Face Image Modeling by Multilinear Subspace Analysis with Missing Values

Xin Geng, Kate Smith-Miles, *Senior Member, IEEE*,
Zhi-Hua Zhou, *Senior Member, IEEE*, Liang Wang, *Senior Member, IEEE*

***Abstract*— Multilinear subspace analysis (MSA) is a promising methodology for pattern recognition problems due to its ability in decomposing the data formed from the interaction of multiple factors. MSA requires a large training set, well organized in a single tensor, which consists of data samples with all possible combinations of the contributory factors. However, such a 'complete' training set is difficult (or impossible) to obtain in many real applications. The missing value problem is therefore crucial to the practicality of MSA, but has hardly been investigated up to the present. To solve the problem, this paper proposes an algorithm named M^{2}SA, which is advantageous in real applications since: 1) it inherits the ability of MSA to decompose the interlaced semantic factors; 2) it does not depend on any assumptions on the data distribution; 3) it can deal with a high percentage of missing values. M^{2}SA is evaluated by face image modeling on two typical multifactorial applications: face recognition and facial age estimation. Experimental results show the effectiveness of M^{2}SA even when the majority of values in the training tensor are missing.**

***Index Terms*— Multilinear subspace analysis, Missing values, Face recognition, Facial age estimation.**

I. Introduction

Multiway data analysis [2] is essentially the extension of vector (1st-order tensor) or matrix (2nd-order tensor) analysis to higher-order tensor analysis. It has been shown in many research areas that organizing raw data in matrices or vectors might result in loss of crucial information. For instance, in image analysis, an image is often transformed into a vector by raster scan. A set of images can then be represented by a matrix ready for two-way data analysis methods such as PCA [17]. However, concatenating the pixels row by row or column by column will lose important spatial information in the original image. A more natural way is to represent each image by a matrix and the image set by a third-order tensor. Multiway data analysis methods are designed to handle such higher-order tensors. Compared with two-way analysis methods, multiway data analysis methods have advantages in uniqueness, robustness to noise, ease of interpretation, etc. [2]. Consequently, they have been increasingly popular in various areas in finding the hidden structures and capturing the underlying correlations between the variables.

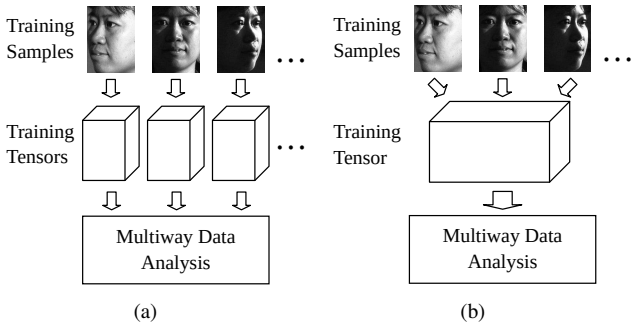

Fig. 1. Two different ways in which the training data are organized for multiway data analysis: (a) Each training sample is transformed into a tensor; (b) All training samples are organized in a single tensor.

Taking the area of computer vision for example, since almost all computer vision problems are related to images or videos which can be inherently represented by tensors, many multiway data analysis methods have been proposed in this area during recent years. Existing methods can be roughly divided into two categories according to the way in which the training data are organized. The first kind of approaches treats each sample as a tensor, as shown in Fig. 1(a). For example, Xu et al. [41] [40] proposed Concurrent Subspaces Analysis (CSA) to derive representative subspaces from images encoded in higher-order tensors, such as video sequences and Gabor filtered images; Yan et al. [44] proposed Multilinear Discriminant Analysis (MDA) to solve the supervised dimensionality reduction problem for face recognition; Lei et al. [23] proposed a Canonical Correlation Analysis (CCA) based mapping from the tensor space of the near infrared (NIR) faces to that of the 3D faces, which was applied to face shape recovery from a single image; Huang and Ding [16] proposed a tensor factorization method using R_1 norm rather than L_2 norm (sum of squared errors) for error accumulation function, which can effectively handle the outliers in applications like face representation and reconstruction; Zhang et al. [47] proposed a directional multilinear extension of Independent Component Analysis (ICA) and achieved better performance than the conventional ICA in face recognition and palmprint recognition.

The second kind of approaches organizes all the training samples in a single tensor, as shown in Fig. 1(b), usually with each mode (also referred to as *dimension* or *way* [2])

Manuscript received xx xxx, 2010; revised xx xxx, xxxx.

Xin Geng is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the School of Mathematical Sciences, Monash University, VIC 3800, Australia (e-mail: xin.geng@sci.monash.edu.au).

Kate Smith-Miles is with the School of Mathematical Sciences, Monash University, VIC 3800, Australia (e-mail: kate.smith-miles@sci.monash.edu.au).

Zhi-Hua Zhou is with the National Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, China (e-mail: zhouzh@nju.edu.cn).

Liang Wang is with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wangliang@nlpr.ia.ac.cn).

explicitly corresponding to a semantic factor contributing to the problem. For instance, Vasilescu and Terzopoulos [35] [36] [37] proposed Multilinear Subspace Analysis (MSA) to decompose the tensor of face images with each mode corresponding to one variable factor in face recognition, such as identity, view angle, illumination, expression, etc.; Lin et al. [25] proposed a tensor factorization method to simultaneously solve for the unknown identity and lighting parameters in the face images; Li et al. [24] proposed an iterative factorization method based on kernel mappings for face image synthesis and recognition; Park and Savvides [28] proposed Individual Kernel Tensor-Subspaces for face recognition, which does not require tensor factorization; Rana et al. [29] proposed a face recognition method exploiting the interaction of all the subspaces resulting from multilinear decomposition (by both multilinear PCA and ICA), which performs well on test faces generated with previously unseen illumination conditions and view angles.

The first category of multiway data analysis methods can be viewed as the generalization from vector (1st-order tensor) sample analysis to higher-order tensor sample analysis. Up to the present, most important vector analysis methods, such as PCA, LDA, and ICA, have been evolved to their multiway versions. The second category, however, has the advantage in extracting the semantic factors contributing to the problem. Natural data (e.g., images) generally result from the interaction of multiple semantic factors. Usually only one or two factors are of interest in a particular problem, and all the others are regarded as interferences. For example, in face recognition, the face images might vary in identity, expression, pose, and illumination. But the only goal is the recognition of identity, regardless of other variations. The traditional PCA-based Eigenface [34] method makes an assumption to simplify the problem, i.e., the most apparent variation among the images is caused by and only by the difference of identity. Actually, PCA models the input data according to only one factor, i.e., the variance of the data. This creates the gap between the statistical factor (variance) and the semantic factors (identity, expression, pose, and illumination). Rather than more efficiently encoding the samples, the methods in the second category aim to extract the unique semantic features from the training ensemble. Among these methods, a relatively early and representative approach is *Multilinear Subspace Analysis* (MSA) [35] [36] [37]. Through the application of *N-mode SVD* (also known as Higher Order Singular Value Decomposition, abbr. HOSVD) [21], a multilinear extension of the matrix SVD on the tensor, MSA decomposes the modes of the tensor and therefore separates and parsimoniously represents each of the semantic factors associated with the tensor modes. Then each data sample can be represented by a set of coefficient vectors, one for each semantic factor. For a particular problem, only the coefficient vector(s) accounting for the factor(s) of interest are used. Thus the influence of interferential factors can be filtered out.

Despite the beauty in theory, there are practical problems in MSA. The most prominent one might be the possible missing values in the training tensor. Although the general problem of matrix and tensor completion has recently attracted a lot of attention [26], not much work has been done for tensor decomposition with missing values. Returning to the history of linear PCA, the missing value problem has long been recognized as an important practical issue and has been intensively investigated. For example, Wiberg [39] suggested to minimize the squared approximation error (as in the standard SVD) with the summation only over those available values. Roweis [30] presented an Expectation-Maximization (EM) algorithm for PCA, which can naturally accommodate the missing values in the training data. Tipping and Bishop [32] dealt with the missing value problem by building a probabilistic model for PCA (PPCA) with certain Gaussian prior assumptions. Although MSA is a multilinear extension of PCA, to the best of our knowledge, no work has been done so far to deal with the missing values in the training tensor of MSA.

