Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 525398, 9 pages
http://dx.doi.org/10.1155/2014/525398

Research Article

Matrix Factorization for Evolution Data

Xiao-Yu Huang,[1,2] Xian-Hong Xiang,[3] Wubin Li,[4] Kang Chen,[5] Wen-Xue Cai,[2] and Lei Li[1]

[1] *Software Institute, Sun Yat-Sen University, Guangzhou 510275, China*
[2] *School of Economics and Commerce, South China University of Technology, Guangzhou 510006, China*
[3] *Department of Interventional Radiology, The First Affiliated Hospital of Sun Yat-Sen University, Guangzhou 510080, China*
[4] *Department of Computing Science, Umeå University, 901 87 Umeå, Sweden*
[5] *Academy of Guangdong Telecom Co. Ltd., Guangzhou 510630, China*

Correspondence should be addressed to Xiao-Yu Huang; echxy@scut.edu.cn

Received 6 January 2014; Accepted 15 April 2014; Published 25 May 2014

Academic Editor: Chien-Yu Lu

We study a matrix factorization problem, that is, to find two factor matrices U and V such that $R \approx U^T \times V$, where R is a matrix composed of the values of the objects $O_1, O_2, \ldots, O_n$ at consecutive time points $T_1, T_2, \ldots, T_t$. We first present MAFED, a constrained optimization model for this problem, which straightforwardly performs factorization on R. Then based on the interplay of the data in U, V, and R, a probabilistic graphical model using the same optimization objects is constructed, in which structural dependencies of the data in these matrices are revealed. Finally, we present a fitting algorithm to solve the proposed MAFED model, which produces the desired factorization. Empirical studies on real-world datasets demonstrate that our approach outperforms the state-of-the-art comparison algorithms.

1. Introduction

The essence of the *matrix factorization* (MF) problem is to find two factor matrices U and V, such that their product can approximate a given matrix R; that is, $R \approx U^T V$. As a fundamental model in machine learning and data mining, MF methods have been widely used in various applications, such as collaborative filtering [1, 2], social network analysis [3], text analysis [4], image analysis [5, 6], and biology analysis [7].

In this work, we study a special variety of the MF problem. Let the matrix $R \in \mathbb{R}^{n \times t}$ consist of the values of the objects $O_1, O_2, \ldots, O_n$ at a serial of time points $T_1, T_2, \ldots, T_t$, where the entry $R_{i,j}$ is the value of O_i at T_j. Our goal is to find the suitable factor matrices U and V, which can not only approximate R, but also depict the *evolution* of the objects over time.

A typical application of this work is to estimate the missing historical traffic speed data, which is necessary for many transportation information systems [8–10]. Given an urban road network with n roads $L_1, L_2, \ldots, L_n$, we let O_i $(1 \le i \le n)$ correspond to the traffic speed of L_i and let entry $R_{i,j}$ be the speed of L_i at T_j; then R is composed of speed values of the roads $L_1 \sim L_n$ at the time $T_1 \sim T_t$. Especially, if the value of $R_{i,j}$ was not collected, we say it is *missing* and denote as $R_{i,j} =$ "$\perp$". With this representation, to estimate the missing values, we can first fit U and V with the nonmissing entries of R, and then for each $R_{i,j} =$ "$\perp$", take its estimation as $\widehat{R}_{i,j} = U_i^T V_j$, where U_i/V_j is the ith/jth column of U/V.

A main difference between this work and the other existing MF models is that we take the time evolution effects into account. Well-studied MF models, such as nonnegative matrix factorization [5], max margin matrix factorization [11, 12], and probabilistic matrix factorization [13], are based on the i.i.d assumption, which implicates that the behavior of the objects evolved with time is ignored and treated as being independent of time, and thus their abilities on describing time-varying data are poor. To tackle this issue, the factors that affect the evolution of the objects are explicitly modeled in our contribution, and that enables our model to interplay with the time dependent information. Furthermore, our empirical studies confirm that the proposed model can scale well with size of the data.

The remainder of this paper is organized as follows. Section 2 briefly summarizes the notations used in the paper. Section 3 reviews the works on matrix factorization. Section 4 presents our matrix factorization model as well as the statistical mechanism of the model. The proposed algorithm is introduced in Section 5. Section 6 is devoted to the analysis of the experiments. Our conclusions and future works are presented in Section 7.

2. Notations

For a vector $V = [v_1, v_2, \ldots, v_n]' \in \mathbb{R}^n$, we denote its 2-Norm as $\|V\|_2$, where

$$\|V\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}. \tag{1}$$

For a matrix $X \in \mathbb{R}^{n\times m}$, we let X_k be the kth column of X and denote its *Frobenius* norm as

$$\|X\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} X_{i,j}^2}. \tag{2}$$

3. Matrix Factorization

The essence of matrix factorization is to find the suitable factor matrices U and V, such that their product can approximate a given matrix R. In principle, the MF problem can be formulated as the optimization model below:

$$\{U^*, V^*\} = \min_{U,V} \text{Loss}\left(U^T V, R\right), \tag{M1}$$

where the function Loss is used to measure the *closeness* of the approximation $U^T V$ to the target R. In general, $\text{Loss}(U^T V, R)$ can be decomposed into the sum of the pairwise loss between the entries of $U^T V$ and R; that is, $\text{Loss}(U^T V, R) = \sum_{i=1}^{n}\sum_{j=1}^{m} \text{Loss}((U^T V)_{i,j}, R_{i,j})$. Most common used forms of the Loss function include the square loss ($\text{Loss}(x, y) = (x - y)^2$) [1, 13, 14], the 0-1 Loss ($\text{Loss}(x, y) = \mathbb{I}(x = y)$) [11], and the divergence loss ($\text{Loss}(x, y) = x \log(x/y) - x + y$) [6].

