

Article

Multi-Matrices Factorization with Application to Missing Sensor Data Imputation

Xiao-Yu Huang ^{1,2,*}, Wubin Li ³, Kang Chen ⁴, Xian-Hong Xiang ⁵, Rong Pan ¹, Lei Li ¹ and Wen-Xue Cai ²

¹ Software Institute, Sun Yat-Sen University, Guangzhou 510275, China;

E-Mails: panr@mail.sysu.edu.cn (R.P.); lnsllilei@mail.sysu.edu.cn (L.L.)

² School of Economics and Commerce, South China University of Technology,

Guangzhou 510006, China; E-Mail: ctwxcai@scut.edu.cn

³ Department of Computing Science, Umeå University, SE-901 87 Umeå, Sweden;

E-Mail: wubin.li@cs.umu.se

⁴ Academy of Guangdong Telecom Co.Ltd, Guangzhou 510630, China; E-Mail: chenkang@gsta.com

⁵ Department of Interventional Radiology, the First Affiliated Hospital of Sun Yat-Sen University, Guangzhou 510080, China; E-Mail: med.interventional@163.com

* Author to whom correspondence should be addressed; E-Mail: echxy@scut.edu.cn;

Tel.: +86-153-2200-5312.

Received: 7 October 2013; in revised form: 28 October 2013 / Accepted: 28 October 2013 /

Published: 6 November 2013

Abstract: We formulate a multi-matrices factorization model (MMF) for the missing sensor data estimation problem. The estimation problem is adequately transformed into a matrix completion one. With MMF, an n -by- t real matrix, R , is adopted to represent the data collected by mobile sensors from n areas at the time, $T_1, T_2, \dots, T_t$, where the entry, $R_{i,j}$, is the aggregate value of the data collected in the i th area at T_j . We propose to approximate R by seeking a family of d -by- n probabilistic *spatial feature matrices*, $U_{(1)}, U_{(2)}, \dots, U_{(t)}$, and a probabilistic *temporal feature matrix*, $V \in \mathbb{R}^{d \times t}$, where $R_j \approx U_{(j)}^T V$. We also present a solution algorithm to the proposed model. We evaluate MMF with synthetic data and a real-world sensor dataset extensively. Experimental results demonstrate that our approach outperforms the state-of-the-art comparison algorithms.

Keywords: matrix factorization; sensor data; probabilistic graphical model; missing estimation

1. Introduction

In this work, we study the following missing sensor data imputation problem: Let the matrix, $R \in \mathbb{R}^{n \times t}$, consist of the data collected by a set of mobile sensors in spacial areas $S_1, S_2, \dots, S_n$ at time points $T_1 < T_2 < \dots < T_t$, where the entry, $R_{i,j}$, is the aggregate value collected by the sensors in S_i at T_j . In particular, if there is no sensor in S_i at time T_j , we denote the value of $R_{i,j}$ as “ $\perp$ ”, which indicates that it is *missing*. Our focus is to find the suitable estimations for the missing values in a given incomplete matrix, R . Results of this research could be helpful in recovering missing values in statistical analyses. For example, to predict floods, people usually place geographically distributed sensors in the water to continuously monitor the rising water levels. However, some data in a critical period of time might be corrupted, due to, e.g., sensor hardware failures. Such a kind of data needs be recovered to guarantee the prediction accuracy.

Many efforts have been devoted to the missing sensor data imputation problem. Typical examples include k nearest neighbor-based imputation [1], multiple imputation [2], hot/cold imputation [3], maximum likelihood and Bayesian estimation [4] and expectation maximization [5]. However, despite the various implementations of these methods, their main essence is based on the *local consistency* of the sensor data, *i.e.*, the data collected at adjacent time points within the same spacial area should be close to each other, as well as the data collected at the same time from neighboring areas. We refer to them as *local models*. As is well known, these local models suffer from the cumulative error problem in scenarios where the missing ratio is high.

Matrix factorization (MF), as a *global model*, has caught substantial attention in recent years. Typically, in the *Netflix* rating matrix completion competition [6], some variations of the MF model, e.g., [7,8], achieved state-of-the-art performances, showing their potential to recover the missing data from highly incomplete matrices. On the other side, many well-studied MF models, such as non-negative matrix factorization [9], max margin matrix factorization [10,11], and probabilistic matrix factorization [7], are based on the i.i.d. assumption [12], which, in terms of our problem, implies that the neighborhood information among the data is disregarded and, hence, leaves vast room for improvement.

We in this paper, we propose a multi-matrices factorization model (MMF), which can be outlined as follows. For a matrix, X , denote X_j the j th column of X . Given a sensor data matrix, R , we seek a set of matrices, $U_{(1)}, U_{(2)}, \dots, U_{(t)} \in \mathbb{R}^{d \times n}$, and a matrix, $V \in \mathbb{R}^{d \times t}$, such that for $i = 1, 2, \dots, t$, $U_{(i)}^T V_i \approx R_i$. Here, $U_{(i)}$ is referred to as the *spatial feature matrix*, in which the j th column, $U_{(i),j}$, is the feature vector of area S_j at T_i . Similarly, V is referred to as the *temporal feature matrix*, in which the j th column, V_j , is the temporal feature vector of T_j . To predict the missing values in R , we first fit the matrices, $U_{(1)}, U_{(2)}, \dots, U_{(t)}$ (single sub-indexes of matrix mean columns) and V , with the non-missing values in R ; then, for each $R_{i,j} = \perp$, we take its estimation as $\hat{R}_{i,j} = U_{(j),i}^T V_j$.

The remainder of the paper is organized as follows: Section 2 summarizes the notations used in the paper. Section 3 studies the related work on matrix factorization. In Section 4, we present our multi-matrices factorization model. The algorithm to solve the proposed model is outlined in Section 5. Section 6 is devoted to the experimental evaluations. Finally, our conclusions are presented in Section 7, followed by a presentation of future work, acknowledgments and a list of references.

2. Notations

For a vector $V = [v_1, v_2, \dots, v_n]' \in \mathbb{R}^n$, we use $\|V\|_0$, $\|V\|_1$ and $\|V\|_2$ to denote its 0-Norm, 1-Norm and 2-Norm, respectively, as follows:

- $\|V\|_0 = \sum_{i=1}^n \mathbb{I}(v_i \neq 0)$, i.e., the number of nonzero entries of V ;
- $\|V\|_1 = \sum_{i=1}^n |v_i|$;
- $\|V\|_2 = \sqrt{\sum_{i=1}^n v_i^2}$.

