IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON BIG DATA ANALYTICS FOR SMART AND CONNECTED HEALTH

Received February 22, 2017, accepted March 23, 2017, date of publication March 29, 2017, date of current version May 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2688449

Supervised Collaborative Filtering Based on Ridge Alternating Least Squares and Iterative Projection Pursuit

BO-WEI CHEN[1] (Senior Member, IEEE), WEN JI[2] (Senior Member, IEEE), SEUNGMIN RHO[3] (Member, IEEE), AND YU GU[4]

[1]School of Information Technology, Monash University, Melbourne, VIC 3800, Australia
[2]Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[3]Department of Media Software, Sungkyul University, Anyang 430742, South Korea
[4]School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China

Corresponding authors: Wen Ji (jiwen@ict.ac.cn) and Yu Gu (guyu@ustb.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572466 and in part by the Beijing Natural Science Foundation under Grant 4162059.

ABSTRACT This paper presents a supervised data imputation based on the class-dependent matrix factors, which are generated during matrix factorization. The proposed ridge alternating least squares imputation uses class information to create substituted values, which approximate the characteristics of their corresponding classes, for missing entries. In the training phase, the incomplete data with label information are divided into different classes based on their labels, such that basis matrices become class-dependent. Subsequently, iterative projection pursuit is proposed to perform imputation for testing data by computing the linear combination of these class-dependent basis matrices and their corresponding reconstruction weights. The class-dependent basis matrix with the minimum loss during reconstruction is regarded as the correct imputation for a testing sample, of which the substituted values are derived from the matrix factors of its class. Experiments on open data sets showed that the proposed method successfully decreased the imputation error by 40.52% on average, better than typical unsupervised collaborative filtering, while maintaining classification accuracy.

INDEX TERMS Incomplete data analysis, privacy preservation, supervised collaborative filtering, collaborative filtering (CF), alternating least squares (ALS), supervised data imputation, data imputation, singular value decomposition (SVD), supervised nonnegative matrix factorization (NMF), recommendation system, low-rank matrix approximation, matrix completion, matrix factorization, iterative projection pursuit.

I. INTRODUCTION

Incomplete data analysis is an important topic in data analytics. When samples contain missing values, they become non-vectorial data. Subsequently, typical Mathematical operations are inapplicable under such circumstances. How to handle such a problem is interesting in data analytics. Although this topic arose in the earlier 1970s [1], it has received much attention recently due to the need for privacy preservation in cloud computing [2]. Data are deliberately removed or masked to protect personal private information and to avoid being maliciously manipulated in cloud data centers [3], [4]. At present, several approaches have been devoted to incomplete data analysis, and they can be roughly classified into two categories. One is single imputation [5], and the other is multiple imputation (e.g., Markov Chain Monte Carlo (MCMC) [6]). Regarding single imputation, it contains various techniques, for example, deletion, fixed-value replacement (e.g., means or medians), hot decks (i.e., inserted values are selected based on the same dataset), cold decks (i.e., insertion is derived from another dataset), stochastic regression (e.g., interpolation), and matrix completion (e.g., imputation based on low-rank matrix approximation [7]). Recently, matrix completion has gained considerable attention due to its success in collaborative filtering.

Collaborative filtering (CF) is one of the subtopics in incomplete data analysis. It has been widely used in recommendation systems [8], [9] because it is difficult to ask every user to give ratings to all the products in the database. Collaborative filtering relies on matrix factorization approaches, e.g., nonnegative matrix factorization (NMF),

singular value decomposition (SVD), and alternating least squares (ALS). These methods are frequently adopted in data imputation areas. With constraints, such as nonnegative elements and the number of factors, NMF, SVD, and ALS have equivalent results.

In earlier research [10], [11], matrix factorization focused on finding bases, where an observed M-by-N matrix $\mathbf{X}$ was decomposed into two low-rank matrix factors, i.e., $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$, as shown in (1). The dimensions of the two matrix factors are M-by-D and D-by-N, respectively. The former signifies the basis matrix, whereas the latter is regarded as the coefficient matrix.

$$\mathbf{X} \approx \mathbf{U}^{\mathrm{T}}\mathbf{V}. \tag{1}$$

The techniques for finding/updating the basis and coefficient matrices ranged from plain ALS regression [12] and multiplicative update methods [13], [14] to stochastic gradient descent ALS [15]. These three methods were the corner stones of collaborative filtering. The common advantage of these methods is simplicity and easy implementation. One may refer to [16] for brief introduction on plain ALS regression and stochastic gradient descent ALS.

To prevent the ALS model from overfitting, the research [8] and [9] examined regularization while performing matrix factorization. Paterek [8] investigated the ALS model by appending additional biases to the cost function while finding matrix factors. These additional biases were jointly discovered along with matrix factors during factorization. Zhou *et al.* [9] proposed the weighted ALS model, where two ridge parameters were respectively imposed upon the matrix factors during factorization. Such an idea was similar to that of ridge regression. Ridge terms stabilize the system when the inverse of matrices is computed. To further standardize and to ensure uniqueness of the discovered bases, orthogonal matrix factorization was developed by Ding *et al.* [17]. However, it is difficult to generate unique and orthogonal bases due to the incomplete information among observed matrices. This deepens the difficulty of matrix factorization in incomplete data analysis, especially when new class-dependent samples arrive.