Notably, if $\{U^*, V^*\}$ is a solution of (M1), then for any scalar $\zeta > 0$, $\{\zeta U^*, (1/\zeta)V^*\}$ is another solution, and hence the problem (M1) is ill posed. To overcome this obstacle, various constraints on U and V are introduced, such as constraints on the entries [5], constraints on the sparseness [15, 16], constraints on the norms [13, 17], and constraints on the ranks [18, 19]. All these constraints, from the perspective of the statistical learning theory, can be regarded as the *length* of the model to be fitted. According to the minimum description length principle [20, 21], smaller length means better model, and thus most of them can be incorporated into (M1) as additional regularized terms; that is,

$$\{U^*, V^*\} = \min_{U,V} \text{Loss}\left(U^T V, R\right) + P(U, V), \tag{M2}$$

where the regularization factor $P(U, V)$ corresponds to the constraints on U and V.

As a transductive model, (M2) has many appealing mathematical properties, such as the generalization error bound [22] and the exactness [14, 23]. However, as well known, when compared with the generative model, a main weakness of the transductive model is that it can hardly be used to describe the relation in the data. In particular, for our studied problem, even though the model (M2) may work well, it is difficult to express the interplay between the model and the evolutions of the objects.

4. The Proposed Model

In this section the proposed model for matrix factorization for evolution data (MAFED) is presented. We first formalize MAFED as a constraint optimization model and then present a probabilistic graphical model, the solution of which is equivalent to the MAFED model. Finally we attain the generative interpretation of MAFED, with which we elaborate the ability of MAFED to describe the data relation. The last subsection is devoted to the solution algorithm for MAFED.

4.1. The Model. Let $R \in \mathbb{R}^{n\times t}$ be the matrix generated by n independent objects $O_1, O_2, O_3, \ldots, O_n$ at t consecutive time points $T_1, T_2, \ldots, T_t$; then our MAFED model can be formulated as follows:

$$\{U^*, V^*\} = \arg\min_{U,V} \sum_{i=1}^{n}\sum_{j=1}^{t}\left(R_{i,j} - U_i^T V_j\right)^2 \tag{3a}$$

$$\text{s.t.} \quad \|U_i\|_2 \leqslant A, \quad i = 1, 2, \ldots, n \tag{3b}$$

$$\|V_1\|_2 \leqslant B \tag{3c}$$

$$\|V_{j+1} - V_j\|_2 \leqslant C_j, \quad j = 1, 2, \ldots, t-1. \tag{3d}$$

Here U_i and V_j correspond to the ith column of U and the jth column of V, respectively.

From constraints (3b)–(3d), we can see that the roles of U and V in (3a) are asymptotic. We call U the object matrix, of which the ith column (U_i) is the invariant feature vector of O_i. Similarly, we call V the environment matrix, of which the jth column (V_j) is the time varying environment feature vector at T_j. Note that for each V_j, the range it takes effects in is global; that is, for each time point T_j, all the objects O_1,O_2,...,O_n share the same environment feature vector V_j. As a result, every entry $R_{i,j}$ ($1 \leq i \leq n, 1 \leq j \leq t$) is composed by the object's intrinsic feature vector U_i and the environment feature vector V_j, and hence for each object O_i, its evolutions over the time $T_1, T_2, \ldots, T_t$ are controlled by $T_1, T_2, \ldots, T_t$. To be more illustrative, let us consider such an example. Let R be the speed matrix of the roads $L_1, L_2, \ldots, L_n$ at the time points $T_1, T_2, \ldots, T_t$, let U be the feature matrix of the roads, and let V be the feature matrix of the environment; then U_i corresponds to the feature description of L_i and the entries of U_i may consist of the intrinsic features of the road, such as the surface rough, the number of lanes, and the role in the traffic network; similarly, for V, every V_j corresponds to the description of the environment at T_j, and the entries of V_j can be the time, the weather, the visibility, and so on. Therefore,

for the product $R_{i,j} = U_i^T V_j$, it can be interpreted as the speed of L_i at T_j is composed by the road feature U_i and the environment feature V_j.

The ability of MAFED to describe the time evolution effect lies in the constraint (3d). As well known, a fundamental characteristic of the evolution is for the all objects; when the interval between two adjacent time points, for example, T_j and T_{j+1}, is small enough, the corresponding values of the objects, that is, R_j and R_{j+1}, should tend to remain the same. To satisfy this constraint, in (3d) we introduce the tunable values $C_1, C_2, \ldots, C_{t-1}$. When $|t_j - t_{j+1}| \to 0$, we let $C_j \to 0$. Now since

$$\begin{aligned} \left\|R_j - R_{j+1}\right\|_2 &= \left\|U^T V_j - U^T V_{j+1}\right\|_2 \\ &\le \|U\|_2 \left\|V_j - V_{j+1}\right\|_2, \\ \|U\|_2 \le \|U\|_F &\le \sum_{i=1}^{n} \left\|U_i\right\|_2 \le n \times A, \end{aligned} \tag{4}$$

we have

$$\left\|R_j - R_{j+1}\right\|_2 \le n \times A \times \left\|V_j - V_{j+1}\right\|_2 \le n \times A \times C_j. \tag{5}$$

When n and A are fixed, $C_j \to 0$ leads to $n \times A \times C_j \to 0$, and so via (3d) it is guaranteed that $|t_j - t_{j+1}| \to 0 \Rightarrow ||R_j - R_{j+1}||_2 \to 0$.