In fact, the missing value problem in MSA is much more common than that in PCA. In addition to the same situation PCA might encounter when some of the values in the training samples are missing due to data acquisition, transmission or storage problems, the following reason makes the missing values more likely to appear in MSA. Instead of a set of samples, the training data of MSA is a single well organized tensor. To fill all the positions in the tensor, a large number of samples with all combinations of the contributory factors are needed. For instance, a four-factor problem with each factor having 10 different values will require a training tensor consisting of 10^4 samples. In many real applications, unfortunately, it is very hard (or impossible) to obtain such a large 'complete' data ensemble. A typical example is facial age estimation [11], where the collection of images at all ages of interest (say 0-60 years old) from each person is impractical. In some other applications, even when the collection of samples with all kinds of variations is possible, the clients may wish to reduce the costs by using as few as possible training samples without noticeable performance deterioration. In such cases, the available training samples might only account for a small portion of the quantity required to compose a 'complete' training tensor. Thus the algorithm must be able to work on a tensor with massive missing values even when there is no missing value in the individual training samples. The missing value problem is therefore crucial for the practicability of MSA.

To solve the problem, we proposed a method in our preliminary work [10] [9] called M^{2}SA (Multilinear Subspace Analysis with Missing values), which is an extension of MSA to handle missing values. After that, an algorithm called CP-WOPT [1] was recently proposed to deal with the missing data problem in the CANDECOMP/PARAFAC (CP) tensor decomposition [4] [14]. The main differences between M^{2}SA and CP-WOPT are:

1) M^{2}SA operates on a Tucker decomposition [33] of a tensor, while CP-WOPT is based on the CANDECOMP/PARAFAC (CP) decomposition. Tucker and CP are two major types of tensor decomposition. Tucker may be regarded as a more flexible CP model because in the CP model, the core tensor is restricted to be diagonal (refer to $\mathcal{Z}$ in Eq. (1)).
2) The columns of the factor matrices (refer to $\mathbf{U}_n(n =$

$1, 2, \ldots, N$) in Eq. (1)) obtained by M^{2}SA are orthonormal while the factor matrices obtained by CP-WOPT may have linearly dependent columns. Generally speaking, orthonormal feature bases are more preferable than linearly dependent feature bases due to their advantages in expressiveness and computational efficiency.

3) CP-WOPT relies on external optimization methods (Nonlinear Conjugate Gradient (NCG) was used in [1]) to solve the weighted least squares problem for the CP model. Choosing a suitable optimization method for a particular application is empirical. On the other hand, M^{2}SA solves the problem by iteratively imputing the missing values, which does not require any external optimization methods.

This paper extends our preliminary work [10] [9] by giving more comprehensive description, analysis, and evaluation of M^{2}SA . Instead of minimizing the reconstruction error of the whole training tensor, M^{2}SA finds the approximation that can best reconstruct the available values in the tensor. Here the reconstruction error is defined as the Frobenius norm of the difference tensor between the original training tensor and the approximated tensor generated from the multilinear subspace. The missing values may appear anywhere in the tensor, and they may even account for the majority of the tensor. This matches many real applications where 1) the available training data for each class are not 'variation-complete', i.e., the data at certain variations are available for some classes, but not for others, and/or 2) only a small portion of all possible combinations of the variations are available as the training data. M^{2}SA is tested by face image modeling on two typical multifactorial applications. First, it is applied to face recognition, where the number of the training face images are gradually reduced in order to examine the performance for missing data. Second, it is used for facial age estimation, where the training tensor inherently contains massive missing values. M^{2}SA performs significantly better in both applications compared to a wide collection of existing algorithms.

The remainder of the paper is organized as follows. Section II introduces the fundamentals of multilinear algebra and the notations used in this paper. Section III proposes the M^{2}SA algorithm to decompose tensors with missing values. Experiments on face recognition are reported in Section IV, and those on facial age estimation are reported in Section V. Finally, conclusions are drawn in Section VI.

II. Fundamentals and Notations of Multilinear Algebra

Just as linear algebra is built on the concept of vectors and the theory of vector spaces, multilinear algebra deals with the tensors in the multilinear space. Tensors are higher-order generalizations of scalars (zero-order tensors), vectors (first-order tensors), and matrices (second-order tensors). The extra structure of a tensor endows it with the inherent advantage in representing real-world data, which usually result from the interaction of multiple factors. In this paper, lowercase italic letters $(a, b, \ldots)$ denote scalars, bold lowercase letters $(\boldsymbol{a}, \boldsymbol{b}, \ldots)$ denote vectors, bold uppercase letters $(\mathbf{A}, \mathbf{B}, \ldots)$ denote matrices, and calligraphic uppercase letters $(\mathcal{A}, \mathcal{B}, \ldots)$ denote tensors. Each dimension of a tensor is called a *mode* (or a way) and the number of variables in each mode indicates the dimensionality of a mode. The *order* of a tensor is determined by the number of its modes. An N^{th}-order tensor can be denoted by $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where its dimensionalities in the first, second, ..., and N^{th} mode are I_1, I_2, ..., and I_N respectively. An element of $\mathcal{A}$ is denoted by $\mathcal{A}_{i_1 i_2 \ldots i_N}$ or $a_{i_1 i_2 \ldots i_N}$, where $1 \leq i_n \leq I_n, n = 1, 2, \ldots, N$.

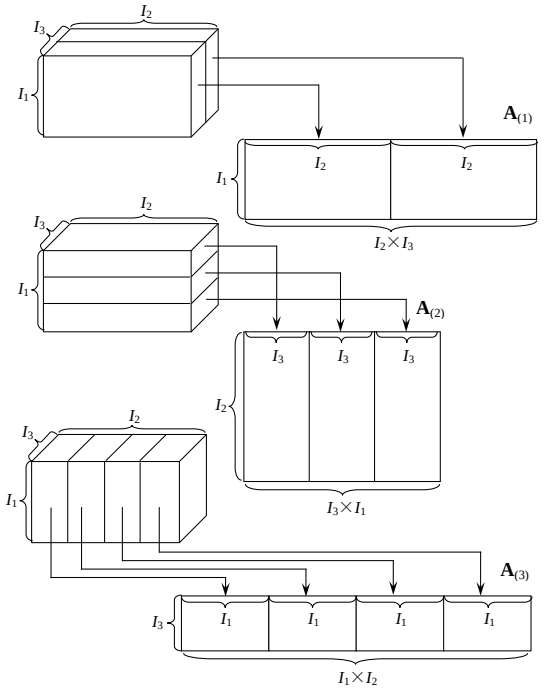

Fig. 2. Flattening a third-order tensor $\mathcal{A}$ into matrices along different modes. The mode-one flattened matrix $\mathbf{A}_{(1)} \in \mathbb{R}^{I_1 \times (I_2 I_3)}$, the mode-two flattened matrix $\mathbf{A}_{(2)} \in \mathbb{R}^{I_2 \times (I_3 I_1)}$, and the mode-three flattened matrix $\mathbf{A}_{(3)} \in \mathbb{R}^{I_3 \times (I_1 I_2)}$ are composed by the mode-one vectors, mode-two vectors, and mode-three vectors of $\mathcal{A}$, respectively.

The *mode-n vectors* of $\mathcal{A}$ are the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying the index i_n while keeping other indices fixed to certain values. A tensor $\mathcal{A}$ can be *flattened* into matrices in different ways. The *mode-n flattened matrix* of $\mathcal{A}$, denoted by $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_N)}$, is obtained by parallel concatenating all the mode-n vectors of $\mathcal{A}$. Each column of $\mathbf{A}_{(n)}$ corresponds to a mode-n vector of $\mathcal{A}$. An example of flattening a third-order tensor in three ways is shown in Fig. 2. The *mode-n rank* of $\mathcal{A}$, denoted by R_n, is defined as the dimensionality of the vector space generated by the mode-n vectors: $R_n = rank_n(\mathcal{A}) = rank(\mathbf{A}_{(n)})$.