The aforementioned typical factorization methodologies did not embed label information in the model. Such a type of collaborative filtering is unsupervised. Unlike unsupervised methods, supervised learning takes advantages of label information and encodes it into the process if labels are observed. Such information helps systems discriminate inputs after training. Data imputation is subsequently based on the statistical distribution of the corresponding classes.

To embed label information into the analysis, this work proposes a supervised ALS imputation for generating class-dependent substituted values. The contributions of this work are summarized as follows.

- Unlike semisupervised CF, which mixes training data and testing data to perform data imputation/classification, the proposed method separates the testing stages from the training stage. This avoids redundant computation.
- When new samples with missing entries arrive, the proposed method can use class-dependent basis matrices to perform vector approximation by using the developed iterative projection pursuit. The imputed data are based on class-dependent bases obtained in the training phase.
- Class-dependent data are imputed by class-dependent basis matrices. This generates data distributions that reflect statistical characteristics.

The rest of this paper is organized as follows. Section II introduces typical collaborative filtering based on ALS. Subsequently, the proposed supervised ALS imputation is detailed in Section 3. Section 4 shows the experimental results, and conclusions are drawn in Section 5.

II. COLLABORATIVE FILTERING BASED ON ALS

This section begins with plain ALS [12], and then ridge ALS [9] is subsequently introduced. Firstly, consider the case of complete data $\mathbf{X}$. Given an M-by-N matrix $\mathbf{X}$ without missing values, where M specifies the number of dimensions, and N denotes the number of observed samples, ALS matrix completion is expressed as

$$E_{\mathrm{ALS}}(\mathbf{U}, \mathbf{V}) = \left\| \mathbf{X} - \mathbf{U}^{\mathrm{T}}\mathbf{V} \right\|_{\mathcal{F}}^{2} \tag{2}$$

where $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ are respectively M-by-D and D-by-N unknown matrices, D is the intermediate dimension, $\|\cdot\|_{\mathcal{F}}$ represents the Frobenius norm, and T is the transpose operator. As mentioned earlier in (1), $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ are low-rank matrices. This implies $D < M$ and $D < N$. The objective of ALS matrix completion is to find $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$, such that the error E_{ALS} becomes minimized. Namely, $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ are used to approximate $\mathbf{X}$. This is equivalent to decomposition of $\mathbf{X}$ into matrix factors, $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$.

Differentiating E_{ALS} with respect to $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ yields

$$\frac{\partial E_{\mathrm{ALS}}}{\partial \mathbf{V}} = -\mathbf{U}\left(\mathbf{X} - \mathbf{U}^{\mathrm{T}}\mathbf{V}\right) \tag{3}$$

and

$$\frac{\partial E_{\mathrm{ALS}}}{\partial \mathbf{U}^{\mathrm{T}}} = -\mathbf{V}\left(\mathbf{X} - \mathbf{U}^{\mathrm{T}}\mathbf{V}\right)^{\mathrm{T}}. \tag{4}$$

Zeroing the equations gives

$$\mathbf{V} = \left(\mathbf{U}\mathbf{U}^{\mathrm{T}}\right)^{-1}\mathbf{U}\mathbf{X} \tag{5}$$

and

$$\mathbf{U} = \left(\mathbf{V}\mathbf{V}^{\mathrm{T}}\right)^{-1}\mathbf{V}\mathbf{X}^{\mathrm{T}}. \tag{6}$$

Notably, as $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ are unknown matrices, heuristic initialization of $\mathbf{U}$ is necessary to compute $\mathbf{V}$. Recursive update of $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ based on (5) and (6) leads to a converged solution. Moreover, D is an unknown variable, which needs to be predefined before recursion.

For ridge ALS, or equivalently "ALS with Weighted-Regularization (ALS-WR)" [9], it uses a ridge parameter

to regularize and prevent $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ from overfitting. The purpose of ridge parameters is like that in Kernel Ridge Regression (KRR) [18]. Let ρ_{U} and ρ_{V} represent the ridge parameters for $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$, respectively. Thus, (2) becomes

$$E_{\mathrm{rALS}}(\mathbf{U},\mathbf{V}) = \left\|\mathbf{X}-\mathbf{U}^{\mathrm{T}}\mathbf{V}\right\|_{\mathcal{F}}^{2} + \rho_{\mathrm{U}}\|\mathbf{U}\|_{\mathcal{F}}^{2} + \rho_{\mathrm{V}}\|\mathbf{V}\|_{\mathcal{F}}^{2}. \tag{7}$$