On the other hand, when the time interval $|t_j - t_{j+1}| \to +\infty$, the values that R_j and R_{j+1} take will tend to be independently. For this case, we let $C_j \to +\infty$, indicating that R_{j+1} can take its value regardless of R_j.

With the constraints (3c) and (3d), we can control the value of $\|V_j\|_2 (j = 2, \ldots, t)$ via

$$\begin{aligned} \left\|V_j\right\|_2^2 &= \left\|V_1 + V_2 - V_1 + \cdots + V_j - V_{j-1}\right\|_2^2 \\ &\le \left\|V_1\right\|_2^2 + \left\|V_2 - V_1\right\|_2^2 + \cdots + \left\|V_j - V_{j-1}\right\|_2^2 \\ &\le B^2 + \sum_{k=1}^{j-1} C_k^2. \end{aligned} \tag{6}$$

The result above shows that we can bound the sum $\sum_{j=1}^{t} \|V_j\|_2^2$ by selecting appropriate parameters B and C. Besides, from the constraint (3b), the sum $\sum_{i=1}^{n} \|U_i\|_2^2$ is bounded by nA^2. According to [12, 24], under some suitable situations, the sum $1/2(\sum_{j=1}^{t} \|V_j\|_2^2 + \sum_{i=1}^{n} \|U_i\|_2^2)$ is the convex envelop of rank $(U^T V)$, and hence MAFED accommodates the ability to control the rank of the model. On the other side, as shown by Candès and Recht [14] and Candès and Tao [23], when the target matrix is *low rank*, it is possible to exactly recover it from only a few of its observations. This declaration shows that MAFED is possible to achieve high accuracy in the applications of missing imputations.

4.2. The Generative Interpretation. In this section, to investigate how MAFED interplays with the evolution of the data from the view of generative modeling, we first present a probabilistic graphical model (PGM [25]) and then show that the maximum likelihood solution to the proposed PGM is exactly the same as the solution of (3a).

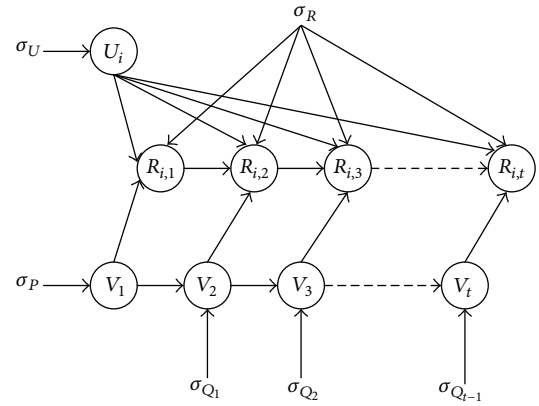

Figure 1: The generative model for MAFED.

First of all, the Lagrange dual of (3a) is

$$\begin{aligned} \{U^*, V^*\} = \arg\min_{U,V} &\sum_{i=1}^{n}\sum_{j=1}^{t} \left(R_{i,j} - U_i^T V_j\right)^2 \\ &+ \alpha \sum_{i=1}^{n} \left\|U_i\right\|_2^2 + \beta \left\|V_1\right\|_2^2 + \sum_{j=1}^{t-1} \lambda_j \left\|V_{j+1} - V_j\right\|_2^2 \\ \text{s.t.} \quad & \alpha, \beta, \lambda_1, \lambda_2, \ldots, \lambda_{t-1} \ge 0. \end{aligned} \tag{7}$$

Here α, β, and λ_i $(i = 1, 2, \ldots, t-1)$ are tunable parameters, corresponding to the upper bounds A, B, and C_i $(i = 1, 2, \ldots, t-1)$ of constraints (3b)–(3d), respectively, where greater bounds correspond to smaller parameters, and vise versa.

Considering the PGM presented in Figure 1, we have the following assumptions for U, V, and R.

(1) The columns of U are from the same Gaussian distribution with mean 0 and covariance matrix $\sigma_U^2 I$; that is, for $1 \le i \le n$, we have (here we use d to denote the dimension of U_i)

$$\Pr\left(U_i \mid \sigma_U\right) = \left(2\pi\sigma_U^2\right)^{-d/2} \exp\left\{-\frac{\left\|U_i\right\|_2^2}{2\sigma_U^2}\right\}. \tag{8}$$

(2) The columns of V are linearly dependent in the order of their subscripts with respect to prespecified priors σ_P and $\sigma_{Q_1}, \sigma_{Q_2}, \ldots, \sigma_{Q_{T-1}}$; that is,

$$\begin{aligned} \Pr\left(V \mid \sigma_P, \sigma_\Lambda\right) &= \Pr\left(V_1 \mid \sigma_P, \sigma_\Lambda\right) \\ &\times \prod_{j=2}^{T} \Pr\left(V_j \mid V_{j-1}, \sigma_P, \sigma_\Lambda\right). \end{aligned} \tag{9}$$

Here for clarity, we let $\Lambda = \{Q_1, Q_2, \ldots, Q_{t-1}\}$ and $\sigma_\Lambda = \{\sigma_{Q_1}, \sigma_{Q_2}, \ldots, \sigma_{Q_{T-1}}\}$.

