INFORMS Journal on Data Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Supervised Tensor Dimension Reduction-Based Prognostic Model for Applications with Incomplete Imaging Data

Chengyu Zhou, Xiaolei Fang

To cite this article:
Chengyu Zhou, Xiaolei Fang (2024) A Supervised Tensor Dimension Reduction-Based Prognostic Model for Applications with Incomplete Imaging Data. INFORMS Journal on Data Science 3(1):84-104. https://doi.org/10.1287/ijds.2022.x022

https://pubsonline.informs.org/journal/ijds

INFORMS JOURNAL ON DATA SCIENCE
Vol. 3, No. 1, April–June 2024, pp. 84–104
ISSN 2694-4022 (print), ISSN 2694-4030 (online)

A Supervised Tensor Dimension Reduction-Based Prognostic Model for Applications with Incomplete Imaging Data

Chengyu Zhou,[a] Xiaolei Fang[a,*]

[a] Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, North Carolina 27606
*Corresponding author
Contact: czhou9@ncsu.edu (CZ); xfang8@ncsu.edu, https://orcid.org/0000-0002-0215-0403 (XF)

Received: July 29, 2022
Revised: March 13, 2023
Accepted: June 2, 2023
Published Online in Articles in Advance: July 17, 2023

https://doi.org/10.1287/ijds.2022.x022

Abstract. Imaging data-based prognostic models focus on using an asset's degradation images to predict its time to failure (TTF). Most image-based prognostic models have two common limitations. First, they require degradation images to be complete (i.e., images are observed continuously and regularly over time). Second, they usually employ an unsupervised dimension reduction method to extract low-dimensional features and then use the features for TTF prediction. Because unsupervised dimension reduction is conducted on the degradation images without the involvement of TTFs, there is no guarantee that the extracted features are effective for failure time prediction. To address these challenges, this article develops a supervised tensor dimension reduction-based prognostic model. The model first proposes a supervised dimension reduction method for tensor data. It uses historical TTFs to guide the detection of a tensor subspace to extract low-dimensional features from high-dimensional incomplete degradation imaging data. Next, the extracted features are used to construct a prognostic model based on (log)-location-scale regression. An optimization algorithm for parameter estimation is proposed, and analytical solutions are discussed. Simulated data and a real-world data set are used to validate the performance of the proposed model.

History: Bianca Maria Colosimo served as the senior editor for this article
Funding: This work was supported by National Science Foundation [2229245].
Data Ethics & Reproducibility Note: The code capsule is available on Code Ocean at https://github.com/czhou9/Code-and-Data-for-IJDS and in the e-Companion to this article (available at https://doi.org/10.1287/ijds.2022.x022).

Keywords: supervised dimension reduction • missing data • remaining useful life • residual useful life

1. Introduction

Degradation is an irreversible process of damage accumulation that results in the failure of engineering systems/assets/components (Bogdanoff and Kozin 1985). Although it is usually challenging to observe a physical degradation process, there often are some manifestations associated with degradation processes that can be monitored by sensing technology, which yields data known as degradation data/signals. Degradation data contains the health condition of engineering assets; thus, if modeled properly, they can be used to predict the assets' time to failure (TTF) via a process known as prognostic. Many prognostic models have been developed in the literature, most of which focus on using time series-based degradation data (Gebraeel et al. 2005; Hong and Meeker 2010, 2013; Liu et al. 2013; Ye et al. 2014; Ye and Chen 2014, Shu et al. 2015; Wang et al. 2022). Recently, prognostic models with imaging-based degradation data have been investigated and have attracted more and more attention. This is because, compared with time-series data, imaging data usually contain much richer information of the object being monitored, and imaging-sensing technologies are noncontact, and thus they can usually be easily deployed. One example of imaging-based degradation data is the infrared image stream that measures the change of temperature distribution of a thrust bearing during its degradation process over time (Aydemir and Paynabar 2019, Fang et al. 2019, Dong et al. 2021, Wang et al. 2021, Jiang et al. 2022a). Another example is the images used to measure the performance degradation of infrared systems such as rotary-wing drones (Dong et al. 2021).

The existing imaging-based prognostic methods include deep-learning-based models and statistical learning methods. Examples of the deep-learning-based models designed for TTF prediction using imaging data include the ones developed by Aydemir and Paynabar (2019), Dong et al. (2021), Yang et al. (2021), Jiang et al. (2022a), Jiang et al. (2022b), and Jiang et al. (2023). Although these models have worked relatively well, they usually provide point estimations of failure times, and it is challenging for them to quantify the uncertainty of predicted TTFs (e.g., providing a failure time distribution). This limits their applicability because the subsequent decision-making analysis such as maintenance/inventory/logistic optimization requires prognostic models to provide a distribution of

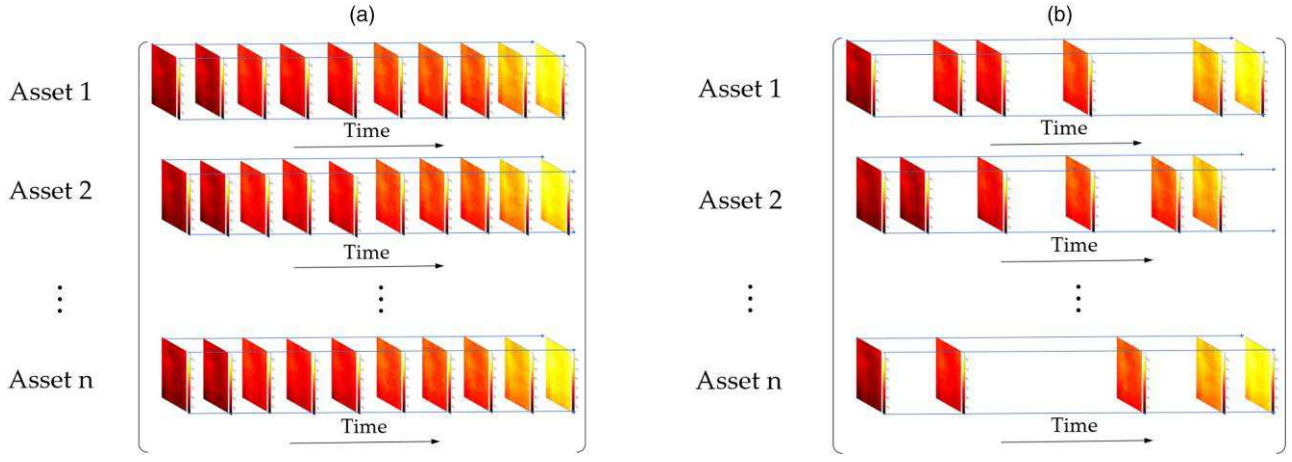
the predicted TTF. Also, deep-learning-based prognostic models often require a relatively large number of historical samples for model training, which cannot be satisfied by many real-world applications with limited historical data. One example of statistical learning methods for image-based prognostic is the penalized (log)-location-scale (LLS) tensor regression proposed by Fang et al. (2019). The model first employs multilinear principal component analysis (MPCA) (Lu et al. 2008) to reduce the dimension of high-dimensional imaging-based degradation data, which yields a low-dimensional feature tensor for each asset. Next, it constructs a prognostic model by regressing an asset's TTF against its low-dimensional feature tensor using LLS regression. In the same article, Fang et al. (2019) also proposed several benchmarking prognostic models that used imaging-based degradation data for TTF prediction. These models also first employed a dimension reduction method, such as functional principal component analysis (FPCA) (Ramsay and Silverman 2005), principal component analysis (PCA) (Abdi and Williams 2010), or B-Spline (Prautzsch et al. 2002), to reduce the dimension of degradation data and then used low-dimensional features to build an LLS regression model for prognostic. Although the aforementioned statistical learning-based prognostic models can provide a distribution for the predicted TTF and their effectiveness has been well investigated, they share two common limitations.

The first limitation is that they assume that imaging-based degradation data (including historical data for model training and real-time data for model test) are complete, which means that images from all the assets should be collected continuously and regularly with the same sampling time interval (see Figure 1(a) for an example). In reality, however, engineering assets often operate in harsh environments that significantly impact the quality of collected data because of errors in data acquisition, communication, read/write operations, etc. As a result, degradation images often contain significant levels of missing observations, which is known as incomplete/missing imaging data (see Figure 1(b) for an example). Such data incompleteness poses a significant challenge for the parameter estimation of existing statistical learning-based prognostic models.

The second common limitation for the existing statistical learning-based prognostic models for applications with imaging data is that they employ unsupervised dimension reduction methods for feature extraction, so there is no guarantee that the extracted features are effective for the subsequent TTF prediction. Specifically, they first use unsupervised dimension reduction methods such as FPCA, PCA, and B Spline to extract features, which are then used to construct prognostic models. Because feature extraction and prognostic model construction are two sequential steps, and no TTF information gets involved in the feature extraction process, it is possible that the extracted features may not be the most suitable for predicting TTFs.

To address the aforementioned challenges, this article proposes a supervised dimension reduction-based prognostic model that uses an asset's incomplete degradation images to predict its TTF. Similar to the existing statistical learning-based prognostic models, the proposed model also consists of two steps: feature extraction and prognostic model construction. However, unlike the existing models, feature extraction in this article is achieved by developing a new supervised tensor dimension reduction method, which uses historical TTFs to supervise the feature extraction process such that the extracted features are more effective for the subsequent TTF prediction. In addition, unlike the existing unsupervised dimension reduction methods that only work for complete imaging data,

Figure 1. (Color online) Degradation Stream Images With and Without Missing Data



Note. (a) Complete image streams; (b) incomplete image streams.

the proposed supervised dimension reduction method works for both complete and incomplete degradation image streams.