Likewise, differentiating E_{rALS} with respect to $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ yields

$$\frac{\partial E_{\mathrm{rALS}}}{\partial \mathbf{V}} = -\mathbf{U}\left(\mathbf{X}-\mathbf{U}^{\mathrm{T}}\mathbf{V}\right) + \rho_{\mathrm{U}}\mathbf{V} \tag{8}$$

and

$$\frac{\partial E_{\mathrm{rALS}}}{\partial \mathbf{U}^{\mathrm{T}}} = -\mathbf{V}\left(\mathbf{X}-\mathbf{U}^{\mathrm{T}}\mathbf{V}\right)^{\mathrm{T}} + \rho_{\mathrm{U}}\mathbf{U}. \tag{9}$$

Zeroing the equations gives

$$\mathbf{V} = \left(\mathbf{U}\mathbf{U}^{\mathrm{T}} + \rho_{\mathrm{U}}\mathbf{I}\right)^{-1}\mathbf{U}\mathbf{X} \tag{10}$$

and

$$\mathbf{U} = \left(\mathbf{V}\mathbf{V}^{\mathrm{T}} + \rho_{\mathrm{V}}\mathbf{I}\right)^{-1}\mathbf{V}\mathbf{X}^{\mathrm{T}} \tag{11}$$

where $\mathbf{I}$ is an identity matrix.

The effect of ridge parameters stabilizes the inverse and avoids singular matrices. As mentioned earlier, the system can iteratively update $\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ based on (10) and (11) to generate a solution. Other variations of ridge ALS are also applicable to computation of solutions, for example, further regularized ridge ALS, called regularized SVD in [8], and ridge ALS with stochastic gradient descent [10].

For matrix $\mathbf{X}$ with missing values, the procedure is still the same, as shown in (12) and (13). The difference is that an element-wise mask G is imposed on $\mathbf{X}$. If an element of $\mathbf{X}$ is missing, then such an entry is temporally substituted with a zero first.

$$\mathbf{V} = \left(\mathbf{U}\mathbf{U}^{\mathrm{T}} + \rho_{\mathrm{U}}\mathbf{I}\right)^{-1}\mathbf{U} \times \mathrm{G}(\mathbf{X}) \tag{12}$$

and

$$\mathbf{U} = \left(\mathbf{V}\mathbf{V}^{\mathrm{T}} + \rho_{\mathrm{V}}\mathbf{I}\right)^{-1}\mathbf{V} \times \mathrm{G}(\mathbf{X})^{\mathrm{T}}. \tag{13}$$

Furthermore, when the system computes the error of imputation, the missing entries are ignored.

$$E_{\mathrm{rALS}}(\mathbf{U},\mathbf{V}) = \left\|\mathrm{G}\left(\mathbf{X}-\mathbf{U}^{\mathrm{T}}\mathbf{V}\right)\right\|_{\mathcal{F}}^{2}. \tag{14}$$

Finally, the missing elements of $\mathbf{X}$ are replaced with the corresponding elements of the generated matrix, i.e., $\mathbf{U}^{\mathrm{T}}\,\mathbf{V}$. This completes the matrix approximation.

TABLE 1. Attributes of the datasets.

Name	#Classes	#Instances	#Dimensions
ECG	6	5201	21
SCN	6	2407	294

TABLE 2. Stopping thresholds (average RMSEs) for iterative updating during training and testing.

Name	Threshold for Training	Threshold for Testing	Intermediate Dimension D
ECG	0.01	0.02	1–9
SCN	0.01	0.02	1–9

Empirical observations found that the stopping threshold for testing should be less strict than that for training

III. PROPOSED RIDGE ALS IMPUTATION BY ITERATIVE PROJECTION PURSUIT

Assume $\mathbf{X}$ is an N-by-M data matrix with missing values, and $\mathbf{y}$ is an N-by-1 vector containing the corresponding labels. Also assume that the number of classes is L. The proposed ALS classifier for supervised data imputation is based on typical ridge ALS matrix approximation, in which $\mathbf{X}$ is divided into $\mathbf{X}_\ell$, and $\ell = 1, .., L$. The size of $\mathbf{X}_\ell$ is $M \times N_\ell$, where $N_1 + N_2 + \ldots + N_L = N$.