A tensor can be multiplied by a matrix. The *mode-n product* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$ and a matrix $\mathbf{M} \in \mathbb{R}^{J_n \times I_n}$ is denoted by $\mathcal{B} = \mathcal{A} \times_n \mathbf{M}$. Note that the number of columns in $\mathbf{M}$ must equal to the dimensionality of the n-th mode of $\mathcal{A}$. The result $\mathcal{B}$ is a tensor of dimensionality $\mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$, whose entries are $\mathcal{B}_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} m_{j_n i_n}$, where $m_{j_n i_n}$ is the element of $\mathbf{M}$ at the position (j_n, i_n). Alternatively, $\mathcal{B}$ can also be calculated by re-tensorizing the matrix

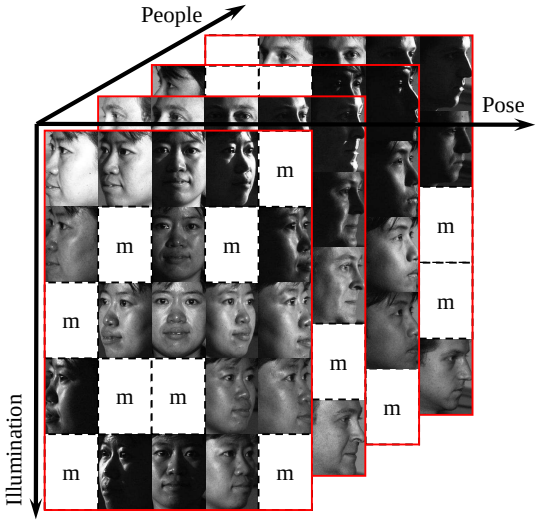

Fig. 3. The tensor representation of a subset from the CMU PIE database. The missing parts are labeled by 'm'.

$$\mathbf{B}_{(n)} = \mathbf{M}\mathbf{A}_{(n)}.$$

III. Decomposition of Incomplete Tensors

A tensor is a natural structure for the data resulting from the interaction of multiple factors. Each mode of the tensor corresponds to one factor. A typical tensor representation of a set of face images is shown in Fig. 3. The face images are a subset of the CMU PIE database [31], which vary in identity, pose and illumination. But not all combinations of the three kinds of variations are available. The images are assembled into a fourth-order tensor, with the first three modes corresponding to identity, pose and illumination, respectively, as shown in Fig. 3, and the fourth mode corresponding to the features extracted from the images (shown in Fig. 3 as images for better display). The missing variation combinations cause missing values in the tensor, marked by 'm' in the figure. The goal of M^{2}SA is to decompose such incomplete tensors into parsimonious features of the semantic factors represented by the modes of the tensors.

A. N-Mode Dimensionality Reduction

Suppose a tensor $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times I_{N+1}}$ consist of samples formed from N factors. Note that the $(N+1)$-th mode is used to store the features extracted from the samples. The N-mode SVD algorithm [36] can be used to decompose $\mathcal{D}$ as the mode-n product of N orthogonal spaces:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \cdots \times_N \mathbf{U}_N, \qquad (1)$$

where $\mathcal{Z}$ is called the *core tensor*, and the mode matrices $\mathbf{U}_n (n = 1, 2, \ldots, N)$ contain the orthonormal vectors spanning the factor spaces, one for each contributory factor corresponding to one mode of $\mathcal{D}$. $\mathcal{Z}$ is analogous to the diagonal singular value matrix in the conventional matrix SVD, but with a more complex $(N+1)^{th}$-order tensor structure. It governs the interaction between the factor spaces. Basically, N-mode SVD applies the matrix SVD to each of the mode-n flattened matrices $\mathbf{D}_{(n)} (n = 1, 2, \ldots, N)$ of $\mathcal{D}$, obtains the left matrix of the SVD as $\mathbf{U}_n$ for each n, and then computes the core tensor

$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T. \qquad (2)$$

In order to get a compact representation of the contributory factors, the dimensionality of the decomposed orthogonal spaces can be reduced. However, the optimal dimensionality reduction in multilinear analysis (operating on tensors) is not as simple as that in PCA (operating on matrices) by directly removing those eigenvectors associated with the smallest eigenvalues. The *N-mode dimensionality reduction* algorithm [36] is summarized in Algorithm 1. The goal is to find a best rank-$(R_1, R_2, \ldots, R_N)$ approximation $\hat{\mathcal{D}} = \hat{\mathcal{Z}} \times_1 \hat{\mathbf{U}}_1 \times_2 \hat{\mathbf{U}}_2 \cdots \times_N \hat{\mathbf{U}}_N$, with orthonormal mode matrices $\hat{\mathbf{U}}_n$ of lower rank $R_n < I_n$ for $n = 1, 2, \ldots, N$.

Algorithm 1: N-Mode Dimensionality Reduction

Input: $\mathcal{D}$ and the target rank $(R_1, R_2, \ldots, R_N)$
Output: Rank-reduced approximation $\hat{\mathcal{D}}$

Apply N-mode SVD algorithm to $\mathcal{D}$;
Truncate each mode matrix $\mathbf{U}_n$ to R_n columns, obtain the initial mode matrices $\mathbf{U}_1^0, \mathbf{U}_2^0, \ldots, \mathbf{U}_N^0$;
$i \leftarrow 0$;
repeat
  $i \leftarrow i + 1$;
  for $n \leftarrow 1$ **to** N **do**
    $\tilde{\mathcal{U}}_n^i \leftarrow \mathcal{D} \times_1 (\mathbf{U}_1^i)^T \cdots \times_{n-1} (\mathbf{U}_{n-1}^i)^T \times_{n+1} (\mathbf{U}_{n+1}^{i-1})^T \cdots \times_N (\mathbf{U}_N^{i-1})^T$;
    Mode-n flatten tensor $\tilde{\mathcal{U}}_n^i$ to obtain $\tilde{\mathbf{U}}_n^i$;
    Set $\mathbf{U}_n^i$ to the first R_n columns of the left matrix of the SVD of $\tilde{\mathbf{U}}_n^i$;
  end
until $\|(\mathbf{U}_n^i)^T \mathbf{U}_n^{i-1}\| > (1 - \varepsilon) R_n \ (n = 1, 2, \ldots, N)$;
$\hat{\mathbf{U}}_n \leftarrow \mathbf{U}_n^i \ (n = 1, 2, \ldots, N)$;
$\hat{\mathcal{Z}} \leftarrow \tilde{\mathcal{U}}_N^i \times_N \hat{\mathbf{U}}_N^T$;
$\hat{\mathcal{D}} \leftarrow \hat{\mathcal{Z}} \times_1 \hat{\mathbf{U}}_1 \times_2 \hat{\mathbf{U}}_2 \cdots \times_N \hat{\mathbf{U}}_N$;