We also assume V_1 is a Gaussian random vector with mean $\mathbf{0}$ and covariance $\sigma_P^2 I$ and V_j $(j > 1)$ is a Gaussian random vector with mean V_{j-1} and covariance $\sigma_{Q_{j-1}}^2 I$; that is,

$$\begin{aligned}
&\Pr\left(V_1 \mid \sigma_P, \sigma_\Lambda\right) = \left(2\pi\sigma_P^2\right)^{-d/2} \exp\left\{-\frac{\|V_1\|_2^2}{2\sigma_P^2}\right\},\\
&\Pr\left(V_j \mid V_{j-1}, \sigma_P, \sigma_\Lambda\right)\\
&\quad = \left(2\pi\sigma_{Q_{j-1}}\right)^{-d/2} \exp\left\{-\frac{\|V_j - V_{j-1}\|_2^2}{2\sigma_{Q_{j-1}}^2}\right\}.
\end{aligned} \tag{10}$$

(3) The (i, j)th entry of R $(1 \le i \le N, 1 \le j \le T)$ is a Gaussian random variable with mean $U_i^T V_j$ and variance σ_R^2; that is,

$$\Pr\left(R_{i,j} \mid U_i^T V_j, \sigma_R^2\right) = \left(2\pi\sigma_R^2\right)^{-1/2} \exp\left\{-\frac{\left(R_{i,j} - U_i^T V_j\right)^2}{2\sigma_R^2}\right\}. \tag{11}$$

With these assumptions, we in the following part of this section show that, for a given matrix R and priors σ_U, σ_P, σ_Λ, σ_R, model (7) is equivalent to the following maximum likelihood fitting problem:

$$\left\{U^*, V^*\right\} = \arg\max_{U,V} \Pr\left(U, V \mid R, \sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R\right). \tag{12}$$

In fact, according to the Bayesian theorem, we have

$$\Pr\left(U, V \mid R, \sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R\right) = \frac{\Pr\left(U, V, R \mid \sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R\right)}{\Pr\left(R \mid \sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R\right)}. \tag{13}$$

Since R is observed and σ_U, σ_P, σ_Λ, and σ_R are prespecified, the denominator $\Pr(R|\sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R)$ can be treated as a constant. Therefore, we get

$$(12) \Longleftrightarrow \left\{U^*, V^*\right\} = \arg\max_{U,V} \Pr\left(U, V, R \mid \sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R\right). \tag{14}$$

Combining Figure 1 and the assumptions (1) ~ (3), we have

$$\begin{aligned}
&\Pr\left(R, U, V \mid \sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R\right)\\
&\quad = \Pr\left(R \mid U, V, \sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R\right)\\
&\qquad \times \Pr\left(U, V \mid \sigma_U, \sigma_P, \sigma_\Lambda, \sigma_R\right)\\
&\quad = \Pr\left(R \mid U, V, \sigma_R\right) \times \Pr\left(U \mid \sigma_U\right) \times \Pr\left(V \mid \sigma_P, \sigma_\Lambda\right)\\
&\quad = \prod_{i=1}^{n}\prod_{j=1}^{t} \Pr\left(R_{i,j} \mid U_i, V_j, \sigma_R\right)\\
&\qquad \times \prod_{i=1}^{n} \Pr\left(U_i \mid \sigma_U\right) \times \Pr\left(V_1 \mid \sigma_P\right)\\
&\qquad \times \prod_{j=2}^{t} \Pr\left(V_j \mid V_{j-1}, \sigma_\Lambda\right)\\
&\quad \propto \exp\left(-\frac{1}{2\sigma_R^2}\sum_{i=1}^{n}\sum_{j=1}^{t}\left(U_i^T V_j - R_{i,j}\right)^2\right)\\
&\qquad \times \exp\left(-\frac{1}{2\sigma_U^2}\sum_{i=1}^{t}\|U_i\|_2^2\right)\\
&\qquad \times \exp\left(-\frac{1}{2\sigma_P^2}\|V_1\|_2^2\right)\\
&\qquad \times \exp\left(-\sum_{j=1}^{t-1}\frac{1}{2\sigma_{Q_j}^2}\|V_{j+1} - V_j\|_2^2\right).
\end{aligned} \tag{15}$$

Taking the logarithm on the two sides of the equation, we get

$$\begin{aligned}
(14) \Longleftrightarrow \left\{U^*, V^*\right\} &= \arg\min_{U,V} \frac{1}{\sigma_R^2}\\
&\times \sum_{i=1}^{n}\sum_{j=1}^{t}\left(R_{i,j} - U_i^T V_j\right)^2 + \frac{1}{\sigma_U^2}\sum_{i=1}^{N}\|U_i\|_2^2\\
&+ \frac{1}{\sigma_P^2}\|V_1\|_2^2 + \sum_{j=1}^{t-1}\frac{1}{\sigma_{Q_j}^2}\|V_{j+1} - V_j\|_2^2.
\end{aligned} \tag{16}$$

Comparing (7) and (16), the PGM model is equivalent to the optimization model (3a) if we let $\alpha = \sigma_R^2/\sigma_U^2$, $\beta = \sigma_R^2/\sigma_P^2$, and $\lambda_j = \sigma_R^2/\sigma_{Q_j}^2$ $(j = 1, 2, \ldots, t-1)$.

5. The Algorithm

We in this section present a fitting algorithm to solve (7). For clarity, we only discuss the case that all the time intervals are of equal length; that is, $T_2 - T_1 = T_3 - T_2 = \cdots = T_t - T_{t-1}$. Under this hypothesis, we take $\lambda_1 = \lambda_2 = \cdots = \lambda_{t-1}$ in model (7). Cases with unequal length time intervals can be handled by removing the constraint $\lambda_i = \lambda_j$ if $T_{i+1} - T_i = T_{j+1} - T_j$.