For a matrix, $X \in \mathbb{R}^{n \times m}$, we denote its *Frobenius* norm as $\|X\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m X_{i,j}^2}$.

3. Matrix Factorization

The essence of the *Matrix Factorization* problem is to find two factor matrices, U and V , such that their product can approximate the given matrix, R , i.e., $R \approx U^T V$. As a fundamental model of machine learning and data mining, the MF method has achieved state-of-the-art performance in various applications, such as collaborative filtering [13], text analysis [14], image analysis [9,15] and biology analysis [16]. In principle, for a given matrix, R , the MF problem can be formulated as the optimization model below:

$$\{U^*, V^*\} = \min_{U, V} \text{Loss}(U^T V, R) \quad (1)$$

where the lose function, *Loss*, is used to measure the *closeness* of the approximation, $U^T V$, to the target, R . Usually, $\text{Loss}(U^T V, R)$ can be decomposed into the sum of the pairwise loss between the entries of $U^T V$ and R ; that is, $\text{Loss}(U^T V, R) = \sum_{i=1}^n \sum_{j=1}^m \text{loss}((U^T V)_{i,j}, R_{i,j})$. Some of the most used forms include the square loss ($\text{loss}(x, y) = (x - y)^2$) [7,8,17], the 0-1 loss ($\text{loss}(x, y) = \mathbb{I}(x = y)$) [11] and the divergence loss ($\text{loss}(x, y) = x \log \frac{x}{y} - x + y$) [9].

It is notable that for Equation (1), if $\{U^*, V^*\}$ is a solution to it, then for any scalar, $\kappa > 0$, $\{\kappa U^*, \frac{1}{\kappa} V^*\}$ is also another solution; hence, problem (1) is ill-posed. To overcome this obstacle, various constraints on U and V are introduced, such as constraints on the entries [15], constraints on the sparseness [18,19], constraints on the norms [7,20] and constraints on the ranks [21,22]. 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 [23,24], a smaller length means a better model; hence, most of them can be incorporated into Model (1) as the additional regularized terms, that is:

$$\{U^*, V^*\} = \min_{U, V} \text{Loss}(U^T V, R) + P(U, V) \quad (2)$$

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

As a transductive model, Model (2) has many nice mathematical properties, such as the generalization error bound [10] and the exactness [17,25]. However, as is well known, when compared with the generative model, one of the main restrictions of the transductive model is that it can hardly be used to describe the relations existing in the data. In particular, in terms of our problem, even though Model (2) may work well, it is laborious to express the dynamics of the data over time.

4. The Proposed Model

In this section, we elaborate on our multi-matrices factorization (MMF) approach. Given the sensor data matrix, R , in which the entry, $R_{i,j}$ ($1 \leq i \leq n$ and $1 \leq j \leq t$), is collected from S_i at T_j , our goal is to find the factor matrices, $U_{(1)}, U_{(2)}, \dots, U_{(t)} \in \mathbb{R}^{d \times n}$ and $V \in \mathbb{R}^{d \times t}$, such that: for $j = 1, 2, \dots, t$,

$$R_j \approx U_{(j)}^T V_j \quad (3)$$

where $U_{(j)}$ is regarded to be composed of the spatial features of all areas at T_j and V is treated as consisting of the temporal features of all time points. We denote the i th column of U_j as $U_{(j),i}$, which corresponds to the spatial feature value of S_i at T_j , and denote the j th column of V as V_j , which corresponds to the temporal feature value of T_j .

Taking advantage of the knowledge of the probability graph model, we assume that the dependent structure of the data in $U_{(1)}, U_{(2)}, \dots, U_{(t)}$, V and R is as illustrated in Figure 1. More specifically, we have the following assumptions:

I. Columns of $U_{(j)}$ ($1 \leq j \leq t$) are linearly independent, *i.e.*,

$$Pr(U_{(j)}) = \prod_{i=1}^n Pr(U_{(j),i})$$

II. $U_{(1),i}$ ($1 \leq i \leq n$) follows the same Gaussian distribution with a mean of zero and a covariance matrix $\sigma_U^2 I$, *i.e.*,

$$Pr(U_{(1),i} | \sigma_U) = (2\pi\sigma_U^2)^{-\frac{d}{2}} \exp\left\{-\frac{\|U_{(1),i}\|_2^2}{2\sigma_U^2}\right\}$$

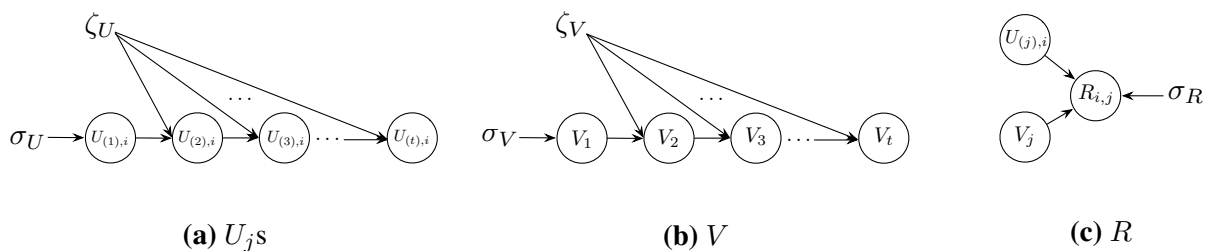
III. $U_{(j),i}$ ($1 \leq i \leq n$) are dependent in time order with the pre-specified priors, ζ_U and σ_U , *i.e.*,

$$Pr(U_{(j),i} | \zeta_U, \sigma_U) = Pr(U_{(1),i} | \zeta_U, \sigma_U) \times \prod_{j=2}^t Pr(U_{(j),i} | U_{(j-1),i}, \zeta_U, \sigma_U)$$

Moreover, for $j > 1$, we assume $U_{(j),i}$ is a Laplace random vector with location parameter $U_{(j-1),i}$ and scale parameter ζ_U , namely:

$$Pr(U_{(j),i} | U_{(j-1),i}, \zeta_U, \sigma_U) = \frac{1}{2\zeta_U} \exp\left\{-\frac{|U_{(j),i} - U_{(j-1),i}|}{\zeta_U}\right\}$$

Figure 1. The structure assumptions.