B. Dealing with Missing Values

In practice, there may be a significant number of missing values in the tensor $\mathcal{D}$, preventing the direct application of Algorithm 1. In order to deal with this problem, here we propose the M^{2}SA algorithm. Suppose the index for the available values in $\mathcal{D}$ is $\mathcal{I}$, which is also a tensor of the same size. $\mathcal{I}_{i_1 i_2 \ldots i_N} = 1$ if $\mathcal{D}_{i_1 i_2 \ldots i_N}$ is available, otherwise, $\mathcal{I}_{i_1 i_2 \ldots i_N} = 0$. Instead of finding a best approximation for $\mathcal{D}$, the goal is changed into finding a best approximation for the available values, i.e., finding a low-rank $\hat{\mathcal{D}}$ which minimizes the reconstruction error of the available values

$$\Delta_a = \|(\mathcal{D} - \hat{\mathcal{D}}).\times \mathcal{I}\|, \qquad (3)$$

where the subscript 'a' denotes the available values, '$.\times$' represents the element-wise multiplication, and $\|\cdot\|$ represents the Frobenius norm of a tensor. M^{2}SA uses an iterative process to gradually reduce Δ_a. When initializing, each missing value

is filled by the mean over all the available values sharing some contributory factors with the missing value. Then the N-mode dimensionality reduction algorithm is applied to the fulfilled tensor to obtain the initial mode matrices $\hat{\mathbf{U}}_n^0$ and the core tensor $\hat{\mathcal{Z}}^0$ $(n = 1, 2, \ldots, N)$. The initial reconstruction of $\mathcal{D}$ is therefore $\hat{\mathcal{D}}^0 = \hat{\mathcal{Z}}^0 \times_1 \hat{\mathbf{U}}_1^0 \times_2 \hat{\mathbf{U}}_2^0 \cdots \times_N \hat{\mathbf{U}}_N^0$. In iteration i, the missing values of $\mathcal{D}$ are updated by the corresponding reconstructions:

$$\mathcal{D}^i = \mathcal{D}.\times\mathcal{I} + \hat{\mathcal{D}}^{i-1}.\times(\sim\mathcal{I}), \tag{4}$$

where $\sim$ is the boolean NOT operator. After that, the N-mode dimensionality reduction algorithm is applied to the updated tensor $\mathcal{D}^i$ to obtain the new mode matrices $\hat{\mathbf{U}}_n^i$ and the new core tensor $\hat{\mathcal{Z}}^i$. The whole procedure repeats until Δ_a becomes smaller than a predefined threshold ε or reaches the maximum number of iteration τ. The convergence of this process is proved as follows.

Proof: Suppose in iteration i, the training tensor is $\mathcal{D}^i$. By applying Algorithm 1, the approximation of $\mathcal{D}^i$ is calculated as $\hat{\mathcal{D}}^i = \mathcal{Z}^i \times_1 \mathbf{U}_1^i \times_2 \mathbf{U}_2^i \cdots \times_n \mathbf{U}_n^i \cdots \times_N \mathbf{U}_N^i$. The reconstruction error of $\mathcal{D}^i$ by $\hat{\mathcal{D}}^i$ is Δ^i, and that of the available features is Δ_a^i.

In the next iteration $i+1$, the updated training tensor $\mathcal{D}^{i+1}$ is obtained by replacing the missing values in $\mathcal{D}^i$ with the corresponding parts in $\hat{\mathcal{D}}^i$, so

$$\|(\mathcal{D}^{i+1} - \hat{\mathcal{D}}^i).\times\mathcal{I}\| = \|(\mathcal{D}^i - \hat{\mathcal{D}}^i).\times\mathcal{I}\|, \tag{5}$$

$$\|(\mathcal{D}^{i+1} - \hat{\mathcal{D}}^i).\times(\sim\mathcal{I})\| = 0. \tag{6}$$

Therefore,

$$\begin{aligned} \Delta_a^i &= \|(\mathcal{D}^i - \hat{\mathcal{D}}^i).\times\mathcal{I}\| \\ &= \|(\mathcal{D}^{i+1} - \hat{\mathcal{D}}^i).\times\mathcal{I}\| + \|(\mathcal{D}^{i+1} - \hat{\mathcal{D}}^i).\times(\sim\mathcal{I})\| \\ &= \|\mathcal{D}^{i+1} - \hat{\mathcal{D}}^i\|. \end{aligned} \tag{7}$$

As proved in [22], Algorithm 1 can find the rank-$(R_1, R_2, \ldots, R_N)$ approximation $\hat{\mathcal{D}}$ which minimizes the reconstruction error, i.e., for iteration $i+1$,

$$\begin{aligned} \hat{\mathcal{D}}^{i+1} &= \arg\min_{\hat{\mathcal{D}}} \|\mathcal{D}^{i+1} - \hat{\mathcal{D}}\| \\ &\quad \text{s.t. } \hat{\mathcal{D}} \text{ is of rank-}(R_1, R_2, \ldots, R_N) \end{aligned} \tag{8}$$

Since $\hat{\mathcal{D}}^i$ is also of rank-$(R_1, R_2, \ldots, R_N)$, then

$$\Delta^{i+1} = \|\mathcal{D}^{i+1} - \hat{\mathcal{D}}^{i+1}\| \le \|\mathcal{D}^{i+1} - \hat{\mathcal{D}}^i\| = \Delta_a^i. \tag{9}$$

Combining Eq. (9) with the trivial inequality $\Delta_a^{i+1} \le \Delta^{i+1}$ leads to

$$\Delta_a^{i+1} \le \Delta^{i+1} \le \Delta_a^i. \tag{10}$$

The first term equals to the second one only when the missing values in $\mathcal{D}^{i+1}$ do not change after the reconstruction. Thus the algorithm will converge to minimize Δ_a. ■

Fig. 4 shows the reconstruction errors of the available features Δ_a^i and those of all the features Δ^i for the first five iterations ($i = 1 \ldots 5$) when training M^{2}SA on the CMU PIE database with 10%, 50%, and 90% missing values in the training tensor. As can be seen from the figure that Inequality (10) always holds. Another observation is that the algorithm will converge more slowly with the increase of missing values (notice the different scale of the vertical axes).

Algorithm 2: M^{2}SA

Input: $\mathcal{D}$, $\mathcal{I}$, and the target rank $(R_1, R_2, \ldots, R_N)$
Output: Rank-reduced approximation $\hat{\mathcal{D}}$

1 Fill each missing value in $\mathcal{D}$ with the mean over all the available values sharing some contributory factors to obtain the initialized training tensor $\mathcal{D}^0$;
2 Apply Algorithm 1 to $\mathcal{D}^0$ to get the initial low-rank approximation $\hat{\mathcal{D}}^0 = \hat{\mathcal{Z}}^0 \times_1 \hat{\mathbf{U}}_1^0 \times_2 \hat{\mathbf{U}}_2^0 \cdots \times_N \hat{\mathbf{U}}_N^0$;
3 $i \leftarrow 0$;
4 **repeat**
5 $\quad i \leftarrow i + 1$;
6 $\quad \mathcal{D}^i \leftarrow \mathcal{D}.\times\mathcal{I} + \hat{\mathcal{D}}^{i-1}.\times(\sim\mathcal{I})$;
7 $\quad$ Apply Algorithm 1 to $\mathcal{D}^i$ to obtain the new low-rank approximation $\hat{\mathcal{D}}^i = \hat{\mathcal{Z}}^i \times_1 \hat{\mathbf{U}}_1^i \times_2 \hat{\mathbf{U}}_2^i \cdots \times_N \hat{\mathbf{U}}_N^i$;
8 $\quad \Delta_a^i \leftarrow \|(\mathcal{D}^i - \hat{\mathcal{D}}^i).\times\mathcal{I}\|$;
9 **until** $\Delta_a^i < \varepsilon$ **or** $i > \tau$;
10 $\hat{\mathcal{D}} \leftarrow \hat{\mathcal{Z}}^i \times_1 \hat{\mathbf{U}}_1^i \times_2 \hat{\mathbf{U}}_2^i \cdots \times_N \hat{\mathbf{U}}_N^i$;

The M^{2}SA algorithm is summarized in Algorithm 2. As can be seen, there are no pre-assumptions in this algorithm (such as Gaussian prior distributions), and the reconstruction error is guaranteed to descend with each iteration until convergence. Through Algorithm 2, an incomplete tensor $\mathcal{D}$ can be decomposed into the mode-n product of N mode matrices, as shown in Eq. (1). The most attractive property of Eq. (1) is that it provides an unique way to represent each value of any contributory factor, regardless of other factors, with the same coefficient vector. In detail, suppose the target factor f_t corresponds to the t-th mode, and the k-th value of f_t is $f_t(k)$. Then all the samples labeled by $f_t(k)$ compose a subtensor of $\mathcal{D}$, which is obtained through fixing the t-th index to k, and varying the other N indices. Denote this subtensor by $\mathcal{D}_{f_t(k)}$, then

$$\begin{aligned} \mathcal{D}_{f_t(k)} &= \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_{t-1} \mathbf{U}_{t-1} \\ &\quad \times_t (\boldsymbol{u}_t^k)^T \times_{t+1} \mathbf{U}_{t+1} \cdots \times_N \mathbf{U}_N \\ &= \mathcal{B} \times_t (\boldsymbol{u}_t^k)^T, \end{aligned} \tag{11}$$

where $\mathcal{B} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_{t-1} \mathbf{U}_{t-1} \times_{t+1} \mathbf{U}_{t+1} \cdots \times_N \mathbf{U}_N$ is a constant tensor, and $(\boldsymbol{u}_t^k)^T$ is the k-th row vector of $\mathbf{U}_t$. Since $\mathcal{B}$ is a constant, $\mathcal{D}_{f_t(k)}$ is totally determined by $(\boldsymbol{u}_t^k)^T$. Therefore, each row vector in $\mathbf{U}_t$ can be regarded as a feature vector uniquely representing all the samples associated with one value of the target factor f_t, no matter how other interferential factors vary. This is a particularly useful property for the multifactorial pattern recognition problems since filtering out the interferential factors is one of the main difficulties confronted in such problems.