Based on ridge regression, for each class, class-dependent matrix factors $\mathbf{U}_\ell$ and $\mathbf{V}_\ell$ are generated during the training phase as follows.

$$\mathbf{V}_\ell = \left(\mathbf{U}_\ell\mathbf{U}_\ell^{\mathrm{T}} + \rho_{\mathrm{U}_\ell}\mathbf{I}\right)^{-1}\mathbf{U}_\ell \times \mathrm{G}(\mathbf{X}_\ell) \tag{15}$$

and

$$\mathbf{U}_\ell = \left(\mathbf{V}_\ell\mathbf{V}_\ell^{\mathrm{T}} + \rho_{\mathrm{V}_\ell}\mathbf{I}\right)^{-1}\mathbf{V}_\ell \times \mathrm{G}(\mathbf{X}_\ell)^{\mathrm{T}}. \tag{16}$$

Only data matrix $\mathbf{X}_\ell$ can be used for formation of $\mathbf{U}_\ell$ and $\mathbf{V}_\ell$. Initialization of $\mathbf{U}_\ell$ in this study is based on class-dependent means plus a vector $\mathbf{Z}$ that contains small random numbers.

$$\mathbf{U}_\ell = \begin{bmatrix}\boldsymbol{\mu}_{\ell,1} & \boldsymbol{\mu}_{\ell,2} & \cdots & \boldsymbol{\mu}_{\ell,D}\end{bmatrix}^{\mathrm{T}} + \mathbf{Z}^{\mathrm{T}} \tag{17}$$

where

$$\boldsymbol{\mu}_{\ell,d} = \sum_{n_\ell=1}^{N_\ell}\mathrm{G}\left(X_{n_\ell}\right). \tag{18}$$

Moreover, d is the index of the intermediate dimension D, n_ℓ signifies the index of the samples in class ℓ, $\boldsymbol{\mu}$ denotes the mean, and X_{n_ℓ} is the n_ℓ-th column of $\mathbf{X}_{n_\ell}$. Let $\mathbf{t}$ denote a new M-by-1 sample without missing values. If $\mathbf{t}$ belongs to span($\mathbf{U}_\ell^{\mathrm{T}}$), this means a linear combination of the vectors in $\mathbf{U}_\ell^{\mathrm{T}}$ exists when $\mathbf{U}_\ell^{\mathrm{T}}$ contains all the bases. The combinational coefficients can be collected in a 1-by-D vector $\mathbf{v}$, such that $\mathbf{t} = \mathbf{U}_\ell^{\mathrm{T}} \times \mathbf{v}_\ell^{\mathrm{T}}$. Besides, $\mathbf{v}_\ell^{\mathrm{T}}$ belongs to $\mathbf{V}_\ell$, where $\mathbf{V}_\ell$ can be viewed as all the possible combinations of the vectors in $\mathbf{U}_\ell^{\mathrm{T}}$ when sufficient samples are observed during the training stage. Thus, $\mathbf{v}_\ell^{\mathrm{T}} = \mathbf{V}_\ell \times \mathbf{a}_\ell$, where $\mathbf{a}_\ell$ is an N_ℓ-by-1 coefficient vector.

TABLE 3. Imputation errors of the testing samples in the ECG dataset by using the poly2 kernel.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF	10.00	0.031054	0.073496	0.097509	0.112665	0.128017
	20.00	0.065481	0.142837	0.179057	0.190775	0.204206
	30.00	0.109287	0.189139	0.226438	0.245206	0.262241
Supervised CF (Proposed)	10.00	0.019570	0.026305	0.040242	0.065014	0.077310
	20.00	0.028409	0.041081	0.130007	0.135237	0.143096
	30.00	0.034180	0.049343	0.175808	0.190039	0.203908

TABLE 4. Imputation errors of the testing samples in the ECG dataset by using the poly3 kernel.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF	10.00	0.030648	0.066987	0.100171	0.115205	0.127747
	20.00	0.066490	0.142542	0.178273	0.192954	0.202889
	30.00	0.108313	0.187804	0.226551	0.245750	0.261222
Supervised CF (Proposed)	10.00	0.019764	0.026264	0.050023	0.064045	0.077331
	20.00	0.027704	0.041807	0.128675	0.137204	0.139920
	30.00	0.034270	0.048586	0.175638	0.191716	0.200432

TABLE 5. Imputation errors of the testing samples in the ECG dataset by using the RBF.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF	10.00	0.029345	0.067766	0.096179	0.113929	0.126411
	20.00	0.065254	0.141885	0.179022	0.197295	0.211543
	30.00	0.111573	0.190663	0.228869	0.248649	0.267108
Supervised CF (Proposed)	10.00	0.017542	0.020302	0.036196	0.056970	0.069664
	20.00	0.025731	0.040135	0.143274	0.143438	0.149798
	30.00	0.032101	0.046965	0.174711	0.183513	0.192027

TABLE 6. Imputation errors of the testing samples in the SCN dataset by using the poly2 kernel.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF	10.00	0.041477	0.038323	0.037121	0.037360	0.038477
	20.00	0.073973	0.071283	0.070724	0.072376	0.074611
	30.00	0.112607	0.111289	0.112330	0.114558	0.117705
Supervised CF (Proposed)	10.00	0.033560	0.030967	0.029176	0.028509	0.028310
	20.00	0.048554	0.045213	0.043957	0.043534	0.043616
	30.00	0.059667	0.057900	0.056655	0.055970	0.056738