IV. The columns of V are linearly dependent in time order with the pre-specified priors, ζ_V and σ_V , i.e.,

$$Pr(V|\zeta_V, \sigma_V) = Pr(V_1|\zeta_V, \sigma_V) \times \prod_{j=2}^t Pr(V_j|V_{j-1}, \zeta_V, \sigma_V)$$

We also assume that, for $j > 1$:

$$Pr(V_j|V_{j-1}, \zeta_V, \sigma_V) = \frac{1}{2\zeta_V} \exp\left\{-\frac{|V_j - V_{j-1}|}{\zeta_V}\right\}$$

V. The (i, j) th entry of R ($1 \leq i \leq n$, $1 \leq j \leq t$) follows Gaussian distribution with a mean of $U_{(j),i}^T V_j$ and variance σ_R^2 , i.e.,

$$Pr(R_{i,j}|U_{(j),i}^T V_j, \sigma_R^2) = (2\pi\sigma_R^2)^{-\frac{1}{2}} \exp\left\{-\frac{(R_{i,j} - U_{(j),i}^T V_j)^2}{2\sigma_R^2}\right\}$$

Now, given R and the priors, σ_U , σ_V , σ_R , ζ_U , and ζ_V , let $U = \{U_{(1)}, U_{(2)}, \dots, U_{(t)}\}$; below, we find a solution to the following equation:

$$\{U^*, V^*\} = \underset{U, V}{\operatorname{argmax}} Pr(U, V|R, \sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V) \quad (4)$$

First, applying Bayes' theorem, we have:

$$Pr(U, V|R, \sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V) = \frac{Pr(U, V, R|\sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V)}{Pr(R|\sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V)}$$

Since R is observed and σ_U , σ_V , σ_R , ζ_U and ζ_V are pre-specified, the denominator, $Pr(R|\sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V)$, can be treated as a constant. Therefore:

Equation (4) $\Leftrightarrow$

$$\{U^*, V^*\} = \underset{U, V}{\operatorname{argmax}} Pr(U, V, R|\sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V) \quad (5)$$

Combing Assumptions (I.) $\sim$ (V.) and the dependency structure illustrated in Figure 1, we have:

$$\begin{aligned} & Pr(R, U, V|\sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V) \\ &= Pr(R|U, V, \sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V) \times Pr(U, V|\sigma_U, \sigma_V, \sigma_R, \zeta_U, \zeta_V) \\ &= Pr(R|U, V, \sigma_R) \times Pr(U|\sigma_U, \zeta_U) \times Pr(V|\sigma_V, \zeta_V) \\ &= \prod_{i=1}^n \prod_{j=1}^t Pr(R_{i,j}|U_{(j),i}, V_j, \sigma_R) \times \prod_{j=1}^t Pr(U_{(j)}|\sigma_U, \zeta_U) \times Pr(V|\sigma_V, \zeta_V) \\ &= \prod_{i=1}^n \prod_{j=1}^t Pr(R_{i,j}|U_{(j),i}, V_j, \sigma_R) \times \prod_{j=1}^t Pr(U_{(j),1}|\sigma_U) \prod_{i=1}^n \prod_{j=2}^t Pr(U_{(j),i}|U_{(j-1),i}, \zeta_U) \\ &\quad \times Pr(V_1|\sigma_V) \times \prod_{j=2}^t Pr(V_j|V_{j-1}, \zeta_V) \\ &\propto \exp\left(-\frac{1}{2\sigma_R^2} \sum_{i=1}^n \sum_{j=1}^t (U_{(j),i}^T V_j - R_{i,j})^2\right) \times \exp\left(-\frac{1}{2\sigma_U^2} \sum_{i=1}^n \|U_{(i),1}\|_2^2\right) \\ &\quad \times \prod_{i=1}^n \exp\left(-\sum_{j=2}^t \frac{|U_{(j),i} - U_{(j-1),i}|}{\zeta_U}\right) \times \exp\left(-\frac{1}{2\sigma_V^2} \|V_1\|_2^2\right) \times \exp\left(-\sum_{j=2}^t \frac{|V_j - V_{j-1}|}{\zeta_V}\right) \end{aligned}$$

Taking the logarithm on both sides and take the missing values into account, then we have:

Equation (5) $\Longleftrightarrow$

$$\begin{aligned} \{U^*, V^*\} = \arg \min_{U, V} & \left\{ \sum_{i=1}^n \sum_{j=1}^t (R_{i,j} - U_{(j),i}^T V_j)^2 \mathbb{I}(R_{i,j} \neq \perp') + \alpha \sum_{j=1}^t \|U_{(j),1}\|_2^2 \right. \\ & \left. + \gamma \sum_{i=1}^n \sum_{j=2}^t \|U_{(j),i} - U_{(j-1),i}\|_1 + \beta \|V_1\|_2^2 + \lambda \sum_{j=2}^t \|V_j - V_{j-1}\|_1 \right\} \end{aligned} \quad (6)$$

where $\alpha = \frac{\sigma_R^2}{\sigma_U^2}$, $\gamma = \frac{\sigma_R^2}{\zeta_U}$, $\beta = \frac{\sigma_R^2}{\sigma_V^2}$ and $\lambda = \frac{\sigma_R^2}{\zeta_V}$ are the regularization parameters.

As a supplement, we have the following comments on Model (6):

- I. *On the selection of the Gaussian prior:* In our model, since no prior information is available for the columns of the matrices, $U_{(1)}$ and V , hence, according to the max entropy principle [26], a reasonable choice for them is the Gaussian prior distribution.
- II. *On the ability to formalize the dynamics of the sensor data:* The ability to characterize the dynamics of the sensor data lies in the terms $\gamma \sum_{i=1}^n \sum_{j=2}^t \|U_{(j),i} - U_{(j-1),i}\|_1$ and $\lambda \sum_{j=2}^t \|V_j - V_{j-1}\|_1$. Obviously, for any two adjacent time points, T_{j-1} and T_j , if the interval is small enough (namely, $|T_j - T_{j-1}| \rightarrow 0$), then for any area, S_i , the values, $R_{i,j-1}$ and $R_{i,j}$, should be close to each other (namely, $|R_{i,j} - R_{i,j-1}| \rightarrow 0$). This can be enforced by tuning the parameters, γ and λ (see the following elaboration).