For a particular missing value $\mathcal{D}_{i_1 i_2 \ldots i_N}$, its p-th contributory factor takes the i_p-th value, $p = 1, 2, \ldots, N$. Then $\mathcal{D}_{i_1 i_2 \ldots i_N}$ is reconstructed by

$$\hat{\mathcal{D}}_{i_1 i_2 \ldots i_N} = \hat{\mathcal{Z}} \times_1 (\boldsymbol{u}_1^{i_1})^T \times_2 (\boldsymbol{u}_2^{i_2})^T \cdots \times_N (\boldsymbol{u}_N^{i_N})^T, \tag{12}$$

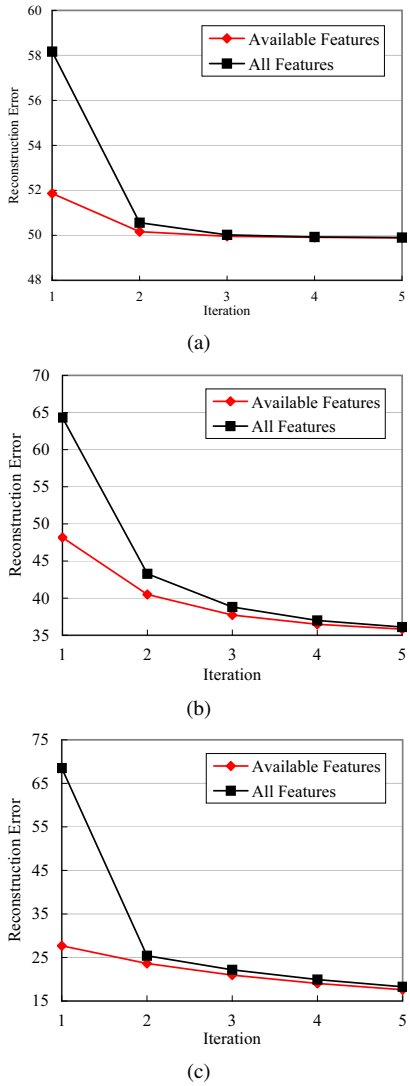

Fig. 4. The reconstruction errors of the available features and all the features in the first five iterations when training M^{2}SA on the CMU PIE database with (a) 10% missing data, (b) 50% missing data, and (c) 90% missing data.

where $(\boldsymbol{u}_p^{i_p})^T$ is the i_p-th row vector of the mode matrix $\hat{\mathbf{U}}_p$. Since $(\boldsymbol{u}_p^{i_p})^T$ is determined by all the samples with the p-th contributory factor equal to the i_p-th value, the missing value can be viewed as reconstructed by those available values sharing some factor values in the multilinear way. An illustration of this process is shown in Fig. 5, where the red cube marked by 'm' represents one missing value $\mathcal{D}_{i_1 i_2 i_3}$ in the third-order tensor $\mathcal{D}$ and the dark cubes represent the available values sharing at least one factor value with $\mathcal{D}_{i_1 i_2 i_3}$. The three factor values associated with $\mathcal{D}_{i_1 i_2 i_3}$ corresponds to three slices in the tensor, represented by the grey planes. Technically speaking, as long as there is one available value on these planes, $\mathcal{D}_{i_1 i_2 i_3}$ can be reconstructed. This means the algorithm can work even when the majority of values in the tensor are missing. However, too few available values will induce a poor approximation of the missing values. A relatively good approximation needs some available values on each of the three planes. According to the later experimental results, the algorithm can work well on a tensor with up to

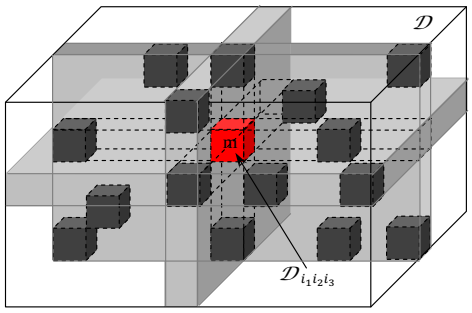

Fig. 5. Multilinear reconstruction of one missing value in the tensor.

80% missing values.

C. Classification of Test Samples

Given a previously unseen test sample, its feature vector $\boldsymbol{b}$ is first extracted. Suppose $\boldsymbol{b}$ can be represented by the mode-n product of a set of coefficient vectors:

$$\boldsymbol{b}^T = \mathcal{Z} \times_1 \boldsymbol{c}_1^T \times_2 \boldsymbol{c}_2^T \cdots \times_N \boldsymbol{c}_N^T, \tag{13}$$

where $\mathcal{Z}$ is the core tensor obtained by applying Algorithm 2 to the incomplete training tensor $\mathcal{D}$, and $\boldsymbol{c}_n (n = 1, 2, \ldots, N)$ is the coefficient vector corresponding to the value of factor n associated to the test sample. Note that both $\mathcal{D}$ and $\mathcal{Z}$ are of order $N+1$, with the $(N+1)$-th mode corresponding to the features. Then the response tensor $\mathcal{R}$ can be calculated by

$$\begin{aligned} \mathcal{R} &= \mathcal{Z}^{+(N+1)} \times_{(N+1)} \boldsymbol{b}^T \\ &= \boldsymbol{c}_1^T \circ \boldsymbol{c}_2^T \cdots \circ \boldsymbol{c}_N^T, \end{aligned} \tag{14}$$

where $\mathcal{Z}^{+(N+1)}$ is the *mode-(N+1) pseudo-inverse tensor* of $\mathcal{Z}$, which can be obtained by re-tensorizing the matrix $\mathbf{P} = \mathbf{Z}_{(N+1)}^{+T}$ (the transpose of the pseudo-inverse of the mode-$(N+1)$ flattened matrix of $\mathcal{Z}$. Please refer to [37] for more details). Thus $\mathcal{R}$ is the outer product of all factor coefficient vectors associated with $\boldsymbol{b}$, and therefore is of rank-$(1, \ldots, 1)$. Then, the N-mode dimensionality reduction algorithm (Algorithm 1) is used to find a best rank-$(1, \ldots, 1)$ approximation of $\mathcal{R}$, which leads to

$$\hat{\mathcal{R}} = \hat{\mathcal{T}} \times_1 \hat{\boldsymbol{c}}_1 \times_2 \hat{\boldsymbol{c}}_2 \cdots \times_N \hat{\boldsymbol{c}}_N, \tag{15}$$

where $\hat{\mathcal{T}}$ is the core tensor, and $\hat{\boldsymbol{c}}_n$ is the approximation of $\boldsymbol{c}_n$. Finally, the coefficient vector $\hat{\boldsymbol{c}}_t$ corresponding to the target factor f_t is compared with each row vector of $\mathbf{U}_t$ (recall that each row vector of $\mathbf{U}_t$ corresponds to one value of f_t), and the most similar row vector will then indicate the predicted f_t value for the test sample. In this paper, the similarity between the coefficient vectors is measured by the angle between them.