The proposed supervised dimension reduction method works as follows. First, it detects a low-dimensional tensor subspace in which the high-dimensional degradation image streams are embedded. This is achieved by constructing an optimization criterion that comprises a feature extraction term and a regression term. The first term extracts low-dimensional features from complete/incomplete degradation image streams of training assets, and the second term builds the connection between these assets' TTFs and the extracted features using LLS regression. LLS regression has been widely used in reliability engineering and survival analysis. It includes a variety of TTF distributions, such as (log)normal, (log)logistic, smallest extreme value, and Weibull, etc., which cover most of the TTF distributions in reality (Doray 1994). Because historical TTFs are used to supervise the feature extraction process, it is expected that the extracted features are more effective for the subsequent prognostic. Solving the optimization criterion of the proposed supervised tensor dimension reduction method yields a set of tensor basis matrices that span the low-dimensional tensor subspace for dimension reduction. We then expand both the historical degradation images in the training data set and real-time degradation images from an asset operating in the field (i.e., test data) using the set of tensor basis matrices to extract the low-dimensional tensor features of the training assets and the test asset. The TTFs of the training assets are then regressed against their tensor features using LLS regression, and the parameters are estimated using maximum likelihood estimation. After that, the tensor features of the test asset are fed into the LLS regression model, and its TTF distribution is predicted.

To solve the optimization criterion of the proposed supervised dimension reduction method, we will first transfer the criterion into a block multiconvex problem. Next, we will propose a block updating algorithm, which cyclically optimizes one-block parameters while keeping other blocks fixed until convergence. In addition, we will demonstrate that when TTFs follow normal or lognormal distributions, each subproblem of the block updating algorithm has a closed-form solution, no matter whether the degradation image streams are complete or incomplete.

The rest of this paper is organized as follows. Section 2 presents the supervised tensor dimension reduction-based prognostic method. Section 3 introduces the block updating algorithm and closed-form solutions when TTFs follow normal/lognormal distributions. Sections 5 and 6 validate the effectiveness of the proposed prognostic model using a simulated data set and data from a rotating machinery, respectively. Section 7 concludes the paper.

2. The Methodology

In this section, we will introduce the proposed supervised tensor dimension reduction-based prognostic model for applications with incomplete imaging data. In Section 2.1, we will present some basic tensor notations and definitions. Section 2.2 introduces the supervised tensor dimension reduction method. In Section 2.3, we will discuss the construction of a prognostic model and how to predict the TTF of an asset operating in the field using its real-time degradation imaging data.

2.1. Preliminaries

In this section, we introduce some basic notations and definitions of tensor operations that are used throughout the article. The *order* of a tensor is the number of dimensions, also known as ways or modes. Vectors (1-order tensors) are denoted by lowercase boldface letters; for example, $\boldsymbol{s}$. Matrices (2-order tensors) are denoted by boldface uppercase letters, for example, $\boldsymbol{S}$. Higher-order tensors (order is 3 or larger) are denoted by calligraphic letters, for example, $\mathcal{S}$. Indices are denoted by lowercase letters whose range is from 1 to the uppercase letter of the index, for example, $n = 1, 2, \ldots, N$. An Nth-order tensor is denoted as $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where I_n represents the nth mode of $\mathcal{S}$. The $(i_1, i_2, \ldots, i_N)$th entry of $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $s_{i_1, i_2, \ldots, i_n}$. A fiber of $\mathcal{S}$ is a vector defined by fixing every index but one. A matrix column is a mode-1 fiber, and a matrix row is a mode-2 fiber. The *vectorization* of $\mathcal{S}$, denoted by $vec(\mathcal{S})$, stacks all the entries of $\mathcal{S}$ into a column vector. The mode-n *matricization* of a tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\boldsymbol{S}_{(n)}$, which arranges the mode-n fibers to be the columns of the resulting matrix. The nth mode product of a tensor $\mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $\boldsymbol{U}_n \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{S} \times_n \boldsymbol{U}_n$, is a tensor whose entry is $(\mathcal{S} \times_n \boldsymbol{U}_n)_{i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N} = \sum_{i_n=1}^{I_N} s_{i_1, \ldots, i_N} u_{j, i_n}$. The *Kronecker product* of two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p \times q}$ is an $mp \times nq$ block matrix:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \ldots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \ldots & a_{mn}\boldsymbol{B} \end{bmatrix}.$$

If $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same number of columns $n = q$, then the *Khatri-Rao* product is defined as the $mp \times n$ column-wise *Kronecker* product: $\boldsymbol{A} \odot \boldsymbol{B} = [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \quad \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \cdots \boldsymbol{a}_n \otimes \boldsymbol{b}_n]$. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors, then $\boldsymbol{A} \otimes \boldsymbol{B} = \boldsymbol{A} \odot \boldsymbol{B}$. More details about tensor notations and operators can be found in Kolda and Bader (2009).

2.2. The Supervised Tensor Dimension Reduction Method

We assume that there exists a historical data set for model training. The data set consists of the degradation image streams of M failed assets along with their TTFs, which are denoted as $\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and $y_m \in \mathbb{R}$, respectively,

where $m = 1, 2, \ldots, M$. For the convenience of introducing the dimension reduction method, we convert the 3D degradation image streams from all the M assets to a 4D tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times M}$, where the sample size M is the fourth mode. Similarly, we let $\boldsymbol{y} = (y_1, \ldots, y_M)^\top \in \mathbb{R}^{M \times 1}$ be the vector containing all of the TTFs of the M assets.

Out of the $I_1 \times I_2 \times I_3 \times M$ entries of $\mathcal{X}$, we use a subset $\Omega \subseteq \{(i_1, i_2, i_3, m), 1 \le i_1 \le I_1, 1 \le i_2 \le I_2, 1 \le i_3 \le I_3, 1 \le m \le M\}$ to denote the indices of the missing ones. To model the missing data, we define a projection operator $\mathcal{P}_\Omega(\cdot)$ as

$$\mathcal{P}_\Omega(\mathcal{X})_{(i_1,i_2,i_3,m)} = \begin{cases} \mathcal{X}_{(i_1,i_2,i_3,m)}, & \text{if } (i_1, i_2, i_3, m) \notin \Omega, \\ 0, & \text{if } (i_1, i_2, i_3, m) \in \Omega, \end{cases} \tag{1}$$

where $\mathcal{X}_{(i_1,i_2,i_3,m)}$ is the (i_1, i_2, i_3, m)-th entry of the 4D tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times M}$. To recover the missing entries in tensor $\mathcal{X}$, we may employ the following Tucker decomposition-based tensor completion method (Liu et al. 2012, Xu et al. 2013, Filipović and Jukić 2015):

$$\min_{\mathcal{S}, \boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3} \|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2. \tag{2}$$

where $\|\cdot\|_F^2$ is the Frobenius norm, $\boldsymbol{U}_1 \in \mathbb{R}^{P_1 \times I_1}$, $\boldsymbol{U}_2 \in \mathbb{R}^{P_2 \times I_2}$, $\boldsymbol{U}_3 \in \mathbb{R}^{P_3 \times I_3}$ are three factor matrices, $\mathcal{S} \in \mathbb{R}^{P_1 \times P_2 \times P_3 \times M}$ is the low-dimensional core tensor, and $\times_n$ is the n-mode product of a tensor with a matrix. The tensor completion criterion (2) can be seen as an *unsupervised dimension reduction method* for tensor data with missing entries. This is because the degradation image tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times M}$ is a 4-order tensor, which resides in the tensor (multilinear) space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \mathbb{R}^{I_3} \otimes \mathbb{R}^{M}$, where $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}, \mathbb{R}^{I_3}, \mathbb{R}^{M}$ are the 4 vector (linear) spaces; $\mathcal{S} \in \mathbb{R}^{P_1 \times P_2 \times P_3 \times M}$ can be seen as a feature tensor that resides in the tensor space $\mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2} \otimes \mathbb{R}^{P_3} \otimes \mathbb{R}^{M}$. Usually, we have $P_1 \ll I_1$, $P_2 \ll I_2$, and $P_3 \ll I_3$ for degradation imaging data because of the high spatio-temporal correlation among pixels. This implies that the dimension of the image stream from the mth asset is reduced from $\mathbb{R}^{I_1 \times I_2 \times I_3}$ to $\mathbb{R}^{P_1 \times P_2 \times P_3}$, where $m = 1, \ldots, M$.

Although criterion (2) can be seen as a dimension reduction method, there is no guarantee that the extracted low-dimensional feature tensor $\mathcal{S}$ is effective for the subsequent TTF prediction. To address this challenge, we propose the following *supervised dimension reduction* method by combining a tensor completion term and an LLS regression term,

$$\min_{\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3, \sigma, \boldsymbol{\beta}_1, \beta_0, \mathcal{S}} \alpha \|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\ell\left(\frac{\boldsymbol{y} - \mathbf{1}_M \beta_0 - \boldsymbol{S}_{(4)}\boldsymbol{\beta}_1}{\sigma}\right), \tag{3}$$

where $\boldsymbol{y} \in \mathbb{R}^{M \times 1}$ is the vector containing all the TTFs of the M assets in the training data set. The matrix $\boldsymbol{S}_{(4)} \in \mathbb{R}^{M \times (P_1 \times P_2 \times P_3)}$ is the mode-4 matricization of the low-dimensional feature tensor $\mathcal{S}$, the mth row of which represents the vectorization of the mth asset's feature tensor, $m = 1, \ldots, M$. β_0 is the intercept, and $\boldsymbol{\beta_1} \in \mathbb{R}^{(P_1 \times P_2 \times P_3) \times 1}$ is the regression coefficient vector. $\mathbf{1}_m \in \mathbb{R}^{M \times 1}$ is an $M \times 1$ vector whose entries are all ones. $\ell(\cdot)$ is the negative log-likelihood function of a location-scale distribution. For example, if TTFs follow normal distributions, then $\ell\left(\frac{\boldsymbol{y} - \mathbf{1}_M\beta_0 - \boldsymbol{S}_{(4)}\boldsymbol{\beta}_1}{\sigma}\right) = \frac{M}{2}\log 2\pi + M\log\sigma + \frac{1}{2}\Sigma_{m=1}^{M}\omega_m^2$, where $\omega_m = \frac{y_m - \beta_0 - \boldsymbol{s}_{(4)}^m\boldsymbol{\beta}_1}{\sigma}$, $\boldsymbol{s}_{(4)}^m$ is the mth row of $\boldsymbol{S}_{(4)}$, and y_m is the TTF of asset m; if TTFs follow logistic distributions, then $\ell\left(\frac{\boldsymbol{y} - \mathbf{1}_M\beta_0 - \boldsymbol{S}_{(4)}\boldsymbol{\beta}_1}{\sigma}\right) = M\log\sigma - \Sigma_{m=1}^{M}\omega_m + 2\Sigma_{m=1}^{M}\log(1 + \exp(\omega_m))$, and if TTFs follow small extreme value (SEV) distributions, then $\ell\left(\frac{\boldsymbol{y} - \mathbf{1}_M\beta_0 - \boldsymbol{S}_{(4)}\boldsymbol{\beta}_1}{\sigma}\right) = n\log\sigma - \Sigma_{m=1}^{M}\omega_m + \Sigma_{m=1}^{M}\exp(\omega_m)$. For assets whose TTFs follow log-location-scale distributions, we may transfer them to the corresponding location-scale distributions by taking their logarithm such that criterion (3) can still be used. For example, log-normal, log-logistics, and Weibull distributions can be transferred to normal, logistics, and SEV distributions, respectively. $\alpha \in [0, 1]$ is a weight, and $\|\cdot\|_F^2$ is the Frobenius norm.