Let

$$\begin{aligned}
S = \arg\min_{U,V} &\sum_{i=1}^{n}\sum_{j=1}^{t}\left(R_{i,j} - U_i^T V_j\right)^2\\
&+ \alpha\sum_{i=1}^{n}\|U_i\|_2^2 + \beta\|V_1\|_2^2 + \lambda\sum_{j=1}^{t-1}\|V_{j+1} - V_j\|_2^2.
\end{aligned} \tag{17}$$

It is straightforward to verify that S is convex with respect to U_i $(1 \le i \le n)$ and V_j $(1 \le j \le t)$, respectively. Therefore we

can achieve the local minimum solution of S via coordinate descent [26]. We first calculate the partial derivative with respect to $U_1, U_2, \ldots, U_n$ and $V_1, V_2, \ldots, V_t$, respectively. Then we have the following.

For $i = 1, 2, \ldots, n$,

$$\frac{\partial S}{\partial U_i} = 2\alpha U_i - 2\sum_{j=1}^{t}\left(R_{ij} - U_i^T V_j\right)V_j \tag{18}$$

and for $j = 2, 3, \ldots, t-1$,

$$\frac{\partial S}{\partial V_j} = 2\lambda\left(2V_j - V_{j-1} - V_{j+1}\right) - 2\sum_{i=1}^{n}\left(R_{i,j} - U_i^T V_j\right)U_i. \tag{19}$$

Similarly,

$$\begin{aligned}\frac{\partial S}{\partial V_1} &= 2\beta V_1 - 2\sum_{i=1}^{n}\left(R_{i,1} - U_i^T V_1\right)U_i - 2\lambda\left(V_2 - V_1\right),\\ \frac{\partial S}{\partial V_t} &= 2\lambda\left(V_t - V_{t-1}\right) - 2\sum_{i=1}^{n}\left(R_{i,t} - U_i^T V_t\right)U_i.\end{aligned} \tag{20}$$

With the equations above, Algorithm 1 is presented.

6. Experiments

In this section, experimental evaluations against a finance dataset and a traffic speed dataset are presented. The proposed MAFED algorithm is compared with the other two state-of-the-art approaches.

6.1. Evaluation Methodology. To evaluate the algorithms, we perform missing imputation on an incomplete matrix R. Similar to many other matrix competition problems, the testing protocol adopted is the *Given X* ($0 < X < 1$) protocol [27]; that is, for R, we only show X of its observed entries to the users, while holding the remaining $(1 - X)$ observations to evaluate the trained model. For example, when $X = 10\%$, *Given X* means that the algorithm is trained with 10% of the nonmissing entries and the rest of 90% nonmissing ones are held and to be recovered.

With the *Given X* setting, (16) can be rewritten as

$$\begin{aligned}\{U^*, V^*\} = \arg\min_{U,V} &\sum_{i=1}^{N}\sum_{j=1}^{T}\left(R_{i,j} - U_i^T V_j\right)^2 \mathbb{I}\left(R_{i,j} \neq \text{“}\perp\text{”}\right)\\ &+ \alpha\sum_{i=1}^{N}\|U_i\|_2^2 + \beta\|V_1\|_2^2 + \lambda\sum_{j=2}^{T}\|V_j - V_{j-1}\|_2^2.\end{aligned} \tag{21}$$

The first selected comparison algorithm is the probabilistic principle component analysis model (PPCA [28]), which achieves the state-of-the-art performance in the missing traffic flow data imputation problem [29]. The second is the probabilistic matrix factorization model (PMF), which is one of the most popular algorithms in the Netflix matrix completion problem [13].

The evaluation criterion employed is the root mean square error (RMSE). More formally, for a test dataset $S = \{s_1, s_2, \ldots, s_n\}$ and the estimated set $\widehat{S} = \{\widehat{s}_1, \widehat{s}_2, \ldots, \widehat{s}_n\}$ ($\widehat{s}_i$ is the estimation of s_i), the RMSE of the estimation is given by $\sqrt{(1/n)\sum_{k=1}^{n}(s_k - \widehat{s}_k)^2}$.

In all our experiments, for every *Given X*, the data partition is repeated for 5 times, and the average results as well as standard deviations are recorded.

6.2. Experiments on the Finance Dataset. The finance dataset is a 524-*by*-245 matrix R, which consists of the opening prices of 524 leading U.S. companies in 245 consecutive trading days from August 21, 2009 to August 20, 2010. Each row in R corresponds to a company and each column corresponds to a day.

To characterize the evolution of the opening prices, we introduce the term *incremental rate*; for the ith company, the *incremental rate* of its opening price in the jth day is given by

$$\text{Inc}_{i,j} = \frac{R_{i,j} - R_{i,j-1}}{R_{i,j-1}}. \tag{22}$$

Intuitively, the *incremental rate* quantifies how *strong* the evolution is. A smaller $|\text{Inc}_{i,j}|$ implies a closer connection between $R_{i,j-1}$ and $R_{i,j}$, and hence we say the evolution from $R_{i,j-1}$ to $R_{i,j}$ is *gradual*; while a larger $|\text{Inc}_{i,j}|$ indicates a looser relation between $R_{i,j}$ and $R_{i,j-1}$, and thus the evolution is considered to be *saltatory*.

We calculate the incremental rates for all companies in the days $2 \sim 245$. Figure 2 illustrates the cumulative probability distribution of these rates. We can observe that almost all the incremental rates locate in a very sharp interval $(-10\%, 10\%)$. This implies that the changes of the opening prices are very slight. In other words, the evolutions of the opening prices are *gradualism*.