First of all, since for any $x \in \mathbb{R}^n$, $\|x\|_2 \leq \|x\|_1$, we have:

$$\begin{aligned} \|R_{i,j} - R_{i,j-1}\|_2 &= \|U_{(j),i}^T V_j - U_{(j-1),i}^T V_{j-1}\|_2 \\ &= \|(U_{(j),i} - U_{(j-1),i})^T V_j + U_{(j-1),i}^T (V_j - V_{j-1})\|_2 \\ &\leq \|U_{(j),i} - U_{(j-1),i}\|_2 \|V_j\|_2 + \|U_{(j-1),i}\|_2 \|V_j - V_{j-1}\|_2 \\ &\leq \|U_{(j),i} - U_{(j-1),i}\|_1 \|V_j\|_2 + \|U_{(j-1),i}\|_2 \|V_j - V_{j-1}\|_1 \end{aligned} \quad (7)$$

Secondly, it is obvious that greater regularization parameters (*i.e.*, α , β , γ and λ) result in smaller corresponding multipliers (*i.e.*, $\sum_{j=1}^t \|U_{(j),1}\|_2^2$, $\|V_1\|_2^2$, $\sum_{i=1}^n \sum_{j=2}^t \|U_{(j),i} - U_{(j-1),i}\|_1$ and $\sum_{j=2}^t \|V_j - V_{j-1}\|_1$). In particular:

$$\gamma \rightarrow \infty \Rightarrow \|U_{(j),i} - U_{(j-1),i}\|_1 \rightarrow 0 \quad (8)$$

and:

$$\lambda \rightarrow \infty \Rightarrow \|V_j - V_{j-1}\|_1 \rightarrow 0 \quad (9)$$

Hence, combining Equations (7)–(9), when $|T_j - T_{j-1}| \rightarrow 0$, we can simply take $\gamma \rightarrow \infty$ and $\lambda \rightarrow \infty$ and achieve $\|R_j - R_{j-1}\|_2 \rightarrow 0$.

On the other side, when $|T_j - T_{j-1}| \rightarrow \infty$, as is well known, the values in R_j and R_{j-1} are regarded as being independent. In this case, we can take $\gamma \rightarrow 0$ and $\lambda \rightarrow 0$, allowing R_j to be irrelevant to R_{j-1} .

III. *On the ℓ_1 norm:* It is straightforward to verify that, if we replace the ℓ_1 terms in Equation (6) with the ℓ_2 terms (equivalently, use the Gaussian distribution instead of the Laplace distribution in Assumptions (III.) and (IV.)), e.g., replacing $\|V_j - V_{j-1}\|_1$ with $\|V_j - V_{j-1}\|_2^2$, $\|R_j - R_{j-1}\|_2$ can still be bounded via tuning the regularization parameters, γ and λ . The reason for adopting the ℓ_1 norm here is two-fold: Firstly, as shown above, the ℓ_1 terms can lead to the bounded difference norm, $\|R_j - R_{j-1}\|_2$, and hence, the proposed model accommodates the ability to characterize the dynamics of the sensor data; secondly, according to the recent emerging works on compressed sensing [27,28], under some settings, the behavior of the ℓ_1 norm is similar to that of the ℓ_0 norm. In terms of our model, this result indicates that the ℓ_1 terms can restrict not only the magnitudes of the dynamics happening to the features, but also the number of features that changed in adjacent time points. In other words, with ℓ_1 norms, our model gains more expressibility.

5. The Algorithm

Below, we present the algorithm to solve Model (6). We denote:

$$W = \arg \min_{U,V} \sum_{i=1}^n \sum_{j=1}^t (R_{i,j} - U_{(j),i}^T V_j)^2 \mathbb{I}(R_{i,j} \neq \perp') + \alpha \sum_{j=1}^t \|U_{(j),1}\|_2^2 \\ + \beta \|V_1\|_2^2 + \gamma \sum_{i=1}^n \sum_{j=2}^t \|U_{(j),i} - U_{(j-1),i}\|_1 + \lambda \sum_{j=2}^t \|V_j - V_{j-1}\|_1$$

Apparently, W is convex with respect to $U_{(j),i}$ and V_j ($1 \leq i \leq n$ and $1 \leq j \leq t$). Therefore, we can obtain the local minimum solution via coordinate descent [29].

First, we introduce the signum function, sgn , for a real variable, x :

$$sgn(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \end{cases}$$

For $X = [x_1, x_2, \dots, x_n]' \in \mathbb{R}^n$, we denote $sgn(X) = [sgn(x_1), sgn(x_2), \dots, sgn(x_n)]'$.

Then, we calculate the partial subgradient of W with regard to $U_{(j),i}$ ($1 \leq i \leq n$ and $1 \leq j \leq t$) as follows:

For $j = 2, 3, \dots, t$, define $F_{j,i,1} = \gamma sgn(U_{(j),i} - U_{(j-1),i})$.

For $j = 1, 2, \dots, t-1$, define $F_{j,i,2} = -\gamma F_{j+1,i,1}$.

Let $F_{1,i,1} = F_{t,i,2} = 0$, and we have:

For $i = 1, 2, \dots, n$:

$$\frac{\partial W}{\partial U_{(1),i}} = 2\alpha U_{(1),i} - 2(R_{i,1} - U_{(1),i}^T V_1) V_1 \mathbb{I}(R_{i,1} \neq \perp') + F_{1,i,1} + F_{1,i,2}$$

For $i = 1, 2, \dots, n$ and $j = 2, 3, \dots, t$:

$$\frac{\partial W}{\partial U_{(j),i}} = -2(R_{i,j} - U_{(j),i}^T V_j) V_j \mathbb{I}(R_{i,j} \neq \perp') + F_{j,i,1} + F_{j,i,2}$$

Similarly, we calculate the partial subgradient of W with regard to V_j ($1 \leq j \leq t$):

For $2 \leq j \leq t$, define $G_{j,1} = \lambda \text{sgn}(V_j - V_{j-1})$.

For $1 \leq j \leq t-1$, denote $G_{j,2} = -G_{j+1,1}$.

Let $G_{1,1} = G_{t,2} = 0$, and we have:

$$\frac{\partial W}{\partial V_1} = 2\beta V_1 - 2 \sum_{i=1}^n (R_{i,1} - U_{(1),i}^T V_1) U_{(1),i} \mathbb{I}(R_{i,1} \neq \perp') + G_{1,1} + G_{1,2}$$

and for $j > 1$:

$$\frac{\partial W}{\partial V_j} = -2 \sum_{i=1}^n (R_{i,j} - U_{(j),i}^T V_j) U_{(j),i} \mathbb{I}(R_{i,j} \neq \perp') + G_{j,1} + G_{j,2}$$

Finally, with the results above, we present the solution algorithm in Algorithm 1.