However, $\mathbf{U}_\ell^{\mathrm{T}}$ and $\mathbf{V}_\ell$ are both low-rank matrices. The objective is to find $\mathbf{a}_\ell$, so that $\mathbf{U}_\ell^{\mathrm{T}}\mathbf{V}_\ell\mathbf{a}_\ell$ can sufficiently approximate $\mathbf{t}$. That is,

$$\mathbf{t} \approx \mathbf{U}_\ell^{\mathrm{T}}\mathbf{V}_\ell\mathbf{a}_\ell. \tag{19}$$

In fact, as $\mathbf{t}$ belongs to span($\mathbf{U}_\ell^{\mathrm{T}}$), the following representation is equivalent

$$\mathbf{t} \approx \mathbf{U}_\ell^{\mathrm{T}}\mathbf{V}_\ell\mathbf{a}_\ell = \mathbf{U}_\ell^{\mathrm{T}}\mathbf{v}_\ell^{\mathrm{T}}. \tag{20}$$

The system can determine the class of the new sample by measuring the reconstruction error.

$$\mathbf{e}_\ell = \mathbf{t} - \mathbf{U}_\ell^{\mathrm{T}}\mathbf{v}_\ell^{\mathrm{T}} \tag{21}$$

where

$$\mathbf{v}_\ell^{\mathrm{T}} = \left(\mathbf{U}_\ell\mathbf{U}_\ell^{\mathrm{T}} + \rho_\ell\mathbf{I}\right)^{-1}\mathbf{U}_\ell\mathbf{t}. \tag{22}$$

With recursive reconstruction and regression, the system can jointly approximate the testing sample and perform classification.

Equation (22) is the solution to ridge regression (RR), i.e.,

$$E_{\mathrm{RR}}\left(\mathbf{v}_\ell\right) = \left\|\mathbf{t} - \mathbf{U}_\ell^{\mathrm{T}}\mathbf{v}_\ell^{\mathrm{T}}\right\|_{\mathcal{F}}^2 + \rho_\ell\left\|\mathbf{v}_\ell^{\mathrm{T}}\right\|_{\mathcal{F}}^2. \tag{23}$$

For $\mathbf{t}$ with missing values, both data imputation and classification should be performed during the above-mentioned process. Iterative projection pursuit based on RR is therefore proposed in our algorithm. Iterative projection pursuit recursively examines the closest distance between the vectors, formed by the class-dependent basis matrices, and the testing incomplete vector. When the most similar vector is found, it is used to represent the incomplete vector. The procedure is listed as follows.

Step 1. Initialize $\hat{\mathbf{t}}_\ell$ by filling in the missing entries of $\mathbf{t}$ with zeros.

Step 2. Compute $\mathbf{v}_\ell$ for each class based on

$$\mathbf{v}_\ell^{\mathrm{T}}[i] = \left(\mathbf{U}_\ell\mathbf{U}_\ell^{\mathrm{T}} + \rho_\ell\mathbf{I}\right)^{-1}\mathbf{U}_\ell\hat{\mathbf{t}}_\ell[i] \tag{24}$$

where i specifies the i-th iteration.

TABLE 7. Imputation errors of the testing samples in the SCN dataset by using the poly3 kernel.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF	10.00	0.041744	0.038424	0.037175	0.037367	0.038225
	20.00	0.074079	0.070909	0.071000	0.072610	0.074959
	30.00	0.112788	0.111145	0.112393	0.114439	0.117350
Supervised CF (Proposed)	10.00	0.033948	0.031350	0.029500	0.028642	0.028198
	20.00	0.049006	0.045351	0.044131	0.043874	0.043858
	30.00	0.060083	0.056946	0.056475	0.056236	0.055510

TABLE 8. Imputation errors of the testing samples in the SCN dataset by using the RBF.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF	10.00	0.041586	0.038602	0.036968	0.037450	0.038178
	20.00	0.073974	0.071156	0.071290	0.072311	0.075109
	30.00	0.112717	0.110842	0.111769	0.113971	0.117808
Supervised CF (Proposed)	10.00	0.033482	0.031408	0.029348	0.028762	0.027996
	20.00	0.048459	0.045199	0.044070	0.043554	0.043801
	30.00	0.060082	0.056824	0.056222	0.055925	0.056241

TABLE 9. Summary of the imputation error.

Method	Missing Rate	Imputation Error		
		ECG	SCN	Average
Unsupervised CF	10.00%	0.087809	0.038565	0.112878
	20.00%	0.157367	0.072691	
	30.00%	0.207254	0.113581	
Supervised CF (Proposed)	10.00%	0.044436	0.030210	0.067134
	20.00%	0.097034	0.045079	
	30.00%	0.128882	0.057165	

Step 3. Impute the missing values by reconstructing $\mathbf{t}$.

$$\hat{\mathbf{t}}_\ell[i] = \mathbf{U}_\ell^{\mathrm{T}} \times \mathbf{v}_\ell^{\mathrm{T}}[i]. \tag{25}$$

$$\hat{\mathbf{t}}_\ell[i+1] = \mathbf{t} \oplus \hat{\mathbf{t}}_\ell[i]. \tag{26}$$

The operator $\oplus$ in (26) means to replace the missing values of $\mathbf{t}$ with the imputed ones of $\hat{\mathbf{t}}$.