We evaluate the parametric sensitivity of the algorithm by tuning the parameters α and λ. In our experiments, we first fix $\alpha = 0.01$ and vary λ via an assignment expression $\lambda = 0.01 \times 2^n$, where $n = 0, 1, \ldots, 9$. Then we do the reverse by fixing $\lambda = 0.01$ and changing α via $\alpha = 0.01 \times 2^n$. For the *Given X* protocol setting, we take $X = 50\%$; that is, 50% of the data in R is randomly selected as training data, with which the algorithm is trained and recovers the remaining 50% as test data. The average RMSEs of the experiments are shown in Figure 3.

As shown in Figure 3, the RMSE values remain stable even when λ is expanded by more than 200 times ($2^8 = 256$). Similar result also appears in the experiments on the parameter α. We can observe that significant changes of the RMSE value only occur in cases where $n > 7$ (i.e., α is expanded by more than 128 times).

To study the prediction ability of the proposed algorithm, we increase the X value in the *Given X* protocol from 10% to 50%. Then for each X setting, missing imputations are performed using MAFED with $\alpha = \beta = \lambda = 0.5$ and the comparison algorithms with the latent feature number $d = 10$ and $d = 30$, respectively. As illustrated in Table 1, MAFED outperforms PPCA significantly in all settings. For

Input: matrix R; number of the latent features d; learning rates η_1, η_2 and η_3; regularization parameters α, β and λ; threshold ϵ
Output: the estimated matrix $U^T V$.
// `Initialize` U `and` V.
(1) Generate random vectors $U_1, U_2, \ldots, U_n, V_1 \sim \mathbf{N}(\mathbf{0}, I)$;
(2) **for** $j = 2$; $j \leq t$; $j++$ **do**
(3) Generate Z with $\mathbf{N}(\mathbf{0}, I)$
(4) Let $V_j = V_{j-1} + Z$
(5) **end**
// `Coordinate descent`.
(6) $S_1 = \sum_{i=1,j=1}^{i=n,j=t} \left(R_{i,j} - U_i^t V_j\right)^2 + \alpha \sum_{i=1}^{n} \|U_i\|_2^2 + \beta \|V_1\|_2^2 + \lambda \sum_{j=1}^{t-1} \|V_{j+1} - V_j\|_2^2$;
(7) $S_2 = \inf$;
(8) **while** $|S_2 - S_1| > \epsilon$ **do**
(9) $S_2 = S_1$;
(10) **for** $i = 1, 2, \ldots, n$ **do**
(11) Let $U_i^{\text{new}} = U_i - \eta_1 (\partial S / \partial U_i)$;
(12) **end**
(13) $V_1^{\text{new}} = V_1 - \eta_2 \dfrac{\partial S}{\partial V_1}$;
(14) **for** $j = 2, \ldots, t$ **do**
(15) Let $V_j^{\text{new}} = V_j - \eta_3 \left(\partial S / \partial V_j\right)$;
(16) **end**
(17) Replace the all $U_i s$ with $U_i^{\text{new}} s$, and $V_j s$ with $V_j^{\text{new}} s$, recompute S_1;
(18) **end**
(19) **return** $U^T V$;

Algorithm 1: MAFED($R, d, \eta_1, \eta_2, \eta_3, \alpha, \beta, \lambda, \epsilon$).

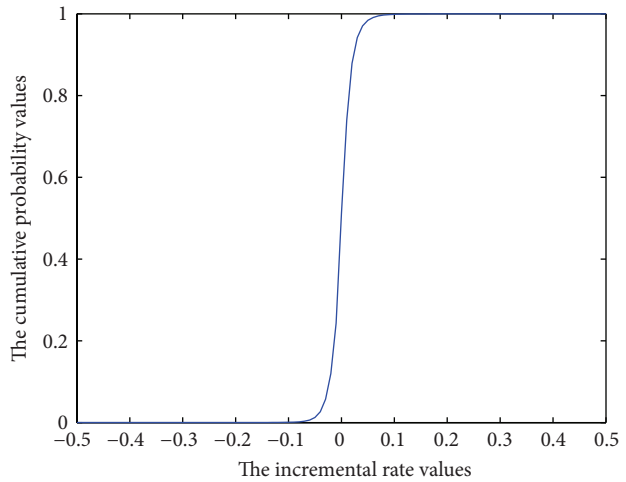

Figure 2: The cumulative probability distribution curve of the incremental rates of the finance dataset.

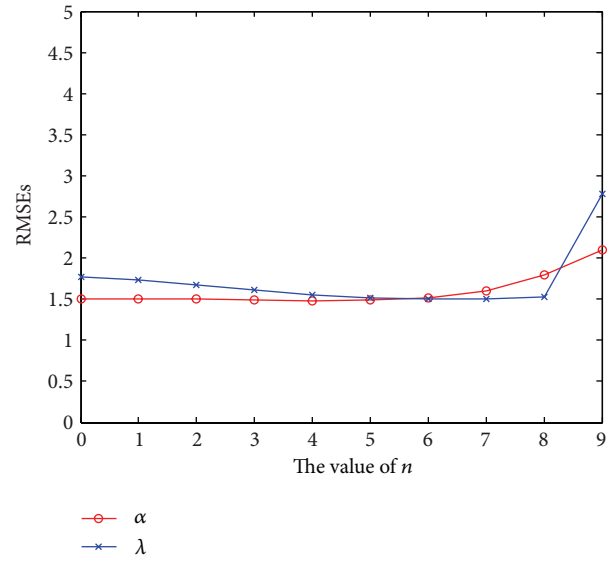

Figure 3: Empirical results on parameter sensitivity.