In criterion (3), the first term $\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2$ is tensor completion from (2), which reduces the dimension of high-dimensional incomplete degradation image streams and extracts low-dimensional tensor features. The second term $\ell\left(\frac{\boldsymbol{y} - \mathbf{1}_M\beta_0 - \boldsymbol{S}_{(4)}\boldsymbol{\beta}_1}{\sigma}\right)$ is LLS regression, which regresses each asset's TTF against its tensor features extracted by the first term. By jointly optimizing the two terms, it is expected that the extracted features are effective for TTF prediction. However, it is challenging to solve criterion (3) because it is neither convex nor block multiconvex. An optimization problem is block multiconvex when its feasible set and objective function are generally nonconvex but convex in each block of variables (Xu and Yin 2013). Thus, to simplify the development of optimization algorithms for model parameter estimation, we first transform criterion (3) to a block multiconvex one. Specifically, we apply the following reparameterization: $\tilde{\sigma} = 1/\sigma, \tilde{\beta}_0 = \beta_0/\sigma, \tilde{\boldsymbol{\beta}}_1 = \boldsymbol{\beta}_1/\sigma$. As a result, criterion (3) can be re-expressed as

$$\min_{\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3, \tilde{\sigma}, \tilde{\boldsymbol{\beta}}_1, \tilde{\beta}_0, \boldsymbol{S}_{(4)}} \alpha \|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_M\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\tilde{\boldsymbol{\beta}}_1), \tag{4}$$

where $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_M\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\tilde{\boldsymbol{\beta}}_1) = \frac{M}{2}\log 2\pi - M\log\tilde{\sigma} + \frac{1}{2}\Sigma_{m=1}^{M}\tilde{\omega}_m^2$ for TTFs following normal distributions, and $\tilde{\omega}_m = \tilde{\sigma}y_m - \tilde{\beta}_0 - \boldsymbol{s}_{(4)}^m\tilde{\boldsymbol{\beta}}_1$, where $\boldsymbol{s}_{(4)}^m$ is the mth row of $\boldsymbol{S}_{(4)}$ and y_m is the TTF of asset m; $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_M\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\tilde{\boldsymbol{\beta}}_1) =$

$-M\log\tilde{\sigma} - \Sigma_{m=1}^{M}\tilde{\omega}_m + 2\Sigma_{m=1}^{M}\log(1+\exp(\tilde{\omega}_m))$ for TTFs following logistics distributions, and $\ell(\tilde{\sigma}\boldsymbol{y} - \mathbf{1}_M\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\tilde{\boldsymbol{\beta}}_1) = -n\log\tilde{\sigma} - \Sigma_{m=1}^{M}\tilde{\omega}_m + \Sigma_{m=1}^{M}\exp(\tilde{\omega}_m)$ for TTFs following SEV distributions.

The optimization algorithm to solve criterion (4), the value of the weight α, and the dimension of the low-dimensional tensor subspace $\{P_1, P_2, P_3\}$ will be discussed in Sections 3 and 4. Solving the optimization criterion (4) using historical training data yields a set of basis matrices $\hat{\boldsymbol{U}}_1 \in \mathbb{R}^{P_1\times I_1}, \hat{\boldsymbol{U}}_2 \in \mathbb{R}^{P_2\times I_2}, \hat{\boldsymbol{U}}_3 \in \mathbb{R}^{P_3\times I_3}$, which contains P_1 basis vectors of the 1-mode linear space $\mathbb{R}^{I_1}$, P_2 basis vectors of the 2-mode linear space $\mathbb{R}^{I_2}$, and P_3 basis vectors of the 3-mode linear space $\mathbb{R}^{I_3}$, respectively. The three linear subspaces form the low-dimensional tensor subspace $\mathbb{R}^{P_1}\otimes\mathbb{R}^{P_2}\otimes\mathbb{R}^{P_3}$ detected by the proposed supervised dimension reduction method.

One of the assumptions of the proposed supervised tensor dimension reduction method in criterion (4) is that the TTF of the asset follows a distribution from the LLS family. This is reasonable because the LLS family includes a variety of TTF distributions, such as (log)normal, (log)-logistic, smallest extreme value, and Weibull, etc., which cover most of the TTF distributions in engineering applications (Doray 1994). Another assumption is that there is a linear relationship between the location parameter and predictors (i.e., degradation signals or their features in this article). Specifically, the location parameters in criterion (3) are expressed as $\mathbf{1}_M\beta_0 + \boldsymbol{S}_{(4)}\boldsymbol{\beta}_1$, which are linear weighted combinations of the rows of $\boldsymbol{S}_{(4)}$. This assumption is widely used in LLS regression (Doray 1994, Fang et al. 2019). However, if a simple linear weighted combination is not adequate to characterize the association between the location parameter and degradation signal features, a high-order polynomial relationship can be constructed (Hastie et al. 2009). By doing so, we can model a more complex association between the location parameter and features. More importantly, the incorporation of high-order polynomial terms into the proposed supervised tensor dimension reduction method does not affect the effectiveness of the optimization algorithms for parameter estimation to be discussed in Sections 3 and 4.

2.3. Prognostic Model Construction and Real-Time TTF Prediction

In this subsection, we discuss how to build a prognostic model based on the supervised dimension reduction method proposed in Section 2.2 and how to predict the TTF distribution of an asset operating in the field using its real-time degradation image data.

Similar to Section 2.2, we denote the training data set as $\{\mathcal{X}_m \in \mathbb{R}^{I_1\times I_2\times D_m}, y_m\}_{m=1}^{M}$, where M is the number of failed assets in the training data set. Notice that D_m might not be the same as $D_{m'}$ for two assets m and m', $m = 1,\ldots,M, m' = 1,\ldots,M, m \neq m'$. This is because different asset's failure times (i.e., TTFs) are different, and usually, no image data can be collected beyond an asset's failure time because the asset is stopped for maintenance or replaced once it is failed. In addition to the training data, we denote the degradation image stream of a test asset by time t as $\mathcal{X}_t \in \mathbb{R}^{I_1\times I_2\times I_t}$. The objectives of this subsection include 1) constructing a prognostic model and estimating its parameters using $\{\mathcal{X}_m, y_m\}_{m=1}^{M}$ in the training data set and 2) using the estimated prognostic model to predict the TTF (denoted as $\hat{y}_t$) of the test asset based on its degradation image stream $\mathcal{X}_t$.

We first use the proposed supervised dimension reduction method to extract low-dimensional features of both the training and test assets. Specifically, as discussed in Section 2.2, we first construct a 4D tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3\times M}$ using the degradation image streams of the training assets, where $I_3 = \max(\{D_m\}_{m=1}^{M})$. Note that $\mathcal{X}$ is an incomplete tensor no matter whether the image streams from the training assets are complete or incomplete. This is because the TTFs of training assets are different, and thus not all the training assets have I_3 images. To detect the low-dimensional tensor subspace in which the high-dimensional degradation images are embedded, we solve optimization criterion (4) by using training data $\{\mathcal{X}, \boldsymbol{y}\}$, where $\boldsymbol{y} = (y_1,\ldots,y_M)^\top$. This yields basis matrices $\hat{\boldsymbol{U}}_1 \in \mathbb{R}^{P_1\times I_1}, \hat{\boldsymbol{U}}_2 \in \mathbb{R}^{P_2\times I_2}, \hat{\boldsymbol{U}}_3 \in \mathbb{R}^{P_3\times I_3}$, which form the low-dimensional tensor subspace $\mathbb{R}^{P_1}\otimes\mathbb{R}^{P_2}\otimes\mathbb{R}^{P_3}$. To extract the low-dimensional features of the training and test assets, we expand the image streams in the low-dimensional tensor subspace $\mathbb{R}^{P_1}\otimes\mathbb{R}^{P_2}\otimes\mathbb{R}^{P_3}$ using the basis matrices $\hat{\boldsymbol{U}}_1, \hat{\boldsymbol{U}}_2, \hat{\boldsymbol{U}}_3$. This is achieved by solving the following optimization criteria,

$$\hat{\mathcal{S}}_m = \arg\min_{\mathcal{S}_m} \|\mathcal{P}_\Omega(\mathcal{X}_m - \mathcal{S}_m \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2. \quad (5)$$

$$\hat{\mathcal{S}}_t = \arg\min_{\mathcal{S}_t} \|\mathcal{P}_\Omega(\mathcal{X}_t - \mathcal{S}_t \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2, \quad (6)$$

where $\{\hat{\mathcal{S}}_m\}_{m=1}^{M}$ are the low-dimensional feature tensors of the M assets in the training data set, and $\hat{\mathcal{S}}_t$ is the low-dimensional feature tensor of the test asset.