Step 4. Repeat steps 2–4 until $\mathbf{e}$ converges

$$\mathbf{e}_\ell[i+1] = \mathrm{G}\left(\mathbf{t} - \hat{\mathbf{t}}_\ell[i+1]\right). \tag{27}$$

The following stopping criterion is used in our study.

$$\varepsilon_{\mathrm{RMSE}_\ell} = \sqrt{\left(\sum_{m=1}^{M} \mathbf{e}_{\ell,m}^2[i+1]\right)/M}. \tag{28}$$

Equation (28) represents average root-mean-square errors (RMSEs), where $\mathbf{e}_{\ell,m}$ denotes the m-th dimension of the error vector for class ℓ.

Step 5. Predicted class is determined by selecting ℓ with the minimum error.

$$\ell^* = \arg\min_{\ell} \varepsilon_\ell. \tag{29}$$

IV. EXPERIMENTAL RESULT

Experiments on open datasets were carried out for evaluating the performance. The information of these datasets is listed in Table 1. The first column shows the name. The rest

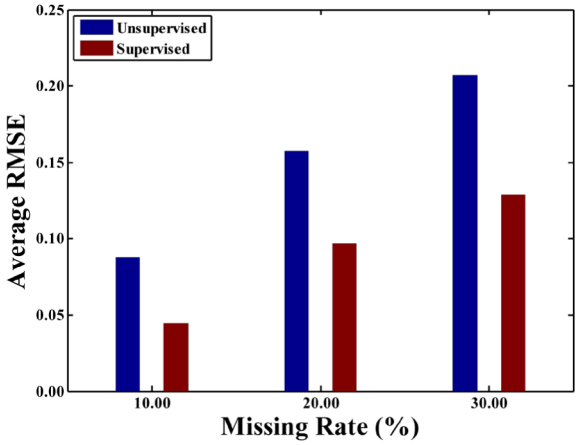

FIGURE 1. Imputation errors of the ECG database.

columns specify the number of classes, samples, and dimensions, respectively. Dataset "MIT/BIH ECG" is available at PhysioNet (www.physionet.org), and "Scene Classification (SCN)" [19] was downloaded from the official LibSVM website. The experiment used randomly selected 80.00% of the data for training, and the rest 20.00% were for testing. Both of the two datasets were normalized before further postprocessing.

The percentage of missing values ranged from 10.00% to 30.00%. Missing entries were randomly and uniformly generated. Notably, missing entries in the incomplete training data and the incomplete testing data were not exactly the same to guarantee randomness. Regarding ridge parameters, ρ_{U}, ρ_{V}, and ρ_ℓ were empirically all 0.50. Furthermore, the threshold of the average RMSEs for both unsupervised and supervised collaborative filtering was 0.01 during the training stage (see Table 2). The intermediate dimensions D during collaborative filtering were 1–9, with a separation of two. To further enhance the accuracy rates of classification, the imputed data by either unsupervised or super-

TABLE 10. Classification accuracy for the ECG dataset by using the poly2 kernel.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF + SVM	10.00	0.839423	0.836346	0.831346	0.816731	0.823846
	20.00	0.822500	0.809231	0.803846	0.796346	0.788077
	30.00	0.785000	0.787500	0.778654	0.773269	0.772692
Supervised CF + SVM (Proposed)	10.00	0.846154	0.843846	0.833077	0.828654	0.824615
	20.00	0.819615	0.790577	0.821346	0.806538	0.794808
	30.00	0.767885	0.759038	0.778654	0.773077	0.771154
Supervised CF (Proposed)	10.00	0.787115	0.781346	0.719231	0.725577	0.703654
	20.00	0.777115	0.770385	0.724615	0.731346	0.674615
	30.00	0.782115	0.760385	0.738846	0.679615	0.698269

TABLE 11. Classification accuracy for the ECG dataset by using the poly3 kernel.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF + SVM	10.00	0.891346	0.889808	0.875000	0.869615	0.858077
	20.00	0.862115	0.861923	0.850769	0.837115	0.830769
	30.00	0.834423	0.835577	0.834038	0.813654	0.814615
Supervised CF + SVM (Proposed)	10.00	0.891731	0.873846	0.862500	0.826538	0.819231
	20.00	0.850577	0.751923	0.791154	0.615769	0.607692
	30.00	0.794808	0.653846	0.802308	0.786731	0.755385
Supervised CF (Proposed)	10.00	0.798462	0.783269	0.696154	0.720769	0.698269
	20.00	0.796731	0.771346	0.763654	0.757885	0.688462
	30.00	0.778077	0.749808	0.741538	0.679423	0.659808