IV. Application to Face Recognition

A. Methodology

The data used for face recognition is the 'illum' subset of the the CMU PIE database [31]. In addition to the variation caused by different people, the face images are also greatly different in pose and illumination. The completely dark (without any illumination) images are removed, which leaves $18,564$ images from 68 individuals to be used in the

Fig. 6. Typical normalized face images from the CMU PIE database.

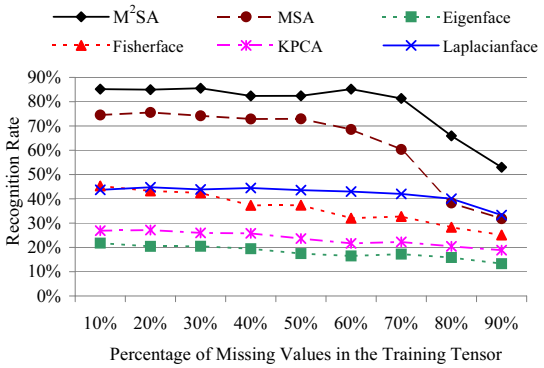

Fig. 7. Face recognition rate on test set (10-fold CV) with different percentage of missing values in the training tensor.

experiments. Each person's face exhibits 13 different poses and 21 different illumination conditions in the images. The faces are normalized by fixing the positions of the two eyes (for those profile faces, the positions of one eye and the nose tip are used). The normalized face image has 67×47 pixels. Some typical normalized faces are shown in Fig. 6.

The 10-fold cross validation is used to test the performance of the algorithms. In each fold, 10% of the $18,564$ face images are randomly selected as the test set, and the remaining are used as the training set. The final result is the average over the 10 folds. The training images are organized into a fourth-order tensor $\mathcal{D} \in \mathbb{R}^{68\times13\times21\times3149}$, where the four modes correspond to people's identity, pose, illumination, and image pixels, respectively. Removing the test images leaves 10% missing values in $\mathcal{D}$. A typical portion of $\mathcal{D}$ has been shown in Fig. 3. Note that the position of a particular value for any contributory factor on the corresponding mode (e.g., the position of a particular view angle on the 'Pose' mode) does not matter much since different positions only indicate different row indices of $(\boldsymbol{u}_t^k)^T$ in $\mathbf{U}_t$ in Eq. (11). What does matter is that all the samples with the same factor value must align to the same position on the corresponding mode. In order to test the capability of M^{2}SA in dealing with the missing values, the images in $\mathcal{D}$ are gradually reduced from 90% to only 10% of the total data set with the step 10%, while the test set remains always 10% of the total data set.

The compared baseline methods include the standard MSA (known as TensorFace [35] when applied to face recognition), two linear methods, i.e., Eigenface [34] and Fisherface [3], and two nonlinear methods, i.e., KPCA [27] and Laplacianface [15]. For M^{2}SA and MSA, if not specified, the rank R_n of the mode-n subspace is set to $2/3$ of that of the original space I_n. In order to apply the standard MSA, the missing values in the tensor are filled with the mean of available values sharing some contributory factors. For other baseline methods (Eigenface, Fisherface, KPCA and Laplacianface), they do not need to organize the training data in a tensor. Thus there is no missing value problem for them, but only a gradual reduction of the training samples. If not explicitly stated, the dimensionality of the subspace is set to explain 95% of the variance (in Eigenface, Laplacianface, and KPCA). Fisherface uses the same settings as in [3]. In Laplacianface, the adjacency graph is constructed by connecting the samples with the same class label. In KPCA, the Gaussian kernel with the standard deviation 1 is used.

B. Performance

The face recognition rates of the algorithms are compared in Fig. 7. M^{2}SA achieves the best performance in all cases. It keeps relatively steady at a high level above 80% while the missing values in the training tensor gradually increase from 10% to 70%. Only when the missing values account for 80% or higher of the training tensor does the performance of M^{2}SA start to notably deteriorate. The standard MSA performs generally better than either the linear methods (Eigenface and Fisherface) or the nonlinear ones (KPCA and Laplacianface). This reveals the superiority of MSA in decomposing the multiple contributory factors concealed in the data. But this superiority rapidly shrinks when the missing values become dominating. It is even surpassed by Laplacianface when there are more than 80% missing values in the training tensor. This is because that the standard MSA lacks the ability to effectively deal with the missing vales. Note that even when 90% of the tensor comprises missing values, the recognition rate of M^{2}SA is still significantly higher than that of the baseline methods.

As a 'byproduct', the missing images in the training tensor can be reconstructed by M^{2}SA. In order to verify this, all of the missing images in the training tensor, i.e., the test images, are reconstructed by M^{2}SA. The average correlation coefficient between the reconstructed images and the original images is 0.92. Fig. 8 shows some typical reconstructed missing face images in the training tensor compared with the original images, as well as the initial approximations of the images (i.e., the initial approximations extracted from $\hat{\mathcal{D}}^0$ in Algorithm 2). As can been seen, the final reconstructions of the face images remarkably improve the initial approximations and match the original images very well in all of the contributory factors including identity, pose and illumination. The effectiveness of imputing the missing values explains why M^{2}SA can achieve good results for face recognition with massive missing values in the training tensor.

Different subspace ranks of M^{2}SA are also tested. While the percentage of the missing values is maintained to be 10%, the ratio of the subspace rank over the original rank decreases from 100% (no dimensionality reduction) to 10% with step

Fig. 8. Typical reconstructed missing face images by M^{2}SA. The first row shows the original test images which are removed from the training tensor, the second row shows the initial approximations of the corresponding images, and the third row shows the final reconstructions of the images.

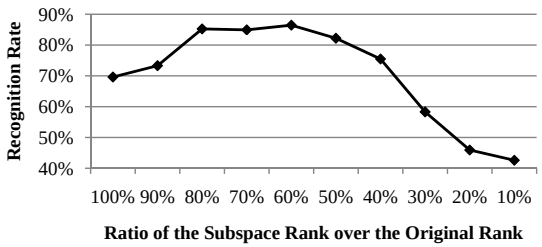

Fig. 9. Face recognition rate of M^{2}SA on test set (10-fold CV) with different ratios of the subspace rank over the original rank and 10% missing values.

10%. For convenience of analysis, this ratio for each mode is assumed to be the same. The recognition rates with different subspace ranks are shown in Fig. 9. It reveals that too high or too low subspace rank will both lead to poor performance. When the subspace rank is too high, say 100% of the original rank, the N-mode SVD is just a rotation of the original mode spaces. The reconstructed values equal the original values. Thus the missing values cannot be inferred from other available values. On the contrary, when the subspace rank is too low, too much information loss will also cause inaccurate approximation of the missing values. The algorithm reaches the best performance around the ratio $2/3$ used in the previous experiments. The performance curve for other missing value percentages might be different, but empirically, the ratio $2/3$ generally performs well.

V. Application to Facial Age Estimation

Automatic estimation of human facial age is an interesting yet challenging topic emerging in recent years. The relatively early work on exact age estimation was done by Lanitis et al. [20] [19], where the aging pattern was represented by a quadratic function called an *aging function*. Their most representative algorithms are the Weighted Appearance Specific (WAS) method [20] and the Appearance and Age Specific (AAS) method [19]. Later, Geng et al. [11] proposed the AGES algorithm based on the subspace trained on a data structure called aging pattern vector. They also suggested to use both the Mean Absolute Error (MAE) and the Cumulative Scores (CS) at different error levels to evaluate the age estimation methods. After that, various novel methods have been proposed for facial age estimation. For example, Fu et al. [6] [8] proposed an age estimation method based on multiple linear regression on the discriminative aging manifold of face images. Guo et al. [12] used the Support Vector Regression (SVR) method to design a locally adjusted robust regressor for the prediction of human ages. Yan et al. [43] regarded age estimation as a regression problem with uncertain nonnegative labels and solved the problem through semidefinite programming (SDP). They also proposed an Expectation-Maximization (EM) algorithm to solve the regression problem and speed up the optimization process [42]. By using spatially flexible patch (SFP) as the feature descriptor, the age regression was further improved with patch-based Gaussian Mixture Model (GMM) [45] and patch-based Hidden Markov Model (HMM) [48]. Guo et al. [13] also proposed to use the biologically inspired features (BIF) for human age estimation from faces. A very recent survey paper by Fu et al. [7] provides a comprehensive review of the existing techniques for facial age estimation.