Next, we construct a prognostic model using the low-dimensional feature tensors of the M assets in the training data set (i.e., $\{\hat{\mathcal{S}}_m\}_{m=1}^{M}$) along with their TTFs $\{y_m\}_{m=1}^{M}$. Specifically, we build the following LLS regression model,

$$y_m = \gamma_0 + vec(\hat{\mathcal{S}}_m)^\top \boldsymbol{\gamma}_1 + \sigma\epsilon_m, \quad (7)$$

where $vec(\hat{\mathcal{S}}_m)$ is the vectorization of $\hat{\mathcal{S}}_m$. $\gamma_0 \in \mathbb{R}$ and $\boldsymbol{\gamma}_1 \in \mathbb{R}^{(P_1\times P_2\times P_3)\times 1}$ are the regression coefficients, σ is the scale parameter, and ϵ_m is the random noise term with a standard location-scale probability density function $f(\epsilon)$. For example, $f(\epsilon) = 1/\sqrt{2\pi}\exp(-\epsilon^2/2)$ for a normal distribution and $f(\epsilon) = \exp(\epsilon - \exp(\epsilon))$ for an SEV distribution. The parameters in criterion (7) can be estimated by

solving the following optimization problem,

$$\min_{\boldsymbol{y},\gamma_0,\boldsymbol{\gamma}_1,\sigma} \ell\left(\frac{\boldsymbol{y}-\mathbf{1}_M\gamma_0-\hat{\mathbf{S}}_{(4)}\boldsymbol{\gamma}_1}{\sigma}\right), \tag{8}$$

where $\ell(\cdot)$ is the negative log-likelihood function of a location-scale distribution, $\boldsymbol{y}=(y_1,y_2,\ldots,y_m)^\top$ and $\hat{\mathbf{S}}_{(4)}=(vec(\hat{\mathcal{S}}_1)^\top,vec(\hat{\mathcal{S}}_2)^\top,\ldots,vec(\hat{\mathcal{S}}_M)^\top)^\top$, and $\ell(\cdot)$ is the negative log-likelihood function. We conduct the following reparameterization to transform the optimization to be a convex one: $\tilde{\sigma}=1/\sigma,\tilde{\gamma}_0=\gamma_0/\sigma,\tilde{\boldsymbol{\gamma}}_1=\boldsymbol{\gamma}_1/\sigma$:

$$\{\hat{\tilde{\gamma}}_0,\hat{\tilde{\boldsymbol{\gamma}}}_1,\hat{\tilde{\sigma}}\}=\arg\min_{\tilde{\gamma}_0,\tilde{\boldsymbol{\gamma}}_1,\tilde{\sigma}} \ell(\tilde{\sigma}\boldsymbol{y}-\mathbf{1}_M\tilde{\gamma}_0-\hat{\mathbf{S}}_{(4)}\tilde{\boldsymbol{\gamma}}_1). \tag{9}$$

Solving (9) provides the estimated parameters $\{\hat{\tilde{\gamma}}_0,\hat{\tilde{\boldsymbol{\gamma}}}_1,\hat{\tilde{\sigma}}\}$, which can be transformed back to the estimation of the parameters in the LLS regression model: $\hat{\gamma}_0=\hat{\tilde{\gamma}}_0/\hat{\tilde{\sigma}}$, $\hat{\boldsymbol{\gamma}}_1=\hat{\tilde{\boldsymbol{\gamma}}}_1/\hat{\tilde{\sigma}}$ and $\hat{\sigma}=1/\hat{\tilde{\sigma}}$. As a result, the fitted LLS regression model is $\hat{y}_m\sim LLS(\hat{\gamma}_0+vec(\hat{\mathcal{S}}_m)^\top\hat{\boldsymbol{\gamma}}_1,\hat{\sigma})$, where $\hat{\gamma}_0+vec(\hat{\mathcal{S}}_m)^\top\hat{\boldsymbol{\gamma}}_1$ and $\hat{\sigma}$ are, respectively, the estimated location and scale parameters.

Finally, we feed the extracted low-dimensional feature tensor of the test asset into the estimated LLS regression model to predict the asset's TTF distribution: $\hat{y}_t\sim LLS(\hat{\gamma}_0+vec(\hat{\mathcal{S}}_t)^\top\hat{\boldsymbol{\gamma}}_1,\hat{\sigma})$.

3. The Optimization Algorithm

In this section, we discuss how to solve the supervised tensor dimension reduction method proposed in Section 2.2. In Section 3.1, we develop a block updating algorithm to solve criterion (4). The algorithm splits the unknown parameters in criterion (4) into several blocks, and it cyclically optimizes one block parameter while keeping other blocks fixed until convergence. The suboptimization problem for each block is convex, so the convergence of the block updating algorithm is guaranteed. In Section 3.2, we discuss the initialization of the proposed algorithm and hyperparameter tuning.

3.1. The Block Updating Algorithm

The block updating algorithm first splits the unknown parameters in criterion (4) into five blocks, that is, $\boldsymbol{U}_1$, $\boldsymbol{U}_2,\boldsymbol{U}_3,\mathcal{S}$ and $\{\tilde{\beta}_0,\tilde{\boldsymbol{\beta}}_1,\tilde{\sigma}\}$. It then cyclically optimizes one block of parameters each time while keeping other blocks fixed.

Specifically, at the kth iteration, $\boldsymbol{U}_1$ is updated by solving the following optimization problem while keeping other blocks (i.e., $\boldsymbol{U}_2^{k-1},\boldsymbol{U}_3^{k-1},\tilde{\sigma}^{k-1},\tilde{\beta}_0^{k-1},\tilde{\boldsymbol{\beta}}_1^{k-1},\mathcal{S}^{k-1}$) fixed:

$$\begin{aligned}\boldsymbol{U}_1^k&=\arg\min_{\boldsymbol{U}_1}\ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}^{k-1}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^{k-1\top}\times_3\boldsymbol{U}_3^{k-1\top})\|_F^2\\&\quad+(1-\alpha)\ell(\tilde{\sigma}^{k-1},\tilde{\beta}_0^{k-1},\tilde{\boldsymbol{\beta}}_1^{k-1},\boldsymbol{S}_{(4)}^{k-1})\\&=\arg\min_{\boldsymbol{U}_1}\ \|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}^{k-1}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^{k-1\top}\times_3\boldsymbol{U}_3^{k-1\top})\|_F^2\end{aligned} \tag{10}$$

Similarly, the remaining blocks are updated as follows:

$$\begin{aligned}\boldsymbol{U}_2^k&=\arg\min_{\boldsymbol{U}_2}\ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}^{k-1}\times_1\boldsymbol{U}_1^{k\top}\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^{k-1\top})\|_F^2\\&\quad+(1-\alpha)\ell(\tilde{\sigma}^{k-1},\tilde{\beta}_0^{k-1},\tilde{\boldsymbol{\beta}}_1^{k-1},\boldsymbol{S}_{(4)}^{k-1})\\&=\arg\min_{\boldsymbol{U}_2}\ \|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}^{k-1}\times_1\boldsymbol{U}_1^{k\top}\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^{k-1\top})\|_F^2\end{aligned} \tag{11}$$

$$\begin{aligned}\boldsymbol{U}_3^k&=\arg\min_{\boldsymbol{U}_3}\ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}^{k-1}\times_1\boldsymbol{U}_1^{k\top}\times_2\boldsymbol{U}_2^{k\top}\times_3\boldsymbol{U}_3^\top)\|_F^2\\&\quad+(1-\alpha)\ell(\tilde{\sigma}^{k-1},\tilde{\beta}_0^{k-1},\tilde{\boldsymbol{\beta}}_1^{k-1},\boldsymbol{S}_{(4)}^{k-1})\\&=\arg\min_{\boldsymbol{U}_3}\ \|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}^{k-1}\times_1\boldsymbol{U}_1^{k\top}\times_2\boldsymbol{U}_2^{k\top}\times_3\boldsymbol{U}_3^\top)\|_F^2\end{aligned} \tag{12}$$

$$\begin{aligned}&\{\tilde{\sigma}^k,\tilde{\beta}_0^k,\tilde{\boldsymbol{\beta}}_1^k\}\\&=\arg\min_{\tilde{\sigma}^k,\tilde{\beta}_0,\tilde{\boldsymbol{\beta}}_1}\ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}^{k-1}\times_1\boldsymbol{U}_1^{k\top}\times_2\boldsymbol{U}_2^{k\top}\times_3\boldsymbol{U}_3^{k\top})\|_F^2\\&\quad+(1-\alpha)\ell(\tilde{\sigma},\tilde{\beta}_0,\tilde{\boldsymbol{\beta}}_1,\boldsymbol{S}_{(4)}^{k-1})\\&=\arg\min_{\tilde{\sigma}^k,\tilde{\beta}_0,\tilde{\boldsymbol{\beta}}_1}\ \ell(\tilde{\sigma},\tilde{\beta}_0,\tilde{\boldsymbol{\beta}}_1,\boldsymbol{S}_{(4)}^{k-1})\end{aligned} \tag{13}$$

$$\begin{aligned}\mathcal{S}^k&=\arg\min_{\mathcal{S}}\ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^{k\top}\times_2\boldsymbol{U}_2^{k\top}\times_3\boldsymbol{U}_3^{k\top})\|_F^2\\&\quad+(1-\alpha)\ell(\tilde{\sigma}^k,\tilde{\beta}_0^k,\tilde{\boldsymbol{\beta}}_1^k,\boldsymbol{S}_{(4)})\end{aligned} \tag{14}$$

We summarize the block updating algorithm in Algorithm 1 below. The convergence criterion can be set as $\Psi(\boldsymbol{U}_1^k,\boldsymbol{U}_2^k,\ \boldsymbol{U}_3^k,\tilde{\sigma}^k,\tilde{\beta}_0^k,\tilde{\boldsymbol{\beta}}_1^k,\mathcal{S}^k)-\Psi(\boldsymbol{U}_1^{k+1},\boldsymbol{U}_2^{k+1},\boldsymbol{U}_3^{k+1},\tilde{\sigma}^{k+1},\tilde{\beta}_0^{k+1},\tilde{\boldsymbol{\beta}}_1^{k+1},\ \mathcal{S}^{k+1})<\epsilon$, where Ψ is the value of the objective function in criterion (4), and ϵ is a small number. It is easy to show that subproblems (10), (11), and (12) are convex. For normal, logistic, and SEV distributions, their negative log-likelihood functions $\ell(\cdot)$ are also convex, so objective functions (13) and (14) are convex as well. As a result, the block updating algorithm converges to a stationary point of criterion (4).