any X value, the RMSE of MAFED is at most 20% of that of PPCA. Specifically, for $X \in \{30\%, 40\%, 50\%\}$, the RMSE of MAFED is even only 10% of PPCA. More notably, the value d has strikingly different impacts on MAFED and PPCA. When d changes from 10 to 30, almost all the RMSEs of PPCA dramatically increase, while for MAFED, RMSEs are decreased by nearly 5% except for the case with $X = 10\%$. This observation suggests that, with carefully setting of the parameter d, it is possible to improve the predicting accuracy of MAFED. We can also observe that, compared with PMF, MAFED achieves smaller RMSEs in the all settings. Especially, when $X \in \{30\%, 40\%, 50\%\}$, the RMSE of MAFED is only half of that of PMF. Another interesting

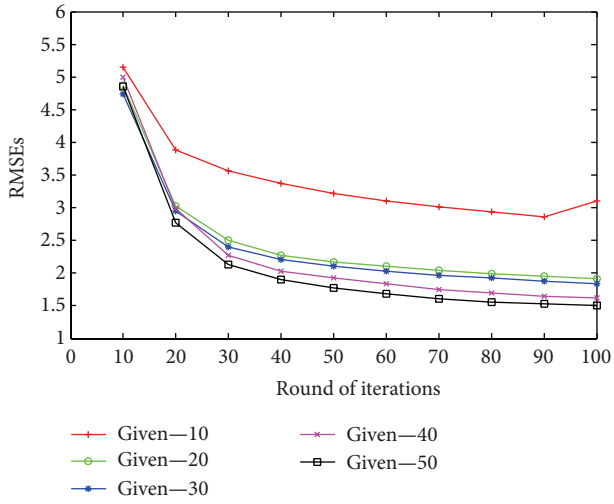

Figure 4: Empirical results on the algorithm convergence.

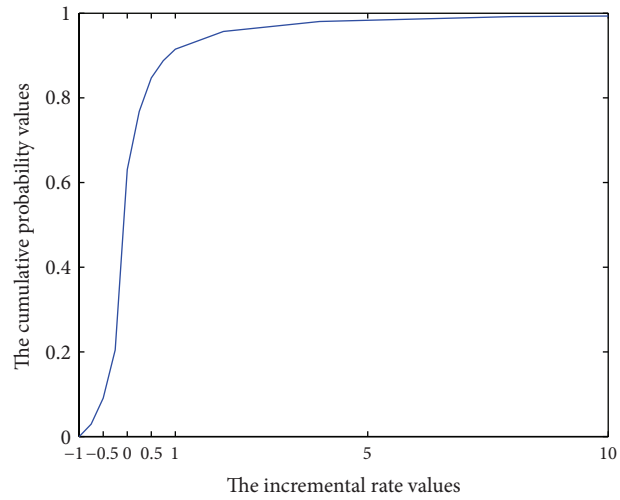

Figure 5: The cumulative probability distribution curve of the incremental rates of the traffic speed dataset.

finding is that, for PMF, the value of d has little impact on predictions..

The convergence rate of MAFED is also explored in this work. We record the RMSEs of data recovered by MAFED every 10 iterations and plot them on Figure 4, where the x-axis is the number of iterations and y-axis represents the RMSEs. We can observe that, for all X values, corresponding curves drop dramatically in the first 20 iterations and remain stable afterwards. From the result presented in Figure 3 and Table 1, we can conclude that, in our experiments, MAFED converges to the local optimization solutions within 100 iterations.

6.3. Experiments on the Traffic Speed Dataset. To reconfirm the recovery performance of MAFED, we in this section conduct another evaluation on a traffic speed dataset collected in the urban road network of Zhuhai City [30], China, from April 1, 2011 to April 30, 2011. Again we adopt a matrix R to represent the data, where R consists of 1853 rows and 8729 columns. Each row corresponds to a road and each column corresponds to a 5-minute-length time interval. All columns are arranged in ascending order of time. The entry $R_{i,j}$ ($1 \leq i \leq 1853$, $1 \leq j \leq 8729$) in R is the aggregate mean traffic speed of the ith road in the jth interval. As the data in R are collected by probing vehicles, the value of $R_{i,j}$ might be *missing* if there is no probing vehicle on the ith road during the jth time interval. In the data set, nearly half of the data, that is, 8 million entries in R, are such missing values.

We calculate the incremental rates of the nonmissing data in R and present the cumulative density distribution for the incremental rates in Figure 5. The evolution of the traffic speeds differs from that of the stock opening prices remarkably. As aforementioned, the incremental rates of the finance data set mainly concentrate on the sharp interval [−10%, 10%]. While for the traffic speed data, as demonstrated in Figure 5, the range of the incremental rates spreads from −1 to 10. This indicates that there are plenty of *sudden changes* of the road speeds occuring in adjacent time intervals. In other words, many of the speed evolutions are *saltation*. A possible cause of this observation could be the effects of the traffic lights in the urban road network. When the traffic light of a road turns from green to yellow (and then red), traffic speed on the road immediately reduces to 0. On the other side, when the light turns from red to green, vehicles are restarted instantly, resulting in a significant acceleration of the speed on the road.

This experiment is conducted with settings $d = 10$, $\alpha = 0.5$, and $\lambda = 10$. Table 2 shows that in the all X settings, MAFED also outperforms PPCA and PMF. In particular, when there are few observations (e.g., $X = 10\%$ and $X = 20\%$), RMSEs of MAFED are 33% lower than those of PPCA and 10% lower than those of PMF. When $X > 20\%$, the RMSE differences between PPCA and MAFED tend to be slight. Despite that, the overall errors of PPCA are roughly 3% ~ 5% higher than those of MAFED, and for PMF, the RMSEs remain roughly 10% higher than those of MAFED.