6. Applications on Missing Sensor Data Imputation

In this section, we evaluate our approach through two large-sized datasets and compare the results with two state-of-the-art algorithms in terms of parametric sensitivity, convergence and missing data recovery performance. The following paragraphs describe the set-up, evaluation methodology and the results obtained. To simplify the parameter tuning, we set $\alpha = \beta$ and $\lambda = \gamma$ in the algorithm implementation.

6.1. Evaluation Methodology

Three state-of-the-art algorithms are selected for comparison to the proposed MMP model. The first one is the k -nearest neighbor-based imputation model [1]. As a *local model*, for every missing entry, $R_{i,j}$, the *knn* method takes the estimation, $\hat{R}_{i,j}$, as the mean of the k nearest neighbors to it. Let $\mathcal{N}(x)$ be the set consisting of the k non-empty entries to x ; then:

$$\hat{R}_{i,j} = \frac{1}{k} \sum_{R_{i,l} \in \mathcal{N}(R_{i,j})} R_{i,l} \quad (10)$$

The second algorithm is the probabilistic principle components analysis model (PPCA) [30,31], which has achieved state-of-the-art performance in the missing traffic flow data imputation problem [31]. Denote the observations of the incomplete matrix, R , as R_o . Let $x \sim N(0, I)$; to estimate the missing values, PPCA first fits the parameters μ and C with:

$$\{\mu^*, C^*\} = \arg \max_{\mu, C} Pr(x|R_o) \sim N(x\sigma^{-2}(I + \sigma^{-2}CC^T)^{-1}CR_o, (I + \sigma^{-2}CC^T)^{-1}) \quad (11)$$

where σ is the tunable parameter. Then, with R_o , μ^* and C^* , it takes the estimation of the missing values (denoted as R_m) as:

$$\hat{R}_m = \arg \max_{R_m} N(R_m|C^T x, \sigma^2 I) \quad (12)$$

The third algorithm is the probabilistic matrix factorization model (PMF) [7], one of the most popular algorithms targeting the Netflix matrix completion problem. PMF first seeks the low rank matrices, U and V , that:

$$\{U^*, V^*\} = \arg \min_{U, V} \sum_{i=1}^n \sum_{j=1}^t (R_{i,j} - U_i^T V_j)^2 \mathbb{I}(R_{i,j} \neq \perp') + \alpha \sum_{i=1}^n \|U_i\|_2^2 + \beta \sum_{j=1}^t \|V_j\|_2^2 \quad (13)$$

then for the missing entry $R_{i,j}$, it takes the estimation as

$$\hat{R}_{i,j} = U_i^T V_j \quad (14)$$

Algorithm 1: The Multi-Matrices Factorization Algorithm.

Input: matrix R ; number of the latent features, d ; learning rates, η_1, η_2, η_3 and η_4 ; regularization parameters, α and λ ; threshold ϵ .

Output: the estimated matrix, $\hat{R}$.

// Initialize U and V .

```

1 Draw random vectors,  $U_{(1),1}, U_{(1),2}, \dots, U_{(1),n}, V_1 \sim N(\mathbf{0}, I)$ ;
2 for  $j = 2; j \leq t; j++$  do
3   | Let  $V_j = V_{j-1} + Z$ ; here  $Z \sim \text{Laplace}(\mathbf{0}, 1)$ ;
4 end
5 for  $j = 2; j \leq t; j++$  do
6   | for  $i = 1; i \leq n; i++$  do
7     | Let  $U_{(j),i} = U_{(j-1),i} + Z$ ; here  $Z \sim \text{Laplace}(\mathbf{0}, 1)$ ;
8   | end
9 end
  // Coordinate descent.
10  $W_1 = \sum_{i=1}^n \sum_{j=1}^t (R_{i,j} - U_{(j),i}^T V_j)^2 \mathbb{I}(R_{i,j} \neq \perp) + \alpha \sum_{j=1}^t \|U_{(j),1}\|_2^2$ 
11    $+ \beta \|V_1\|_2^2 + \gamma \sum_{i=1}^n \sum_{j=2}^t \|U_{(j),i} - U_{(j-1),i}\|_1 + \lambda \sum_{j=2}^t \|V_j - V_{j-1}\|_1$ ;
12  $W_2 = \inf$ ;
13 while  $|W_2 - W_1| > \epsilon$  do
14   |  $W_2 = W_1$ ;
15   | for  $i = 1, 2, \dots, n$  do
16     | Let  $U_{(1),i}^{new} = U_{(1),i} - \eta_1 \frac{\partial W_1}{\partial U_{(1),i}}$ ;
17   | end
18   | for  $j > 1$  and  $i = 1, 2, \dots, n$  do
19     | Let  $U_{(j),i}^{new} = U_{(j),i} - \eta_2 \frac{\partial W_1}{\partial U_{(j),i}}$ ;
20   | end
21   |  $V_1^{new} = V_1 - \eta_3 \frac{\partial W_1}{\partial V_1}$ ;
22   | for  $j = 2, \dots, t$  do
23     | Let  $V_j^{new} = V_j - \eta_4 \frac{\partial W_1}{\partial V_j}$ ;
24   | end
25   | Replace all  $U_{(j),i}$ s with  $U_{(j),i}^{new}$ s, and  $V_j$ s with  $V_j^{new}$ s, recompute  $W_1$ ;
26 end
27 return  $\hat{R}$ , where  $\hat{R}_j = U_{(j)}^T V_j$ ;

```

These three algorithms, as well as the proposed MMF, are employed to perform missing imputations for the incomplete matrix, R , on the same datasets.

The testing protocol adopted here is the *Given X* ($0 < X < 1$) protocol [32], i.e., given a matrix, R , only X percent of its observed entries are revealed, while the remaining observations are concealed to

evaluate the trained model. For example, a setting with $X = 10\%$ means that the algorithm is trained with 10% of the non-missing entries, and the rest of the 90% non-missing ones are held and are to be recovered. In both of the experiments on synthetic and real datasets, the data partition is repeated five times, and the average results, as well as the standard deviations over the five repetitions are recorded.

Similar to many other missing imputation problems [1,3–5,7,13,33–35], we employ the root mean square error (RMSE) to depict the distance between the real values and the estimations: Let $S = \{s_1, s_2, \dots, s_n\}$ be the test dataset and $\hat{S} = \{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_n\}$ be the estimated set; here, $\hat{s}_i$ is the estimation of s_i . Then, the RMSE of the estimation is given by $\sqrt{\frac{1}{n} \sum_{k=1}^n (s_k - \hat{s}_k)^2}$.