Algorithm 1 (Block Updating Algorithm for Solving Criterion (4))

1. **Input:** Tensor $\mathcal{X}$ constructed from the (incomplete) degradation image streams of M assets and the TTF vector $\boldsymbol{y}$; the dimension of the low-dimensional tensor subspace $\{P_1,P_2,P_3\}$
2. **Initialization:** Initialize $(\boldsymbol{U}_1^0,\boldsymbol{U}_2^0,\boldsymbol{U}_3^0,\tilde{\sigma}^0,\tilde{\beta}_0^0,\tilde{\boldsymbol{\beta}}_1^0,\mathcal{S}^0)$ randomly or heuristically
3. **While** convergence criterion not met, **do**
4. $\boldsymbol{U}_1^k\leftarrow$ (10)
5. $\boldsymbol{U}_2^k\leftarrow$ (11)
6. $\boldsymbol{U}_3^k\leftarrow$ (12)
7. $(\tilde{\sigma}^k,\tilde{\beta}_0^k,\tilde{\boldsymbol{\beta}}_1^k)\leftarrow$ (13)

8. $\mathcal{S}^k \leftarrow (14)$
9. $k = k+1$ **End While**
10. **Output:** Basis matrices of the low-dimensional tensor subspace $\{\boldsymbol{U}_1^k, \boldsymbol{U}_2^k, \boldsymbol{U}_3^k\}$

3.2. Initialization and Hyperparameter Tuning

To run Algorithm 1, we need to initialize the parameters $\boldsymbol{U}_1^0, \boldsymbol{U}_2^0, \boldsymbol{U}_3^0, \tilde{\sigma}^0, \tilde{\beta}_0^0, \tilde{\boldsymbol{\beta}}_1^0, \mathcal{S}^0$. The initialization can be accomplished randomly or heuristically. In this article, we propose a heuristic initialization method. Specifically, if tensor $\mathcal{X}$ has no missing entries, MPCA (Lu et al. 2008) is applied to tensor $\mathcal{X}$, which yields $\{\boldsymbol{U}_1^0, \boldsymbol{U}_2^0, \boldsymbol{U}_3^0\}$. Next, we compute $\mathcal{S}^0$ by solving $\mathcal{S}^0 = \arg\min_{\mathcal{S}} \|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S} \times_1 \boldsymbol{U}_1^{0\top} \times_2 \boldsymbol{U}_2^{0\top} \times_3 \boldsymbol{U}_3^{0\top})\|_F^2$. Finally, $\tilde{\beta}_0^0, \tilde{\boldsymbol{\beta}}_1^0, \tilde{\sigma}^0$ are computed by solving $\min_{\tilde{\beta}_0^0, \tilde{\boldsymbol{\beta}}_1^0, \tilde{\sigma}^0} \ell(\tilde{\sigma}^0 \boldsymbol{y} - \mathbf{1}_M \tilde{\beta}_0^0 - \boldsymbol{S}_{(4)}^0 \tilde{\boldsymbol{\beta}}_1^0)$, where $\boldsymbol{S}_{(4)}^0$ is the mode-4 matricization of $\mathcal{S}^0$. If tensor $\mathcal{X}$ has missing values, a tensor completion method (Liu et al. 2012, Xu et al. 2013, Filipović and Jukić 2015) can be conducted before applying MPCA.

In addition to the initialization, the hyperparameter parameters, including the weight α and the dimension of tensor subspace (P_1, P_2, P_3), also need to be predetermined. It is known that α controls the weights of the feature extraction term and the regression term, and $\alpha \in [0,1]$. To select an appropriate weight parameter, we will first split the range $[0,1]$ into $L+1$ intervals equally, which yields $\alpha_0 = 0/L, \alpha_1 = 1/L, \alpha_2 = 2/L, \ldots, \alpha_L = L/L$. Next, we employ cross-validation to select the weight that achieves the highest prediction accuracy. If the weight at the boundary is selected (i.e., $\alpha_0 = 0/L$ or $\alpha_L = L/L$), we further split the interval closest to the boundary and conduct cross-validation again. For example, if $\alpha_0 = 0/L$ is chosen as the best weight, we will split $[0/L, 1/L]$ into $(L+1)$ intervals equally and reconduct the cross-validation. This process is repeated until a nonboundary weight is selected. Of course, a maximum number of repetitions needs to be set to control the computational time.

The values of $\{P_1, P_2, P_3\}$ can be determined using cross-validation as well. To be specific, we may try a certain number of candidate values for $\{P_1, P_2, P_3\}$ and run Algorithm 1 to extract low-dimensional features, which are then used to build the prognostic model discussed in Section 2.3 for TTF prediction. The values that achieve the smallest prediction error will be chosen. It is known that there usually exist high spatio-temporal correlations among degradation image streams (Fang et al. 2019), so the dimension of the tensor subspace is usually low, which helps reduce the computation intensity of model selection. The values of $\{P_1, P_2, P_3\}$ can also be determined heuristically. For example, if MPCA is employed for parameter initialization, then the fraction of variance explained (Lu et al. 2008) can be used to determine the dimension of tensor subspace.

4. Analytical Solutions

In this section, we discuss the closed-form solutions of optimization problems (10), (11), (12), (13), and (14) in Algorithm 1. Specifically, we will discuss the solutions when degradation image streams are complete and incomplete in Sections 4.1 and 4.2, respectively. For simplicity, we will remove the superscripts k and $k-1$.

4.1. Analytical Solutions for Complete Data

4.1.1. Solution Procedure for $\boldsymbol{U}_1$. When degradation image streams are complete (i.e., the 4D image tensor $\mathcal{X}$ in criterion (4) has no missing entries), we have the following proposition, which provides the analytical solution to problem (10).

Proposition 1. *If the 4D tensor $\mathcal{X}$ has no missing values, optimization problem (10) has the following analytical solution,*

$$\boldsymbol{U}_1 = (\boldsymbol{X}_{(1)} \cdot \boldsymbol{S}_{U_1(1)}^\top \cdot (\boldsymbol{S}_{U_1(1)} \cdot \boldsymbol{S}_{U_1(1)}^\top)^{-1})^\top,$$

where $\boldsymbol{X}_{(1)}$ is the mode-1 matricization of $\mathcal{X}$, $\mathcal{S}_{U_1} = \mathcal{S} \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top$, $\boldsymbol{S}_{U_1(1)}$ is the mode-1 matricization of $\mathcal{S}_{U_1}$, and the operator "$\cdot$" represents multiplication.

4.1.2. Solution Procedure for $\boldsymbol{U}_2$. When degradation image streams are complete, the proposition below gives the analytical solution to problem (11).

Proposition 2. *If the 4D tensor $\mathcal{X}$ has no missing values, optimization problem (11) has the following analytical solution,*

$$\boldsymbol{U}_2 = (\boldsymbol{X}_{(2)} \cdot \boldsymbol{S}_{U_2(2)}^\top \cdot (\boldsymbol{S}_{U_2(2)} \cdot \boldsymbol{S}_{U_2(2)}^\top)^{-1})^\top,$$

where $\boldsymbol{X}_{(2)}$ is the mode-2 matricization of $\mathcal{X}$, $\mathcal{S}_{U_2} = \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_3 \boldsymbol{U}_3^\top$, and $\boldsymbol{S}_{U_2(2)}$ is the mode-2 matricization of $\mathcal{S}_{U_2}$.

4.1.3. Solution Procedure for $\boldsymbol{U}_3$. When degradation image streams are complete, the proposition below provides the analytical solution to problem (12).

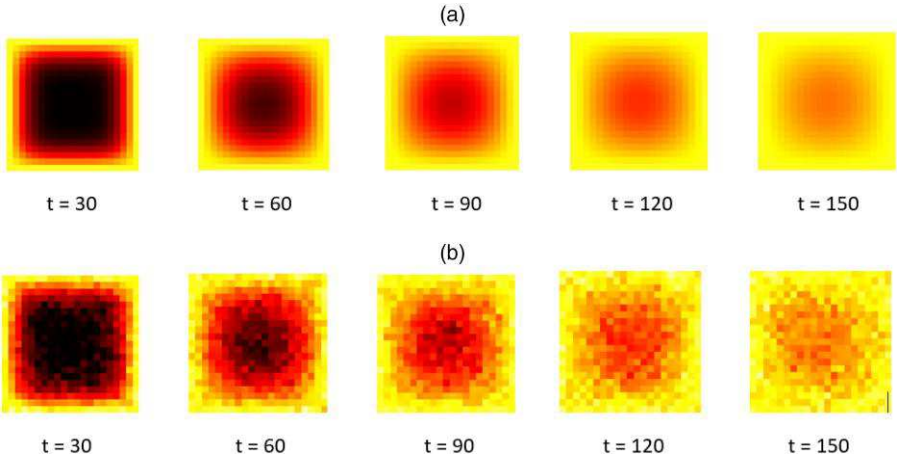
Proposition 3. *If the 4D tensor $\mathcal{X}$ has no missing values, optimization problem (12) has the following analytical solution,*

$$\boldsymbol{U}_3 = (\boldsymbol{X}_{(3)} \cdot \boldsymbol{S}_{U_3(3)}^\top \cdot (\boldsymbol{S}_{U_3(3)} \cdot \boldsymbol{S}_{U_3(3)}^\top)^{-1})^\top,$$

where $\boldsymbol{X}_{(3)}$ is the mode-3 matricization of $\mathcal{X}$, $\mathcal{S}_{U_3} = \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top$, and $\boldsymbol{S}_{U_3(3)}$ is the mode-3 matricization of $\mathcal{S}_{U_3}$.