Fig. 10. Typical aging faces in the FG-NET Aging Database. Each row shows the aging faces of one person.

Compared with other facial variations, such as expression, gender and identity, the collection of sufficient training data for age variation is extremely laborious. The available images usually only account for a small portion of the whole aging pattern. Moreover, the aging pattern is highly related to personal factors like genetics, health, lifestyle, etc. Thus facial age estimation is a typical multi-factor problem with a significant proportion of missing values.

A. Methodology

The FG-NET Aging Database [20] is used in the experiment. There are $1,002$ face images from 82 subjects in this database. Each subject has 6-18 face images at different ages. The ages are distributed in a wide range from 0 to 69. Besides age variation, most of the age-progressive image sequences display other types of facial variations, such as significant changes in pose, illumination, expression, *etc*. However, since there are no labels for these variations in the database, they are not embodied in the tensor. Some typical aging face sequences are shown in Fig. 10. For 82 people and 70 ages, a 'complete' training tensor requires $5,740$ images. Thus the missing values account for more than 82% $((5740-1002)/5740)$ of the training tensor. The aging face images are naturally organized in a third-order tensor $\mathcal{F} \in \mathbb{R}^{70\times 82\times 200}$: the first mode corresponds to the age, the second corresponds to the identity, and the last corresponds to the features extracted from the face images. The feature extractor used here is the Appearance

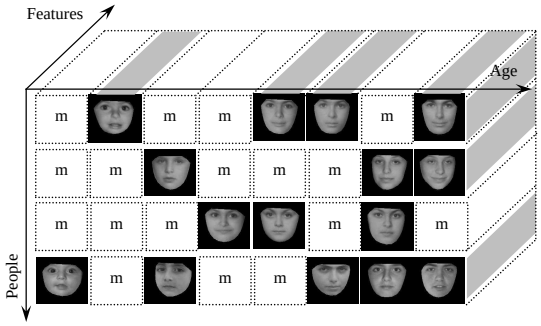

Fig. 11. Tensor representation of a subset from the FG-NET Aging Database. The missing parts are labeled by 'm'.

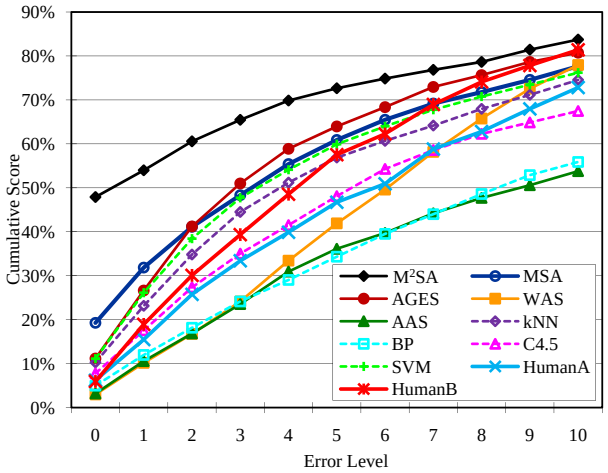

Fig. 12. Cumulative scores of the algorithms and the human tests at the error levels 0-10.

Model [5]. The main advantage of this model is that the extracted features combine the shape and the intensity of the face images, which are both important in the aging progress. The extracted features require 200 model parameters to retain about 95% of the variability in the training data. A typical portion of $\mathcal{F}$ is shown in Fig. 11.

M^{2}SA is compared with the standard MSA (replacing the missing values by means), AGES [11], WAS [20], AAS [19], as well as some conventional classification methods including k-Nearest Neighbors (kNN), Back Propagation neural network (BP), C4.5 decision tree (C4.5), and Support Vector Machines (SVM). The algorithms are tested through the Leave-One-Person-Out (LOPO) mode, *i.e.*, in each fold, the images of one person are used as the test set and those of the others are used as the training set. After 82 folds, each subject has been used as test set once, and the final results are calculated from all the estimates.

For M^{2}SA and MSA, the parameters are set to the same values as in the face recognition experiments. For all other algorithms, several parameter configurations are tested and the best result is reported. For AGES, the aging pattern subspace dimensionality is set to 20. In AAS, the error threshold in the appearance cluster training step is set to 3, and the age ranges for the age specific classification are set as 0-9, 10-19, 20-39 and 40-69. The k in kNN is set to 30. The BP neural network has a single hidden layer of 100 neurons. The parameters of C4.5 are set to the default values of the J4.8 implementation. SVM uses the RBF kernel with the inverse width of 1.

As an important baseline, the human ability in age perception is also tested. About 5% of the database (51 face images) are randomly selected and presented to 29 human observers. There are two stages in the experiment. In each stage, the images are randomly shown to the observers, and the observers are asked to choose an age from 0 to 69 for each image. The difference is that in the first stage (denoted by HumanA), only the gray-scale face regions are shown, while in the second stage (denoted by HumanB), the whole color images are shown. HumanA intends to test age estimation ability purely based on the pixel intensity within the face region, which is also the input to the algorithms, while HumanB intends to test the human estimation ability based on multiple traits including face, hair, skin color, clothes, etc.

B. Performance

First, the performance of age estimation is evaluated by the Mean Absolute Error (MAE), i.e., the average absolute difference between the estimated age and the real age. The results are tabulated in Table I. The algorithms performing better than HumanA are highlighted by boldface and those better than HumanB are underlined. As can be seen, M^{2}SA is significantly better than all the compared algorithms. It is worth mentioning that there are other existing techniques which have reported lower MAE than the 5.36 of M^{2}SA [7]. For example, by using the biologically inspired features (BIF), Guo et al. [13] achieved the lowest ever MAE 4.77 on the FG-NET database. However, the focus of this paper is not to get the lowest MAE, but to solve the missing value problem in MSA. Without the mechanism to deal with missing values, MSA gives a poor result. Including M^{2}SA, there are four algorithms (M^{2}SA, AGES, WAS, and SVM) obtaining lower MAE than that of HumanA. It is interesting to note that the MAE of M^{2}SA is even lower than that of HumanB, where the human observers are provided with more clues for age estimation than that input into the algorithms.

In addition to MAE, the performance is also evaluated by the *cumulative scores* at different error levels. Suppose there are M test images, $M_{e\leq l}$ is the number of test images on which the age estimator makes an absolute error no higher than l (years), then the cumulative score at error level l is calculated by $CumScore(l) = M_{e\leq l}/M \times 100\%$. The cumulative scores of the age estimators at the error levels from 0 to 10 are compared in Fig. 12. As can be seen, the cumulative scores of M^{2}SA at all error levels are remarkably higher than those of the compared algorithms as well as the human tests. The advantage is more significant at the relatively more important low error levels. Combined with the results in Table I, we can conclude that, at least under this experimental setting, M^{2}SA outperforms not only all the compared algorithms, but also the human observers in the ability of facial age estimation.

TABLE I
MEAN ABSOLUTE ERROR (IN YEARS) OF AGE ESTIMATION ON THE FG-NET AGING DATABASE[1]

Method	M^2SA	MSA	AGES	WAS	AAS	kNN	BP	C4.5	SVM	HumanA	HumanB
MAE	**5.36**	9.88	**6.77**	**8.06**	14.83	8.24	11.85	9.34	**7.25**	8.13	6.23

[1] The lowest ever MAE based on the biologically inspired features (BIF) [13]: 4.77.

VI. CONCLUSION

Multilinear Subspace Analysis (MSA) methods possess inherent advantages over linear subspace methods (e.g., PCA) in explicitly decomposing the interlaced semantic factors for the multifactorial problems. The basis of MSA is a well organized training tensor with each mode corresponding to one contributory factor. This requires a large training set with at least one sample at each possible combination of all the factors. Unfortunately, such a 'complete' training set is usually unavailable in many real applications. This paper extends our preliminary study [10] [9], which proposes the M^2SA algorithm to enable MSA to work on tensors with a large percentage of missing values. Experiments on face recognition and facial age estimation reveal that M^2SA can perform stably with missing values accounting for up to 80% of the total data. This greatly improves the applicability of the MSA methods.