6.2. Synthetic Validation

To conduct a synthetic validation of the studied approaches, we randomly draw a $100 \times 10,000$ matrix R using the procedure detailed in Algorithm 2. The rows in R correspond to the areas, $S_1, S_2, \dots, S_n$, and the columns correspond to the time. Thus, $R_{i,j}$ represents the data collected in S_i at time T_j . Notably, the parameter, $r_{i,j}$, in Algorithm 2 is used to control the magnitude of the variation happening to S_i from T_{j-1} to T_j . Combining lines 4 and 5, we have: for $i \in [1, n]$ and $j \in [1, t]$, $|\frac{R_{i,j} - R_{i,j-1}}{R_{i,j-1}}| = |r_{i,j}| \leq 0.1$. This constraint ensures that the data collected in S_i does not change too much over time T_{j-1} to T_j .

Algorithm 2: Synthetic Data Generating Procedure.

```

1 for  $i = 1, 2, \dots, n$  do
2   Draw  $R_{i,1} \sim N(0, 1)$ ;
3   for  $j = 2, 3, \dots, t$  do
4     Let  $r_{i,j} \sim \text{Uniform}(-0.1, 0.1)$ ;
5      $R_{i,j} \leftarrow R_{i,j-1} + r_{i,j} R_{i,j-1}$ ;
6   end
7 end

```

We first evaluate the sensitivity of the proposed algorithm to the regularization parameters, α and λ . Half of the entries in R are randomly selected as testing data and recovered using the remaining 50% as the training data. Namely, we take $X = 50\%$ in the *Given X* protocol. In the experiments, we first fix $\alpha = 0.01$, tune λ via $\lambda = 0.01 \times 2^n$ ($n = 0, 1, \dots, 7$) and, then, do the reverse by changing α via $\alpha = 0.01 \times 2^n$ ($n = 0, 1, \dots, 7$), but setting $\lambda = 0.01$. The average RMSEs with the same parameter settings on different data partitions are summarized in Figure 2.

In Figure 2, the RMSE-1 curve represents the recovery errors obtained by fixing α and changing the value of λ . The RMSE-2 curve corresponds to the errors with different α values and fixed λ . We can see that even when λ is expanded by more than 100 times ($2^7 = 128$), the RMSE still remains stable. A similar result also appears in the experiments on the parameter, α , where a significant change of the RMSE only occurs when n is greater than six, *i.e.*, when α is expanded more than 60 times.

The second experiment we conduct is to study the prediction ability of the proposed algorithm, as well as that of the comparison algorithms. In the *Given X* protocol, we set $X = 10\%, 20\%, \dots, 90\%$ in sequence. Then, for each X value, we perform missing imputations via our algorithm and

the comparison algorithms. In the all implementations, we set $k = 5$ for the knn model. As with the MF-based algorithms, we examine their performance with respect to the latent feature dimension $d = 10$ and $d = 30$ respectively. Furthermore, in the implementation of MMF, we fix $\alpha = \lambda = 0.1$, $\eta_1 = \eta_2 = \eta_3 = \eta_4 = 0.01$. All results are summarized in Table 1.

Figure 2. Empirical studies on parameter sensitivity.

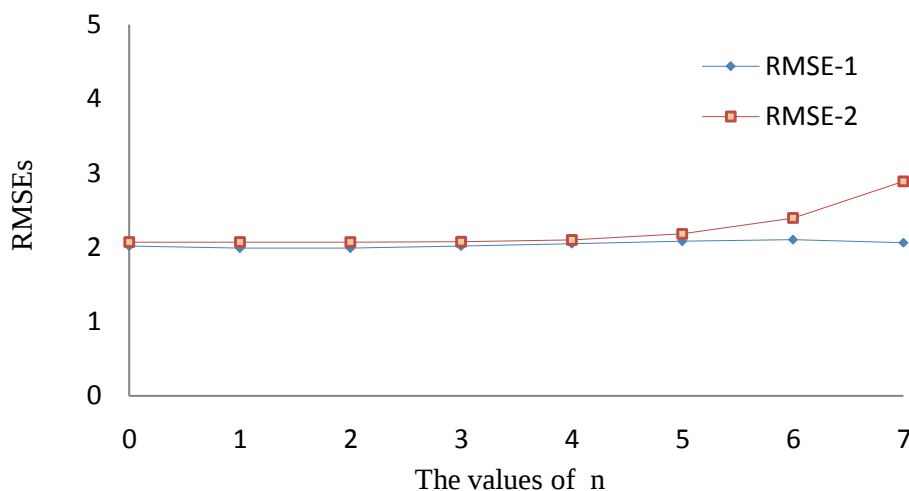


Table 1. Recovery errors on the synthetic dataset (mean $\pm$ std).

		10%	20%	30%	40%	50%	60%	70%	80%	90%
d = 10	knn	13.41 $\pm$ 0.62	6.80 $\pm$ 0.19	4.44 $\pm$ 0.06	3.12 $\pm$ 0.05	2.27 $\pm$ 0.02	1.81 $\pm$ 0.09	1.72 $\pm$ 0.01	1.69 $\pm$ 0.05	1.62 $\pm$ 0.06
	PPCA	17.09 $\pm$ 2.08	20.24 $\pm$ 1.30	22.75 $\pm$ 1.32	23.96 $\pm$ 2.19	20.79 $\pm$ 1.00	16.67 $\pm$ 1.34	18.68 $\pm$ 0.81	11.98 $\pm$ 0.70	5.11 $\pm$ 3.10
	PMF	3.23 $\pm$ 0.23	3.33 $\pm$ 0.19	3.29 $\pm$ 0.12	3.34 $\pm$ 0.07	3.29 $\pm$ 0.09	1.69 $\pm$ 0.04	1.83 $\pm$ 0.02	1.81 $\pm$ 0.03	1.85 $\pm$ 0.04
	MMF	3.07 $\pm$ 0.07	2.21 $\pm$ 0.10	2.14 $\pm$ 0.09	1.98 $\pm$ 0.06	1.93 $\pm$ 0.06	1.92 $\pm$ 0.04	1.75 $\pm$ 0.03	1.84 $\pm$ 0.02	1.80 $\pm$ 0.03
d = 30	PPCA	19.65 $\pm$ 3.01	22.48 $\pm$ 0.86	24.86 $\pm$ 1.54	23.99 $\pm$ 0.61	22.67 $\pm$ 0.49	20.22 $\pm$ 1.46	17.53 $\pm$ 0.70	14.23 $\pm$ 1.70	11.14 $\pm$ 1.72
	PMF	3.20 $\pm$ 0.21	3.32 $\pm$ 0.13	3.35 $\pm$ 0.09	3.36 $\pm$ 0.11	3.31 $\pm$ 0.07	1.78 $\pm$ 0.04	1.81 $\pm$ 0.02	1.79 $\pm$ 0.02	1.86 $\pm$ 0.11
	MMF	3.06 $\pm$ 0.05	2.17 $\pm$ 0.08	2.07 $\pm$ 0.05	1.94 $\pm$ 0.03	1.74 $\pm$ 0.03	1.62 $\pm$ 0.02	1.64 $\pm$ 0.03	1.65 $\pm$ 0.01	1.69 $\pm$ 0.03