4.1.4. Solution Procedure for $\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}_1, \tilde{\sigma}$. For general LLS distributions, there is no closed-form solution for $\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}_1, \tilde{\sigma}$. As a result, we may use existing algorithms (Doray 1994) or convex optimization packages to solve problem (13). However, if the TTF follows a normal (or lognormal) distribution, we may replace the negative log-likelihood term in criterion (3) with a mean squared error-based loss function, which results in the following

Figure 2. (Color online) Simulated Degradation Images Based on Heat Transfer Process



Note. (a) Without noise; (b) with noise.

then optimization problem (10) has the following analytical solution,

$$\boldsymbol{U}_1 = (\boldsymbol{X}^{\pi}_{(1)} \cdot \boldsymbol{S}^{\pi}_{U_1(1)}{}^{\top} \cdot (\boldsymbol{S}^{\pi}_{U_1(1)} \cdot \boldsymbol{S}^{\pi}_{U_1(1)}{}^{\top})^{-1})^{\top},$$

where $\boldsymbol{X}^{\pi}_{(1)}$ is a matrix consisting of the π columns of $\boldsymbol{X}_{(1)}$, $\mathcal{S}_{U_1} = \mathcal{S} \times_2 \boldsymbol{U}_2^{\top} \times_3 \boldsymbol{U}_3^{\top}$, $\boldsymbol{S}_{U_1(1)}$ is the mode-1 matricization of $\mathcal{S}_{U_1}$, and $\boldsymbol{S}^{\pi}_{U_1(1)}$ denotes a matrix constituting the π columns of $\boldsymbol{S}_{U_1(1)}$.

4.2.2. Solution Procedure for $\boldsymbol{U}_2$. Similar to $\boldsymbol{U}_1$, there is no closed-form solution for $\boldsymbol{U}_2$ in optimization criterion (11) when tensor $\mathcal{X}$ has a general entry-wise missing structure. However, Lemma 1 implies that we may also decompose optimization problem (11) into multiple subcriteria, each of which has a closed-form solution. Specifically, denote the i_2th column of matrix $\boldsymbol{U}_2 \in \mathbb{R}^{P_2 \times I_2}$ as $\boldsymbol{u}_2^{i_2} \in \mathbb{R}^{P_2 \times 1}, i_2 = 1, \ldots, I_2$, and we can replace optimization problem (11) with I_2 subproblems by separately optimizing $\boldsymbol{u}_2^1, \boldsymbol{u}_2^2, \ldots, \boldsymbol{u}_2^{I_2}$. Proposition 7 shows that there is an analytical solution for $\boldsymbol{u}_2^{i_2}$.

Proposition 7. *When optimizing the i_2th column of $\boldsymbol{U}_2$ in problem (11), we have the following analytical solution,*

$$\boldsymbol{u}_2^{i_2} = (\boldsymbol{x}_{(2)}^{i_2,\pi_{i_2}} \cdot \boldsymbol{S}^{\pi_{i_2}}_{U_2(2)}{}^{\top} \cdot (\boldsymbol{S}^{\pi_{i_2}}_{U_2(2)} \cdot \boldsymbol{S}^{\pi_{i_2}}_{U_2(2)}{}^{\top})^{-1})^{\top}$$

where $\boldsymbol{x}_{(2)}^{i_2}$ denotes the i_2th row of $\boldsymbol{X}_{(2)}$, π_{i_2} is a set consisting of the indices of available entries of $\boldsymbol{x}_{(2)}^{i_2}$, $\boldsymbol{x}_{(2)}^{i_2,\pi_{i_2}}$ is a vector consisting of the available entries in the i_2th column of $\boldsymbol{X}_{(2)}$, $\mathcal{S}_{U_2} = \mathcal{S} \times_1 \boldsymbol{U}_1^{\top} \times_3 \boldsymbol{U}_3^{\top}$, $\boldsymbol{S}_{U_2(2)}$ is the mode-2 matricization of $\mathcal{S}_{U_2}$, and $\boldsymbol{S}^{\pi_{i_2}}_{U_2(2)}$ denotes a matrix comprising the π_{i_2} columns of $\boldsymbol{S}_{U_2(2)}$.

Similar to $\boldsymbol{U}_1$, when tensor $\mathcal{X}$ has the *image-wise missing* structure, we do not have to optimize each of the columns of $\boldsymbol{U}_2$ separately. Proposition 8 below gives an analytical solution to $\boldsymbol{U}_2$ when tensor $\mathcal{X}$ has missing images.

Proposition 8. *If the indices of tensor $\mathcal{X}$'s missing entries can be denoted as $\Omega \subseteq \{(:, :, i_3, m), 1 \le i_3 \le I_3, 1 \le m \le M\}$, where ":" denotes all the indices in a dimension, then $\mathcal{X}$'s mode-2 matricization $\boldsymbol{X}_{(2)}$ has missing columns. Let π be the set consisting of the indices of available columns in $\boldsymbol{X}_{(2)}$, and then optimization problem (11) has the following analytical solution,*

$$\boldsymbol{U}_2 = (\boldsymbol{X}^{\pi}_{(2)} \cdot \boldsymbol{S}^{\pi}_{U_2(2)}{}^{\top} \cdot (\boldsymbol{S}^{\pi}_{U_2(2)} \cdot \boldsymbol{S}^{\pi}_{U_2(2)}{}^{\top})^{-1})^{\top},$$

where $\boldsymbol{X}^{\pi}_{(2)}$ is a matrix consisting of the π columns of $\boldsymbol{X}_{(2)}$, $\boldsymbol{S}_{U_2(2)}$ is the mode-2 matricization of $\mathcal{S}_{U_2}$, and $\boldsymbol{S}^{\pi}_{U_2(2)}$ denotes a matrix constituting the π columns of $\boldsymbol{S}_{U_2(2)}$.

4.2.3. Solution Procedure for $\boldsymbol{U}_3$. There is no closed-form solution for $\boldsymbol{U}_3$ in optimization criterion (12) when tensor $\mathcal{X}$ has missing entries. Based on Lemma 1, we decompose optimization problem (12) into multiple subcriteria, each of which has a closed-form solution. Denote the i_3th column of matrix $\boldsymbol{U}_3 \in \mathbb{R}^{P_3 \times I_3}$ as $\boldsymbol{u}_3^{i_3} \in \mathbb{R}^{P_3 \times 1}, i_3 = 1, \ldots, I_3$, and we replace optimization problem (12) with I_3 subproblems by separately optimizing $\boldsymbol{u}_3^1, \boldsymbol{u}_3^2, \ldots, \boldsymbol{u}_3^{I_3}$ respectively. Proposition 9 suggests that there is an analytical solution when optimizing $\boldsymbol{u}_3^{i_3}$.

Proposition 9. *When optimizing the i_3th column of $\boldsymbol{U}_3$ in problem (12), we have the following analytical solution,*

$$\boldsymbol{u}_3^{i_3} = (\boldsymbol{x}_{(3)}^{i_3,\pi_{i_3}} \cdot \boldsymbol{S}^{\pi_{i_3}}_{U_3(3)}{}^{\top} \cdot (\boldsymbol{S}^{\pi_{i_3}}_{U_3(3)} \cdot \boldsymbol{S}^{\pi_{i_3}}_{U_3(3)}{}^{\top})^{-1})^{\top},$$

where $\boldsymbol{x}_{(3)}^{i_3}$ denotes the i_3th row of $\boldsymbol{X}_{(3)}$, π_{i_3} is a set consisting of the indices of available entries of $\boldsymbol{x}_{(3)}^{i_3}$, $\boldsymbol{x}_{(3)}^{i_3,\pi_{i_3}}$ is a vector consisting of the available entries in the i_3th row of $\boldsymbol{X}_{(3)}$, $\mathcal{S}_{U_3} = \mathcal{S} \times_1 \boldsymbol{U}_1^{\top} \times_2 \boldsymbol{U}_2^{\top}$, $\boldsymbol{S}_{U_3(3)}$ is the mode-3 matricization of $\mathcal{S}_{U_3}$, and $\boldsymbol{S}^{\pi_{i_3}}_{U_3(3)}$ denotes a matrix comprising the π_{i_3} columns of $\boldsymbol{S}_{U_3(3)}$.

4.2.4. Solution Procedure for $\tilde{\boldsymbol{\beta}}_0, \tilde{\boldsymbol{\beta}}_1, \tilde{\sigma}$. Whether tensor $\mathcal{X}$ contains missing entries does not affect the methods

for $\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}_1$, and $\tilde{\sigma}$ estimation. Therefore, the estimation methods discussed in Section 4.1.4 can still be used.

4.2.5. Solution Procedure for $\mathcal{S}$. For general LLS distributions, there is no closed-form solution for $\mathcal{S}$. Therefore, we may use existing convex optimization packages to solve problem (14). However, if the TTF follows a normal (or lognormal) distribution, problem (14) is equivalent to (16) (see Section 4.1.5 for details). Based on Lemma 1, we may optimize each row of $\boldsymbol{S}_{(4)}$ separately. We denote the mth row of matrix $\boldsymbol{S}_{(4)} \in \mathbb{R}^{M\times(P_1\times P_2\times P_3)}$ as $\boldsymbol{s}^m_{(4)} \in \mathbb{R}^{1\times(P_1\times P_2\times P_3)}$, $m = 1,\ldots,M$, and replace optimization problem (16) with M subproblems, that is, separately optimizing $\boldsymbol{s}^1_{(4)}, \boldsymbol{s}^2_{(4)}, \ldots, \boldsymbol{s}^M_{(4)}$. Proposition 10 suggests that there is an analytical solution when optimizing $\boldsymbol{s}^m_{(4)}$.

Proposition 10. *When optimizing the mth row of matrix $\boldsymbol{S}_{(4)}$ in problem (16), we have the following analytical solution,*

$$\begin{aligned}\boldsymbol{s}^m_{(4)} = [&\alpha\cdot\boldsymbol{x}^{m,\pi_m}_{(4)}\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^{\pi_m\top}\\ &+(1-\alpha)\cdot(y_m-\tilde{\beta}_0)\cdot\tilde{\boldsymbol{\beta}}_1^\top]\cdot\\ [&\alpha\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^{\pi_m}\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^{\pi_m\top}\\ &+(1-\alpha)\cdot\tilde{\boldsymbol{\beta}}_1\cdot\tilde{\boldsymbol{\beta}}_1^\top]^{-1},\end{aligned}$$

where $\boldsymbol{x}^m_{(4)}$ represents the mth row of $\boldsymbol{X}_{(4)}$, π_m denotes the set consisting of the indices of available entries in $\boldsymbol{x}^m_{(4)}$, $\boldsymbol{x}^{m,\pi_m}_{(4)}$ is a vector consisting of the available entries in the mth row of $\boldsymbol{X}_{(4)}$, and $(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^{\pi_m}$ denotes a matrix comprising the π_m columns of matrix $\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1$.