7. Conclusion and Future Works

Matrix factorization models are fundamental in machine learning and data mining. In this paper, we present MAFED, a matrix factorization model for evolution data. In MAFED, the two factor matrices are treated as composed of intrinsic features of the objects and time-varying features of the environment, respectively. Hence, their product accommodates the ability to describe the evolution of the objects over time. Besides, we construct a probabilistic graphical model, with which the statistical natural of MAFED is elaborated. We also present a fitting algorithm to find the solution of MAFED. Finally, we evaluate MAFED through missing data imputation on two real-world datasets. Experimental results indicate that the proposed model outperforms the comparison algorithms in both gradualism cases and saltation cases.

According to the generative interpretation of MAFED, its success mainly attributes to the introduction of the linear

Table 1: Imputation results on the finance dataset.

		10%	20%	30%	40%	50%
$D = 10$	PPCA	18.52 ± 0.73	20.84 ± 0.81	24.18 ± 0.65	22.82 ± 0.59	19.57 ± 0.33
	PMF	3.33 ± 0.03	3.29 ± 0.03	3.30 ± 0.02	3.30 ± 0.00	3.28 ± 0.02
	MAFED	**3.14** ± 0.01	**2.92** ± 0.01	**2.84** ± 0.02	**2.32** ± 0.01	**2.05** ± 0.02
$D = 30$	PPCA	24.22 ± 0.61	21.84 ± 0.93	24.51 ± 0.51	23.22 ± 0.37	22.61 ± 0.35
	PMF	3.33 ± 0.03	3.29 ± 0.02	3.30 ± 0.01	3.30 ± 0.02	3.29 ± 0.02
	MAFED	**3.11** ± 0.03	**2.87** ± 0.02	**2.70** ± 0.00	**2.21** ± 0.01	**2.09** ± 0.00

Table 2: Imputation results on the traffic dataset.

	10%	20%	30%	40%	50%
PPCA	17.88 ± 0.33	17.36 ± 0.35	12.00 ± 0.26	12.26 ± 0.12	11.47 ± 0.06
PMF	14.44 ± 0.10	12.75 ± 0.11	12.49 ± 0.07	12.39 ± 0.06	12.36 ± 0.05
MAFED	**13.30** ± 0.09	**12.17** ± 0.10	**11.92** ± 0.07	**11.84** ± 0.06	**11.04** ± 0.08

chain structural prior; as a result, a natural extension of the present work is to deal with the factorizations on matrices with more complex structural priors. In particular, as have been shown in the empirical studies section, many of the pieces of data in real-world scenarios are highly incomplete, so it is important and interesting to study the factorization on the incomplete matrix with no prespecified structural information.

Conflict of Interests

The authors declare that there is no conflict of interests regerding the publication of this paper.

Authors' Contribution

Huang and Xiang contributed equally and Huang corresponds to this paper.

Acknowledgments

This work is supported in part by National High-Tech R&D Program (863 Program) of China under Grant no. 2012AA12A203.

References

[1] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 426–434, August 2008.

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

[3] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using Bayesian personalized ranking for social network data," in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM '12)*, pp. 173–182, February 2012.

[4] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 267–273, 2003.

[5] 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, 1999.

[6] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[7] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 12, pp. 4164–4169, 2004.

[8] S. Turner, L. Albert, B. Gajewski, and W. Eisele, "Archived intelligent transportation system data quality: preliminary analyses of San Antonio Transguide data," *Transportation Research Record*, no. 1719, pp. 77–84, 2000.

[9] B. L. Smith, W. T. Scherer, and J. H. Conklin, "Exploring imputation techniques for missing data in transportation management systems," *Transportation Research Record*, no. 1836, pp. 132–142, 2003.

[10] D. Ni, J. D. Leonard, A. Guin, and C. Feng, "Multiple imputation scheme for overcoming the missing values and variability issues in ITS data," *Journal of Transportation Engineering*, vol. 131, no. 12, pp. 931–938, 2005.

[11] N. Srebro, J. D. Rennie, and T. Jaakkola, "Maximum-margin matrix factorization," *Advances in Neural Information Processing Systems*, vol. 17, no. 5, pp. 1329–1336, 2005.

[12] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 713–720, August 2005.

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

[14] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[15] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[17] Q. Ke and T. Kanade, "Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 739–746, June 2005.

[18] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proceedings of the 20th International Conference on Machine Learning*, vol. 20, p. 720.

[19] J. Abernethy, F. Bach, T. Evgeniou, and J. P. Vert, "Low-rank matrix factorization with attributes," http://arxiv.org/abs/cs/0611124.

[20] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[21] J. Rissanen, *Minimum Description Length Principle*, Springer, New York, NY, USA, 2010.

[22] N. Srebro, *Learning with matrix factorizations [Ph.D. thesis]*, Citeseer, 2004.

[23] E. J. Candès and T. Tao, "The power of convex relaxation: near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[24] M. Fazel, H. Hindi, and S. P. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *Proceedings of the American Control Conference*, pp. 4734–4739, June 2001.

[25] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, Cambridge, Mass, USA, 2009.

[26] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Nashua, NH, USA, 1999.

[27] B. Marlin, *Collaborative filtering: a machine learning perspective [Ph.D. thesis]*, University of Toronto, 2004.

[28] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society B: Statistical Methodology*, vol. 61, no. 3, pp. 611–622, 1999.

[29] L. Qu, L. Li, Y. Zhang, and J. Hu, "PPCA-based missing data imputation for traffic flow volume: a systematical approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 512–522, 2009.

[30] Zhuhai City, China, http://en.wikipedia.org/wiki/Zhuhai .