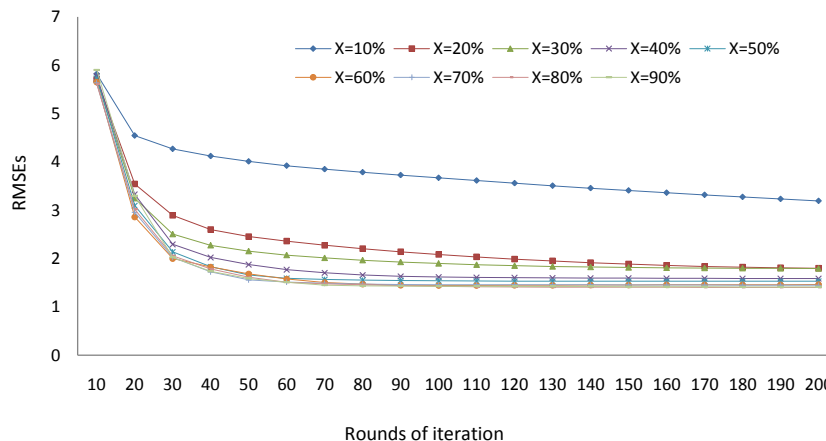
As shown in Table 1, when X is large, e.g., $X \geq 50\%$, knn is competitive with the matrix factorization methods, while in the other situations, the MF methods outperform it significantly. In terms of the MF-based methods, we find that our algorithm outperforms PPCA significantly in all settings. The RMSEs of our algorithm are at most roughly 20% of that of PPCA. Specifically, for $X \in [30\%, 80\%]$, the RMSEs of the proposed algorithm are even only 10% of that of PPCA. We can also observe that the parameter, d , has a different impact on the performances of our algorithm and PPCA: When d changes from 10 to 30, most of the RMSEs of PPCA increase evidently, while for our algorithm, the RMSEs are reduced by roughly 5%.

When compared with PMF, our algorithm also performs better in most of the settings: PMF achieves lower RMSEs than MMF only in two cases, in which $d = 10, X = 60\%$ and $d = 10, X = 80\%$ respectively. Another interesting finding is that the promotion of the feature number, d (from 10 to 30), has little impact on the performance of PMF.

We also exam the convergence speed of the proposed algorithm. In the missing recovery experiments conducted above, for each X setting, we record the average RMSE of the recovered results after every 10 iterations of all data partitions. We can see from Figure 3 that, for all X values, the errors drop

dramatically in the first 20 iterations and remain stable after the first 100 iterations. We can conclude that the proposed algorithm converges to the local optimization solutions after around 100 iterations.

Figure 3. Empirical studies on convergence speed.



6.3. Application to Impute the Missing Traffic Speed Values

To evaluate the feasibility of the proposed approach on real-world applications, in this section, we conduct another experiment on a traffic speed dataset, which was collected in the urban road network of Zhuhai City [36], China, from April 1, 2011 to April 30, 2011. The data matrix, R , consists of 1,853 rows and 8,729 columns. Each row corresponds to a road, and each column corresponds to a five minute-length time interval. All columns are arranged in ascending order of time. An entry, $R_{i,j}$ ($1 \leq i \leq 1,853, 1 \leq j \leq 8,729$), in R is the aggregate mean traffic speed of the i th road in the j th interval. Since all the data in R are collected by floating cars [37], the value of $R_{i,j}$ could be missing if there is no car on the i th road in the j th time interval. Our statistics show that in R , there are nearly half of the entries, *i.e.*, eight million entries are missing values.

We perform missing imputation on matrix R using the studied algorithms with parameter settings $k = 5$ and $d = 10$. In the implementation of MMF, we fix $\alpha = 0.25$, $\lambda = 0.5$, $\eta_1 = \eta_2 = \eta_3 = \eta_4 = 0.5$. We summarize all results in Table 2, from which both the feasibility and effectiveness of MMF are well verified. In detail, when X is large enough, *e.g.*, $X \geq 80\%$, knn is competitive, while in the other cases, knn cannot work as well as the MF based algorithms. As for the MF algorithms, we see that the proposed MMF outperforms PPCA and PMF in all X settings. Particularly, when the observations are few ($X = 10\%$ and $X = 20\%$), the errors of our algorithm reduce by 33% compared to those of PPCA and by 10% compared to those of PMF, respectively. When $X > 20\%$, the RMSE differences between PPCA and our algorithm tend to be slight, but the overall errors of PPCA are roughly 3% ~ 5% higher than those of MMF. For PMF, the RMSEs remain about 10% higher than MMF in all settings.

Table 2. Recovery errors on the transportation dataset (mean $\pm$ std).

	10%	20%	30%	40%	50%	60%	70%	80%	90%
knn	40.47 $\pm$ 0.02	31.79 $\pm$ 0.01	25.41 $\pm$ 0.02	21.35 $\pm$ 0.00	18.33 $\pm$ 0.00	15.89 $\pm$ 0.01	13.73 $\pm$ 0.01	11.67 $\pm$ 0.00	9.45 $\pm$ 0.02
PPCA	17.90 $\pm$ 0.01	17.36 $\pm$ 0.01	13.00 $\pm$ 0.02	12.25 $\pm$ 0.01	11.47 $\pm$ 0.01	11.31 $\pm$ 0.03	11.19 $\pm$ 0.02	11.14 $\pm$ 0.04	11.16 $\pm$ 0.10
PMF	14.41 $\pm$ 0.01	12.83 $\pm$ 0.03	12.43 $\pm$ 0.01	12.33 $\pm$ 0.02	12.36 $\pm$ 0.01	12.35 $\pm$ 0.01	12.13 $\pm$ 0.00	11.99 $\pm$ 0.03	11.96 $\pm$ 0.02
MMF	11.79 $\pm$ 0.02	11.51 $\pm$ 0.01	11.43 $\pm$ 0.01	11.05 $\pm$ 0.02	11.05 $\pm$ 0.01	11.01 $\pm$ 0.00	10.83 $\pm$ 0.01	10.69 $\pm$ 0.01	10.70 $\pm$ 0.02

7. Conclusion

Missing estimation is one of the main concerns in current studies on sensor data-based applications. In this work, we formulate the estimation problem as a matrix completion one and present a multi-matrices factorization model to address it. In our model, each column, R_j , of the target matrix, R , is approximated by the product of a spatial feature matrix, $U_{(j)}$, and a temporal feature vector, V_j . Both U_j and V_j are time dependent, and hence, their product accommodates the ability to describe the time variant sensor data. We also present a solution algorithm to the factorization model. Empirical studies on a synthetic dataset and real sensor data show that our approach outperforms the comparison algorithms.