The proof of all Propositions 5, 6, 7, 8, 9, and 10 can be found in the Appendix.

5. Numerical Studies

In this section, we validate the effectiveness of our proposed supervised tensor dimension reduction-based prognostic model using simulated data.

5.1. Data Generation

We generate degradation image streams for 500 assets. The image stream from asset m, which is denoted by $\mathcal{X}_m(x,y,t)$, $m = 1,2,\ldots,500$, is generated from the following heat transfer equation,

$$\frac{\partial\mathcal{X}_m(x,y,t)}{\partial t} = \alpha_m\left(\frac{\partial^2\mathcal{X}_m}{\partial x^2}+\frac{\partial^2\mathcal{X}_m}{\partial y^2}\right), \tag{17}$$

where $(x,y), 0\le x,y\le 0.2$ represents the location of each image pixel. α_m is the thermal diffusivity coefficient, which is randomly generated from a uniform distribution $\mathcal{U}(0.5\times10^{-4}, 1\times10^{-4})$. t is the time index. The initial and boundary conditions are set such that $\mathcal{X}|_{t=1} = 0$ and $\mathcal{X}_m|_{x=0} = \mathcal{X}_m|_{x=0.2} = \mathcal{X}_m|_{y=0} = \mathcal{X}_m|_{y=0.2} = 30$. At each time t, the image is recorded at locations $x = \frac{j}{n+1}, y = \frac{k}{n+1}, j,k = 1,\ldots,n$, resulting in an $n\times n$ matrix. Here, we set $n = 21$ and $t = 1,2,\ldots,150$, which yields 150 images of size 21×21 for each asset. This implies that the degradation image stream of each asset can be represented by a $21\times21\times150$ tensor. In addition, an independent and identically distributed random noise $\epsilon \sim N(0, 0.1)$ is added to each pixel. Figure 2 demonstrates an example of some images with and without noise from one of the assets simulated in this study.

To determine the TTF of an asset, we first transform the asset's $21\times21\times150$ tensor to a 1×150 time series by taking the average pixel intensity of each image. The time series signal indicates how the average heat of the asset involves over time. Next, we let the TTF of the asset be the time point where the amplitude of the time series signal crosses a predefined soft failure threshold, which is set as 23 in this study. Because the images of different assets are generated with different thermal diffusivity coefficients, the time points where their time series signals go beyond the threshold may be different. Thus, the TTF of different assets may also be different. To mimic reality, we truncate the image stream of each asset by keeping only the images observed before its TTF. In other words, any images observed after an asset's TTF are removed from the image tensor of the asset. Such a truncation is normal in reality because an asset usually gets maintained or replaced once its degradation signal crosses the soft failure threshold. Consequently, the third dimension of the tensor of different assets might be different. In addition, to reduce the computation load, we keep one of every 10 images in the truncated image stream of each asset.

5.2. The Benchmark and Performance Comparison

We randomly split the generated data into a training data set consisting of 400 assets and a test data set consisting of the remaining 100 assets. To test the robustness of the proposed method, we consider four levels of data incompleteness: (1) 0% missing, (2) 10% missing, (3) 50% missing, and (4) 90% missing. For the first scenario, (1) 0% missing, we use all of the generated data for model training and testing. Please notice that even though all of the available images are used, the image tensor $\mathcal{X}$ is still incomplete because of failure time truncation; that is, different assets may have different TTFs and thus different numbers of images (see the discussion in the second paragraph of Section 2.3). For the remaining scenarios, we randomly remove some images from each asset's image stream. For example, with 10% missing, we randomly remove 10% of the images (rounding to the nearest integer) from the image stream of each asset.

We compare the performance of our proposed method with an unsupervised tensor dimension reduction-based

Figure 3. (Color online) Prediction Errors When Data are Complete in Numerical Study

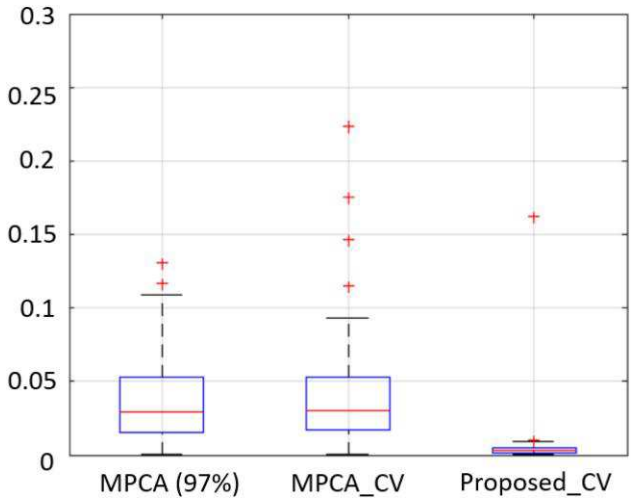
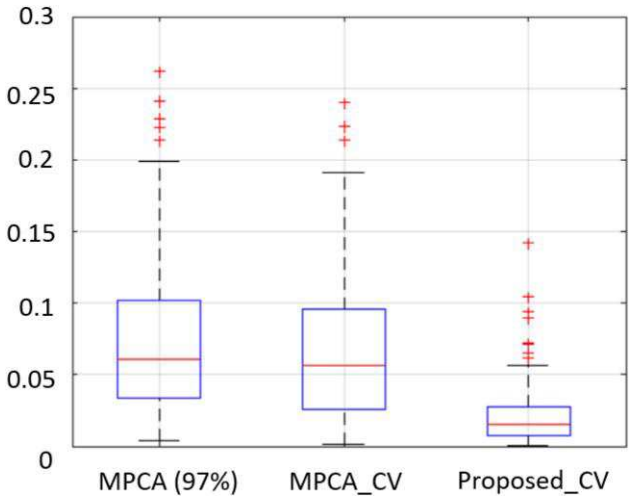


Figure 4. (Color online) Prediction Errors When 10% Data are Missing in Numerical Study



benchmark. Considering that image streams are incomplete, the baseline model first applies a tensor completion method known as TMac, developed by Xu et al. (2013) to impute the missing values of the image tensor. Next, an unsupervised tensor dimension reduction method, MPCA (Lu et al. 2008), is employed to reduce the dimension of the imputed image tensor to reduce dimension and extract low-dimension features, which are then used to build an LLS-based prognostic model, as we discussed in Section 2.3. MPCA is a widely used dimension reduction method for tensor data. It projects a high-dimensional tensor into a subspace but maximizes the total tensor scatter, which is assumed to measure the variations in the original tensor objects. Lu et al. (2008) proposed a fraction-of-variation-explained (FVE) method to determine the dimension of the low-dimension tensor subspace/features, which represents the percentage of variation of the original high-dimensional tensor preserved by the low-dimensional tensor features. Because the optimal FVE suggested by Lu et al. (2008) was 97%, we will first set FVE as 97% in this study, and the corresponding baseline model is designated as "MPCA (97%)." In addition to the FVE method, we also use cross-validation (CV) to select an appropriate dimension for the tensor subspace. Specifically, we use the training data to conduct a 10-fold CV for various combinations of (P_1, P_2, P_3), where $P_1 = 1, \ldots, 4$, $P_2 = 1, \ldots, 4$, and $P_3 = 1, \ldots, 4$. The baseline model is referred to as "MPCA_CV". We also use 10-fold CV to determine the value of the weight parameter α and the appropriate dimension of the tensor subspace of our proposed method.

We use the heuristic method discussed in Section 3.1 to initialize the block updating algorithm. In this study, we use lognormal regression to build the prognostic model. The proposed method is denoted as "Proposed_CV". The prediction errors of our proposed method and two benchmarks are calculated by using the equation below and reported in Figures 3–6.

$$\text{Prediction Error} = \frac{|\text{Estimated TTF} - \text{True TTF}|}{\text{True TTF}}. \tag{18}$$

5.3. Results and Analysis

Figure 3 reports the prediction errors of the two benchmarks and our proposed method when data are complete, which means no image is removed on purpose. Figure 4 shows the prediction errors when 10% entries in the third mode (time) of degradation image streams are missing, whereas Figures 5 and 6 demonstrate the errors when 50% and 90% images are missing, respectively.

Figure 5. (Color online) Prediction Errors When 50% Data are Missing in Numerical Study

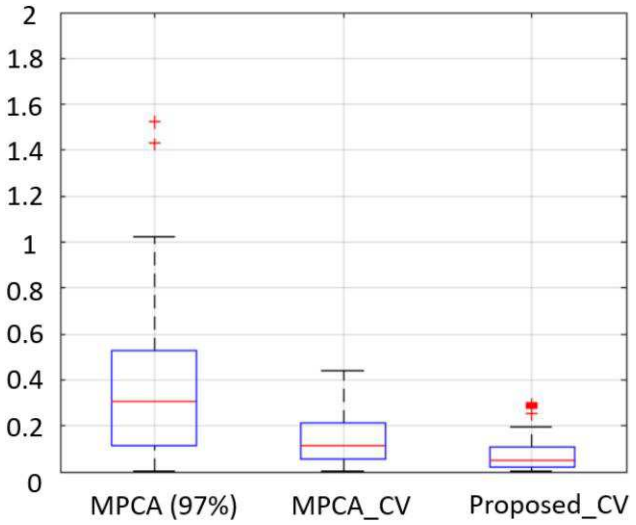
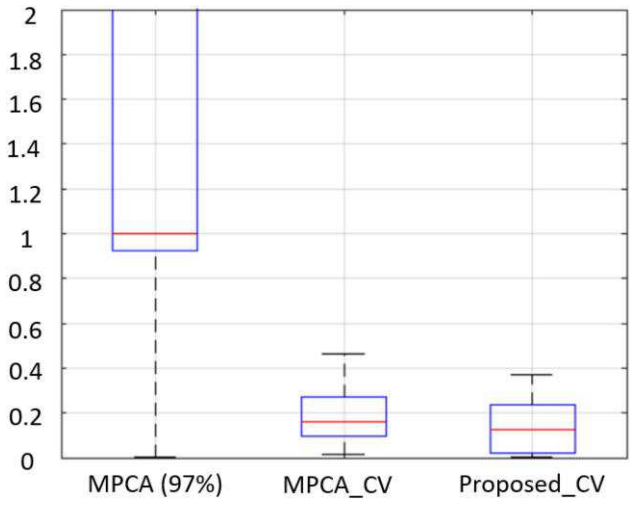