TABLE 12. Classification accuracy for the ECG dataset by using the RBF.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF + SVM	10.00	0.823269	0.826346	0.819038	0.813077	0.803269
	20.00	0.809231	0.807308	0.783077	0.780962	0.775962
	30.00	0.792692	0.785962	0.766538	0.766346	0.761154
Supervised CF + SVM (Proposed)	10.00	0.826538	0.828846	0.828077	0.808077	0.799423
	20.00	0.809423	0.802115	0.789615	0.783654	0.786154
	30.00	0.765192	0.766923	0.744231	0.759231	0.745192
Supervised CF (Proposed)	10.00	0.797115	0.754808	0.703654	0.741346	0.675000
	20.00	0.782308	0.762308	0.779423	0.679423	0.717308
	30.00	0.774038	0.757308	0.777308	0.664615	0.651346

TABLE 13. Classification accuracy for the SCN dataset by using the poly2 kernel.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF + SVM	10.00	0.711019	0.718503	0.714761	0.714345	0.718503
	20.00	0.684407	0.683992	0.683160	0.686071	0.679002
	30.00	0.647817	0.639085	0.632017	0.629106	0.619958
Supervised CF + SVM (Proposed)	10.00	0.733056	0.737630	0.743035	0.748857	0.764657
	20.00	0.730977	0.736798	0.732640	0.720582	0.733056
	30.00	0.698960	0.681497	0.696466	0.669023	0.662786
Supervised CF (Proposed)	10.00	0.674428	0.689397	0.711019	0.709771	0.748025
	20.00	0.689397	0.689397	0.713929	0.713514	0.708108
	30.00	0.688981	0.695218	0.699792	0.694802	0.686071

vised collaborative filtering were sent into Support Vector Machines (SVMs) for training. The kernels included the second-order polynomial (poly2) function, the third-order polynomial (poly3) function, and the radial basis function (RBF). For RBFs, the radius for ECG and SCN data was 1.00 and 10.00, respectively. In total, three systems were examined. One is typical unsupervised collaborative filtering with SVMs, another is the proposed ridge ALS imputation with SVMs, and the other is the proposed ridge ALS imputation without SVMs.

In the testing phase, the testing samples were imputed first by unsupervised and the proposed supervised collaborative

TABLE 14. Classification accuracy for the SCN dataset by using the poly3 kernel.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF + SVM	10.00	0.721414	0.735967	0.728067	0.730977	0.738877
	20.00	0.698545	0.703534	0.700208	0.693971	0.695634
	30.00	0.658628	0.652391	0.666112	0.660291	0.649480
Supervised CF + SVM (Proposed)	10.00	0.740125	0.742620	0.750936	0.750936	0.762578
	20.00	0.728898	0.730146	0.734304	0.746362	0.743035
	30.00	0.691892	0.702703	0.691060	0.673181	0.661954
Supervised CF (Proposed)	10.00	0.681081	0.688150	0.707277	0.725988	0.736383
	20.00	0.679834	0.686902	0.704782	0.720166	0.719751
	30.00	0.681913	0.696050	0.703534	0.698960	0.703119

TABLE 15. Classification accuracy for the SCN dataset by using the RBF.

Method	Missing Rate (%)	Intermediate Dimension D				
		1	3	5	7	9
Unsupervised CF + SVM	10.00	0.731809	0.733056	0.719751	0.725572	0.725988
	20.00	0.717256	0.720166	0.714761	0.730977	0.714761
	30.00	0.684823	0.681497	0.677339	0.683992	0.683576
Supervised CF + SVM (Proposed)	10.00	0.743451	0.744699	0.738046	0.739709	0.737630
	20.00	0.728898	0.729314	0.738462	0.722661	0.724324
	30.00	0.692723	0.664033	0.660707	0.637422	0.613306
Supervised CF (Proposed)	10.00	0.699376	0.689397	0.707277	0.720998	0.740125
	20.00	0.691476	0.695634	0.720166	0.708940	0.723909
	30.00	0.669854	0.685655	0.688565	0.696466	0.708524

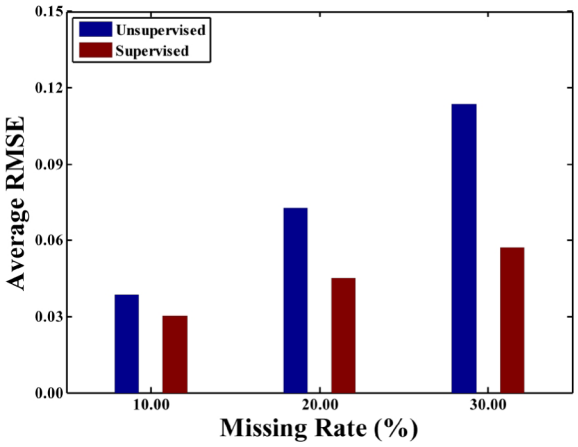

FIGURE 2. Imputation errors of the SCN database.

filtering, respectively. Subsequently, the imputed data were sent into the SVMs for classification. For the proposed supervised collaborative filtering at the testing stage, average RMSEs were used as the stopping criterion. It was empirically set to 0.02, less strict than that in the training phase. When the threshold is smaller, imputation errors become smaller. Notably, typical unsupervised collaborative filtering used smaller thresholds than the proposed method did.