M^2SA relies on iteratively decomposing and reconstructing the training tensor. Nevertheless, analogous to the methods proposed for PCA with missing values [39] [30] [32], other ways of dealing with missing values are also worth investigating in the future, such as introducing prior distribution assumptions to improve the performance or speedup the convergence of the algorithm.

Essentially, M^2SA is based on weighted least-squares estimation, and hence is not so robust against outliers that often exist in realistic training sets, e.g., due to occlusion, illumination, image noise, or errors from the underlying data generation method [18]. There are many approaches toward robust linear subspace learning [46] [17] [18]. Incorporating these robust learning methods into M^2SA is another important future direction.

Apart from MSA, there are other multiway analysis methods (e.g., [24], [25], [28], [29] [35], [38]) that organize all training samples in a single tensor with each mode corresponding to one factor contributing to the problem. Handling of missing values in the tensor is a common problem in this kind of approaches. Thus developing extensions for these methods to deal with the missing values will become a series of important future studies.

ACKNOWLEDGMENT

This work is supported by the Australian Research Council Discovery Grant (DP0987421), the National Science Foundation of China (60905031, 6107309), the Jiangsu Science Foundation (BK2009269, BK2008018), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, the National Fundamental Research Program of China (2010CB327903), and the Jiangsu 333 High-Level Talent Cultivation Program.

REFERENCES

[1] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations with missing data," in *Proc. SIAM Int'l Conf. Data Mining*, Columbus, OH, 2010, pp. 701–712.

[2] E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *IEEE Trans. Knowledge and Data Engineering*, in press 2008.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

[4] J. D. C. J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of 'eckart-young' decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[5] G. J. Edwards, A. Lanitis, and C. J. Cootes, "Statistical face models: Improving specificity," *Image Vision Comput.*, vol. 16, no. 3, pp. 203–211, 1998.

[6] Y. Fu, Y. Xu, and T. S. Huang, "Estimating human age by manifold analysis of face pictures and regression on aging features," in *Proc. IEEE Int'l Conf. Multimedia and Expo*, Beijing, China, 2007, pp. 1383–1386.

[7] Y. Fu, G. Guo, and T. S. Huang, "Age synthesis and estimation via faces: A survey," *IEEE Trans. Pattern Anal. Machine Intell.*, In press, 2010.

[8] Y. Fu and T. Huang, "Human age estimation with regression on discriminative aging manifold," *IEEE Trans. Multimedia*, vol. 10, no. 4, pp. 578–584, 2008.

[9] X. Geng and K. Smith-Miles, "Facial age estimation by multilinear subspace analysis," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, 2009, pp. 865–868.

[10] X. Geng, K. Smith-Miles, Z.-H. Zhou, and L. Wang, "Face image modeling by multilinear subspace analysis with missing values," in *Proc. ACM Int'l Conf. Multimedia*, Beijing, China, 2009, pp. 629–632.

[11] X. Geng, Z.-H. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2234–2240, 2007.

[12] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression," *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1178–1188, 2008.

[13] G. Guo, G. Mu, Y. Fu, and T. S. Huang, "Human age estimation using bio-inspired features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 112–119.

[14] R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an 'explanatory' multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[15] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, 2005.

[16] H. Huang and C. H. Q. Ding, "Robust tensor factorization using r1 norm," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1–8.

[17] I. T. Jolliffe, *Principal Component Analysis, 2nd Edition*. New York: Springer-Verlag, 2002.

[18] F. D. la Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 117–142, 2003.

[19] A. Lanitis, C. Draganova, and C. Christodoulou, "Comparing different classifiers for automatic age estimation," *IEEE Trans. Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 34, no. 1, pp. 621–628, 2004.

[20] A. Lanitis, C. J. Taylor, and T. Cootes, "Toward automatic simulation of aging effects on face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 442–455, 2002.

[21] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[22] ——, "On the best rank-1 and rank-(r1,r2,. . .,rn) approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[23] Z. Lei, Q. Bai, R. He, and S. Z. Li, "Face shape recovery from a single image using cca mapping between tensor spaces," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1–7.

[24] Y. Li, Y. Du, and X. Lin, "Kernel-based multifactor analysis for image synthesis and recognition," in *Proc. IEEE Int'l Conf. Computer Vision*, Beijing, China, 2005, pp. 114–119.

[25] D. Lin, Y. Xu, X. Tang, and S. Yan, "Tensor-based factor decomposition for relighting," in *Proc. IEEE Conf. Image Processing*, Genoa, Italy, 2005, pp. 386–389.

[26] J. Liu, P. Musialski, P. Wonka, and J. Ye., "Tensor completion for estimating missing values in visual data," in *Proc. 12th IEEE Int'l Conf. Computer Vision*, Kyoto, Japan, 2009, pp. 2114–2121.

[27] B. Moghaddam, "Principal manifolds and probabilistic subspaces for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 780–788, 2002.

[28] S. W. Park and M. Savvides, "Individual kernel tensor-subspaces for robust face recognition: A computationally efficient tensor framework without requiring mode factorization," *IEEE Trans. Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 37, no. 5, pp. 1156–1166, 2007.

[29] S. Rana, W. Liu, M. M. Lazarescu, and S. Venkatesh, "Recognising faces in unseen modes: A tensor based approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1–8.

[30] S. T. Roweis, "EM algorithms for PCA and SPCA," in *Advances in Neural Information Processing Systems 10*, Denver, CO, 1997, pp. 626–632.

[31] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, 2003.

[32] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, vol. 61, pp. 611–622, 1999.

[33] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[34] M. A. Turk and A. Pentland, "Eigenface for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[35] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. Euro. Conf. Computer Vision*, Copenhagen, Denmark, 2002, pp. 447–460.

[36] ——, "Multilinear subspace analysis of image ensembles," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Madison, WI, 2003, pp. 93–99.

[37] ——, "Multilinear projection for appearance-based recognition in the tensor framework," in *Proc. Int'l Conf. Computer Vision*, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[38] H. Wang and N. Ahuja, "Rank-R approximation of tensors using image-as-matrix representation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 346–353.

[39] T. Wiberg, "Computation of principal component when data are missing," in *Proc. the 2nd Symp. on Computational Statistics*, Berlin, Germany, 1976, pp. 229–236.

[40] D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, and T. S. Huang, "Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 18, no. 1, pp. 36–47, 2008.

[41] D. Xu, S. Yan, L. Zhang, H. Zhang, Z. Liu, and H.-Y. Shum, "Concurrent subspaces analysis," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 203–208.

[42] S. Yan, H. Wang, T. S. Huang, Q. Yang, and X. Tang, "Ranking with uncertain labels," in *Proc. IEEE Int'l Conf. Multimedia and Expo*, Beijing, China, 2007, pp. 96–99.

[43] S. Yan, H. Wang, X. Tang, and T. S. Huang, "Learning auto-structured regressor from uncertain nonnegative labels," in *Proc. IEEE Int'l Conf. Computer Vision*, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[44] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Processing*, vol. 16, no. 1, pp. 212–220, 2007.

[45] S. Yan, X. Zhou, M. Liu, M. Hasegawa-Johnson, and T. S. Huang, "Regression from patch-kernel," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[46] T.-N. Yang and S.-D. Wang, "Robust algorithms for principal component analysis," *Pattern Recognition Letters*, vol. 20, no. 9, pp. 927–933, 1999.

[47] L. Zhang, Q. Gao, and D. Zhang, "Directional independent component analysis with tensor representation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1–7.

[48] X. Zhuang, X. Zhou, M. Hasegawa-Johnson, and T. S. Huang, "Face age estimation using patch-based hidden markov model supervectors," in *Int'l Conf. Pattern Recognition*, Tampa, FL, 2008, pp. 1–4.