Reviewing the present work, it is notable that the proposed model only incorporates the temporal structure information, while the information on the spatial structure is disregarded, e.g., the data collected in two adjacent areas, S_k and S_l , should be close to each other. Hence, our next step is to extend our model with more complex structured data.

Acknowledgments

This work has been partially supported by National High-tech R&D Program (863 Program) of China under Grant 2012AA12A203.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. García-Laencina, P.J.; Sancho-Gómez, J.L.; Figueiras-Vidal, A.R.; Verleysen, M. K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputation* **2009**, *72*, 1483–1493.
2. Ni, D.; Leonard, J.D.; Guin, A.; Feng, C. Multiple imputation scheme for overcoming the missing values and variability issues in ITS data. *J. Transport. Eng.* **2005**, *131*, 931–938.
3. Smith, B.L.; Scherer, W.T.; Conklin, J.H. Exploring imputation techniques for missing data in transportation management systems. *Transport. Res. Record. J. Transport. Res. Board* **2003**, *1836*, 132–142.
4. Qu, L.; Zhang, Y.; Hu, J.; Jia, L.; Li, L. A BPCA Based Missing Value Imputing Method for Traffic Flow Volume Data. In Proceedings of 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 985–990.
5. Jiang, N.; Gruenwald, L. Estimating Missing Data in Data Streams. In *Advances in Databases: Concepts, Systems and Applications*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 981–987.
6. Netflix Prize. Available online: <http://www.netflixprize.com> (accessed on 1 July 2013).
7. Salakhutdinov, R.; Mnih, A. Probabilistic matrix factorization. *Adv. Neural Inf. Process. Syst.* **2008**, *20*, 1257–1264.

8. Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 2008; pp. 426–434.
9. Seung, D.; Lee, L. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2001**, *13*, 556–562.
10. Srebro, N. Learning with Matrix Factorizations. Ph.D Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA, 2004.
11. Srebro, N.; Rennie, J.D.; Jaakkola, T. Maximum-margin matrix factorization. *Adv. Neural Inf. Process. Syst.* **2005**, *17*, 1329–1336.
12. Independent and identically distributed random variables. Available online: http://en.wikipedia.org/wiki/independent_and_identically_distributed_random_variables (accessed on 1 July 2013).
13. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37.
14. Xu, W.; Liu, X.; Gong, Y. Document Clustering Based on Non-Negative Matrix Factorization. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Pisa, Italy, 28 July–1 August 2003; pp. 267–273.
15. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791.
16. Brunet, J.P.; Tamayo, P.; Golub, T.R.; Mesirov, J.P. Metagenes and molecular pattern discovery using matrix factorization. *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 4164–4169.
17. Candès, E.J.; Recht, B. Exact matrix completion via convex optimization. *Found. Comput. Math.* **2009**, *9*, 717–772.
18. Hoyer, P.O. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.* **2004**, *5*, 1457–1469.
19. Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.* **2010**, *11*, 19–60.
20. Ke, Q.; Kanade, T. Robust L_1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), San Diego, CA, USA, 20–26 June 2005; pp. 739–746.
21. Nati, N.S.; Jaakkola, T. Weighted Low-Rank Approximations. In proceedings of the 20th International Conference on Machine Learning (ICML 2003), Washington, DC, USA, 21–24 August 2003; pp. 720–727.
22. Abernethy, J.; Bach, F.; Evgeniou, T.; Vert, J.P. *Low-Rank Matrix Factorization with Attributes*; Technical Report; N-24/06/MM; Paris, France; September 2006.
23. Vapnik, V. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998.
24. Rissanen, J. *Minimum Description Length Principle*; Springer: Berlin, Germany, 2010.
25. Candès, E.J.; Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inf. Theory* **2010**, *56*, 2053–2080.
26. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 2012.

27. Candès, E.J.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509.
28. Donoho, D.L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306.
29. Bertsekas, D.P. *Nonlinear Programming*; Athena Scientific: Belmont, MA, USA, 1999.
30. Tipping, M.E.; Bishop, C.M. Probabilistic principal component analysis. *J. Royal Stat. Soc. Ser. B Stat. Methodol.* **1999**, *61*, 611–622.
31. Qu, L.; Hu, J.; Li, L.; Zhang, Y. PPCA-based missing data imputation for traffic flow volume: A systematical approach. *IEEE Trans. Intell. Transport. Syst.* **2009**, *10*, 512–522.
32. Marlin, B. Collaborative Filtering: A Machine Learning Perspective. Ph.D Thesis, University of Toronto, ON, Canada, 2004.
33. Nguyen, L.N.; Scherer, W.T. *Imputation Techniques to Account for Missing Data in Support of Intelligent Transportation Systems Applications*; No. UVACTS-13-0-78; University of Virginia Charlottesville: Charlottesville, VA, USA, 2003.
34. Gold, D.L.; Turner, S.M.; Gajewski, B.J.; Spiegelman, C. Imputing Missing Values in Its Data Archives for Intervals under 5 Minutes. In Proceedings of 80th Annual Meeting of Transportation Research Board, Washington, DC, USA, 7–11 January 2001.
35. Shuai, M.; Xie, K.; Pu, W.; Song, G.; Ma, X. An Online Approach Based on Locally Weighted Learning for Real Time Traffic Flow Prediction. The 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2008), Irvine, CA, USA, 5–7 November 2008.
36. Zhuhai. Available online: <http://en.wikipedia.org/wiki/Zhuhai> (accessed on 1 July 2013).
37. Floating Car Data. Available online: http://en.wikipedia.org/wiki/Floating_car_data (accessed on 1 July 2013).