The average imputation error after the test is summarized in Tables 3–8. The error was calculated based on the difference between the imputed values and the correct answers. The experimental results showed that the proposed supervised CF generated smaller errors than the unsupervised CF did in all the cases. The average imputation error of the proposed supervised CF was 0.067134, whereas that of the unsupervised CF was 0.112878. The proposed method decreased the imputation error by 40.52% on average. The detail is shown in Table 9, and the results are displayed in Figs. 1 and 2. For classification, observations revealed that the proposed supervised CF + SVM could achieve approximately the same accuracy as the unsupervised CF + SVM did. The former generated an accuracy rate of 75.37%, and the latter yielded an accuracy rate of 75.53% on average. When the proposed supervised CF did not employ SVMs, the accuracy was 71.94%. The numerical results are displayed in Tables 10–15.

V. CONCLUSION

This work presents supervised collaborative filtering, where class-dependent basis matrices are used for data imputation. The proposed ridge ALS imputation relies on class-dependent regression weights, derived from coefficient matrices, to jointly impute and classify new incomplete samples. To search the weight, this study develops iterative projection pursuit. It recursively examines the closest distance between the vectors, formed by the class-dependent basis matrices, and the testing incomplete vector.

Experiments on open datasets were conducted to compare the performance between the proposed system and the baseline, including imputation errors and classification accuracy. The experimental results showed that the proposed method yielded smaller errors than the baseline while simultaneously maintaining classification accuracy. Such findings verified the effectiveness of the proposed method.

REFERENCES

[1] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*. New York, NY, USA: Wiley, 1987.

[2] A. Mehmood, I. Natgunanathan, Y. Xiang, G. Hua, and S. Guo, "Protection of big data privacy," *IEEE Access*, vol. 4, pp. 1821–1834, Apr. 2016.
[3] B. M. Gaff, H. E. Sussman, and J. Geetter, "Privacy and big data," *IEEE Comput. Mag.*, vol. 47, no. 6, pp. 7–9, Jun. 2014.
[4] D. Quick and K.-K. R. Choo, "Digital droplets: Microsoft SkyDrive forensic data remnants," *Future Generat. Comput. Syst.*, vol. 29, no. 6, pp. 1378–1394, Aug. 2013.
[5] C. Anagnostopoulos and P. Triantafillou, "Scaling out big data missing value imputations: Pythia vs. godzilla," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, Aug. 2014, pp. 651–660.
[6] J. L. Schafer, *Analysis of Incomplete Multivariate Data*. New York, NY, USA: Chapman & Hall, 1997.
[7] O. Troyanskaya *et al.*, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.
[8] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Jose, CA, USA, Aug. 2007, pp. 39–42.
[9] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the Netflix prize," in *Proc. 4th Int. Conf. Algorithmic Appl. Manage.*, Shanghai, China, Jun. 2008, pp. 337–348.
[10] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.
[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[12] J.-H. Wang, P. K. Hopke, T. M. Hancewicz, and S. L. Zhang, "Application of modified alternating least squares regression to spectroscopic image analysis," *Anal. Chim. Acta*, vol. 476, no. 1, pp. 93–109, Jan. 2003.
[13] 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.
[14] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. 14th Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Denver, CO, USA, Nov. 2000, pp. 556–562.
[15] M. Chu, F. Diele, R. Plemmons, and S. Ragni, "Optimality, computation, and interpretation of nonnegative matrix factorizations," Dept. Math., North Carolina State Univ., Raleigh, NC, USA, Tech. Rep., Oct., 2004.
[16] C.-C. Ma and L. Chih-Jen, "Large-scale collaborative filtering algorithms: A guide to singular value decomposition for collaborative filtering," M.S. thesis, Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Jun. 2008.
[17] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Philadelphia, PA, USA, Aug. 2006, pp. 126–135.
[18] S.-Y. Kung, *Kernel Methods and Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, Jun. 2014.
[19] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

BO-WEI CHEN (M'14–SM'17) is currently with the School of Information Technology, Monash University, Australia. His research interests include data analytics, machine learning, and audiovisual sensor networks. He serves as the Chair of the Signal Processing Chapter with the IEEE Harbin Section.

WEN JI (M'09–SM'16) is currently an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences, China. Her research areas include video communication and networking, video coding, channel coding, information theory, optimization, network economics, and pervasive computing. She is the Vice Chair of the Signal Processing Chapter with the IEEE Harbin Section.

SEUNGMIN RHO (M'08) is currently a Faculty Member with the Department of Media Software, Sungkyul University, South Korea. His research interests include database, big data analysis, music retrieval, multimedia systems, machine learning, knowledge management, and computational intelligence.

YU GU is currently a Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, China, where he is also the Head.