1648, Dec. 2014.
[28] L. Dong, B. Shi, G. Tian, Y. B. Li, B. Wang, and M. C. Zhou, "An accurate de novo algorithm for glycan topology determination from mass spectra," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 12, no. 3, pp. 568–578, May/Jun. 2015.
[29] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in Internet of vehicles: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2339–2352, Oct. 2015.
[30] G. Xiong *et al.*, "Cyber-physical-social system in intelligent transportation," *IEEE/CAA J. Autom. Sinica*, vol. 2, no. 3, pp. 320–333, Jul. 2015.
[31] H. Sun, C. Jiang, Z. Ding, P. Wang, and M. Zhou, "Topic-oriented exploratory search based on an indexing network," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 2, pp. 234–247, Feb. 2016.

XIN LUO (M'14) 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, in 2011. He is currently a Professor with the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China. His research interests are in big-data analysis and applications-related areas, especially in recommender systems, social-network analysis, services computing, and bioinformatics.

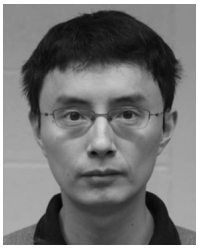


MENGCHU ZHOU (S'88–M'90–SM'93–F'03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, in 1990. He joined the New Jersey Institute of Technology, Newark, NJ, in 1990, and is currently a Distinguished Professor of Electrical and Computer Engineering. His research interests are in Petri nets, Internet of Things, big data, Web services, manufacturing, transportation, and energy systems. He has over 640 publications including 12 books, 330+ journal papers (240+ in the IEEE TRANSACTIONS), and 28 book-chapters. He is the Founding Editor of the *IEEE Press Book Series on Systems Science and Engineering*. He is a recipient of the Humboldt Research Award for U.S. Senior Scientists, the Franklin V. Taylor Memorial Award, and the Norbert Wiener Award from the IEEE Systems, Man and Cybernetics Society. He is a Life Member of the Chinese Association for Science and Technology-USA and served as its President in 1999. He is a fellow of the International Federation of Automatic Control and the American Association for the Advancement of Science.

MINGSHENG SHANG received the B.E. degree in management from Sichuan Normal University, Chengdu, China, in 1995, and the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, in 2007. He is currently a Professor with 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.



SHUAI LI (M'14) received the B.E. degree in mechanical engineering from the Hefei University of Technology, Hefei, China, the M.E. degree in control engineering from the University of Science and Technology of China, Hefei, and the Ph.D. degree in electrical engineering from the Stevens Institute of Technology, Hoboken, NJ, USA. He is currently a Research Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University. His research interests include distributed estimation and control, neural networks, multi-robotics, and big data.



YUNNI XIA (M'12–SM'15) received the B.S. degree in computer science from Chongqing University, Chongqing, China, in 2003, and the Ph.D. degree in computer science from Peking University, Beijing, China, in 2008. He joined the College of Computer Science, Chongqing University, in 2008, and is currently a Professor of Computer Science and Engineering. His research interests are in Petri nets, service computing, cloud computing, and Web services.