Figure 6. (Color online) Prediction Errors When 90% Data are Missing in Numerical Study



Figures 3–6 illustrate that our proposed method outperforms the benchmarks under all data missing rates. For example, when the degradation image signals are complete, the median absolute prediction errors (and the interquartile ranges, i.e., IQRs) of the proposed method and the two benchmarks are 0.003 (0.003), 0.027 (0.035), and 0.025 (0.033), respectively; when 10% images are missing, the median absolute prediction errors (and IQRs) of the three methods are, respectively, 0.019 (0.017), 0.058 (0.067), and 0.053 (0.063); when 50% of images are missing, they are 0.052 (0.084), 0.302 (0.405), and 0.104 (0.168). We believe this is because our proposed method applies historical TTFs to supervise the low-dimensional tensor dimension reduction, and thus the extracted features are more effective for failure time prediction. Unlike our method, the two baseline models use MPCA, an unsupervised tensor dimension reduction method, for feature extraction. Because the extracted features are determined only by the image streams, and no TTF gets involved, they are not as effective as the features extracted by our proposed method, and thus their failure time prediction accuracy and precision are compromised.

Figures 3–6 also suggest that the performances of all the three models deteriorate, and the superiority of our proposed method over the two benchmarks decreases, with the increase of data missing rate. For example, when data are complete, the median absolute prediction errors (and IQRs) of "Proposed_CV" and "MPCA_CV" are 0.003 (0.003) and 0.025 (0.033), respectively; when the missing rate increases to 90%, they are, respectively, 0.13 (0.21) and 0.16 (0.19), which are almost comparable. This is reasonable because the performances of all the models are compromised more when more data are missing. In addition, no model will perform well if a high percentage (say more than 90%) of data are missing because it implies that very limited useful degradation information is available for modeling.

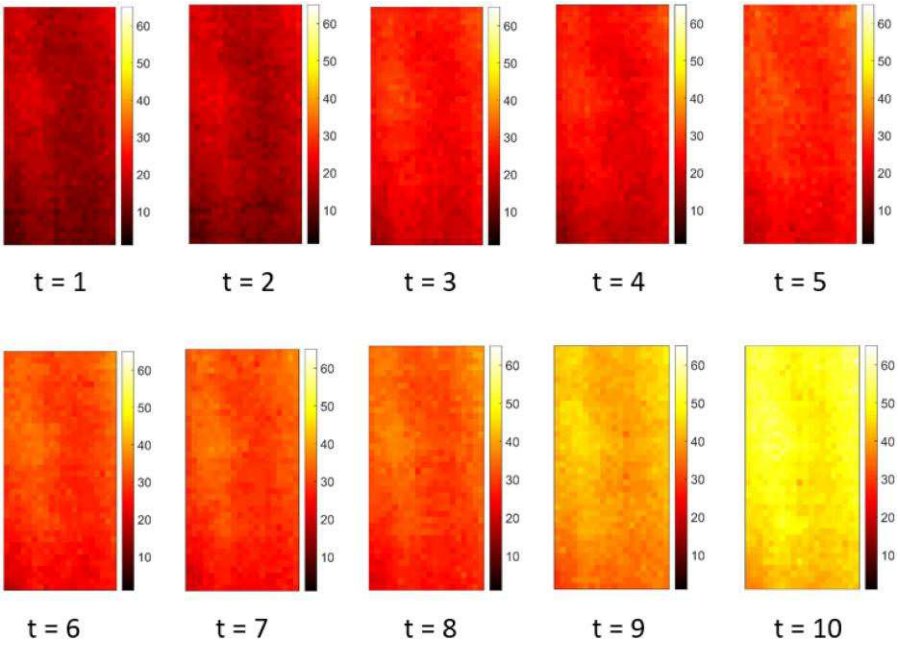
Figures 3–6 also demonstrate that "MPCA_CV" always outperforms "MPCA (97%)", and the superiority of "MPCA_CV" is augmented with the increase of data missing rate. For instance, when 10% images are missing, the median absolute prediction errors (and IQR) of "MPCA (97%)" and "MPCA_CV" are 0.058 (0.067) and 0.053 (0.063), respectively; when the missing rate is 50%, they are 0.302 (0.405) and 0.104 (0.168). One of the possible reasons is that "MPCA (97%)" determines the dimension of the tensor subspace by setting the "FVE" as 97%, which usually results in relatively high-dimensional features, although the dimension is smaller than that of the original image tensor. Relatively high-dimensional features imply an insufficient dimension reduction. In addition, it means the number of parameters in the subsequent LLS-based prognostic model is relatively large, which poses estimation challenges given that the number of samples (assets) for model training is limited.

6. Case Study

In this section, we use degradation image streams obtained from a rotating machinery test bed to validate the effectiveness of our proposed method. The test bed is designed to perform accelerated degradation tests on rolling entry thrust bearings. Specifically, bearings were run from brand new to failure. An FLIR T300 infrared camera was used to monitor the degradation process and collect degradation images over time. In the meantime, an accelerometer was mounted on the test bed to monitor the vibration of the bearing, and the failure time was defined as the time point where the amplitude of defective vibration frequencies crossed a threshold based on ISO standards for machine vibration. The data set consists of 284 degradation image streams and their corresponding TTFs, and each image has 40×20 pixels. As an illustration, a sequence of images obtained at different (ordered) time periods of one of the bearings is shown in Figure 7. More details about the experimental setup and the data set can be found in Gebraeel et al. (2009) and Fang et al. (2019).

We use fivefold cross-validation to evaluate the performance of our proposed model and the two benchmarks discussed in Section 5. Similar to the simulation study, we conduct 10-fold cross-validation to determine the optimal weight parameter in criterion (4) and the most appropriate dimension of the tensor subspace. In addition, we also consider four levels of data incompleteness: (1) 0% missing (i.e., complete), (2) 10% missing, (3) 50% missing, and (4) 90% missing. Figure 8 illustrates the absolute prediction errors when degradation image streams are complete. Figure 9 shows prediction errors when 10% of the images of each bearing

Figure 7. (Color online) An Illustration of One Infrared Degradation Image Stream



are missing. Figures 10 and 11 demonstrate the absolute prediction errors when the missing rates are 50% and 90%, respectively.

Similar to the discovery in the numerical study in Section 5, Figures 8–11 indicate that our proposed method constantly works better than the two benchmarks under all the 4 data missing rates. For example, the median absolute prediction errors (and IQRs) of our proposed method and the two benchmarks are 0.03 (0.04), 0.3 (0.24), and 0.1 (0.16), respectively, when the degradation image streams are complete. When 50% of the images are missing, the median absolute prediction errors (and IQR) are, respectively, 0.09 (0.17), 0.62 (0.65), and 0.15 (0.19). We believe this is because our proposed model is a supervised dimension reduction-based method, which uses TTF information to supervise the defection of the low-dimensional tensor subspace, whereas the benchmarks are unsupervised dimension reduction-based methods without TTF information involved. Because our method considers TTF information when detecting the tensor subspace, the extracted features are more effective for failure time prediction.

Figures 8–11 also show that the prediction errors of all the 3 methods increase with the increase of data missing

Figure 8. (Color online) Prediction Errors When Data are Complete in Case Study

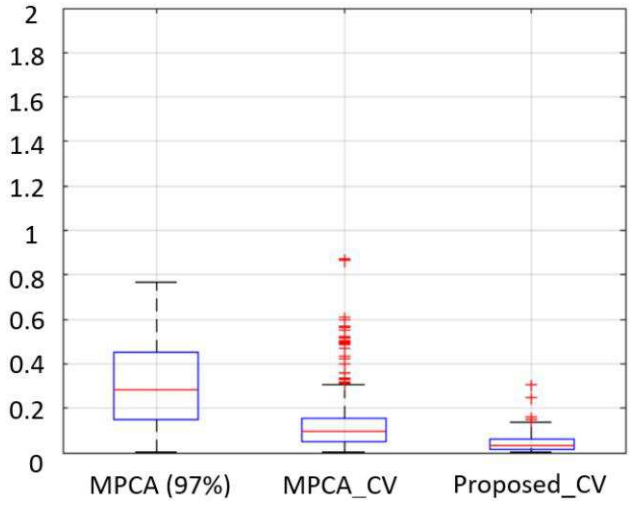


Figure 9. (Color online) Prediction Errors When 10% Data are Missing in Case Study

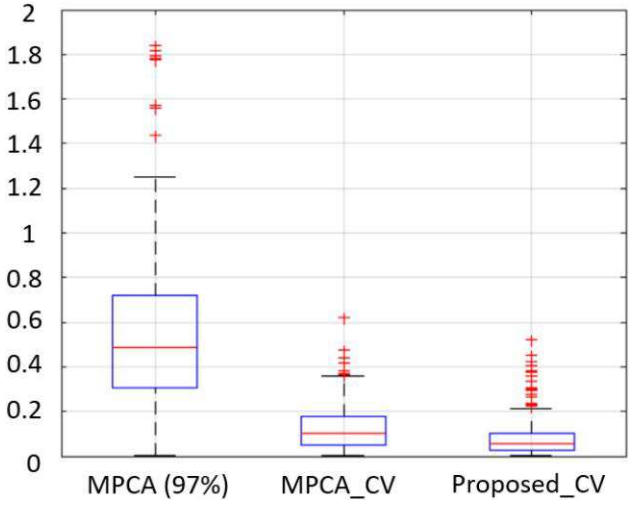


Figure 10. (Color online) Prediction Errors When 50% Data are Missing in Case Study

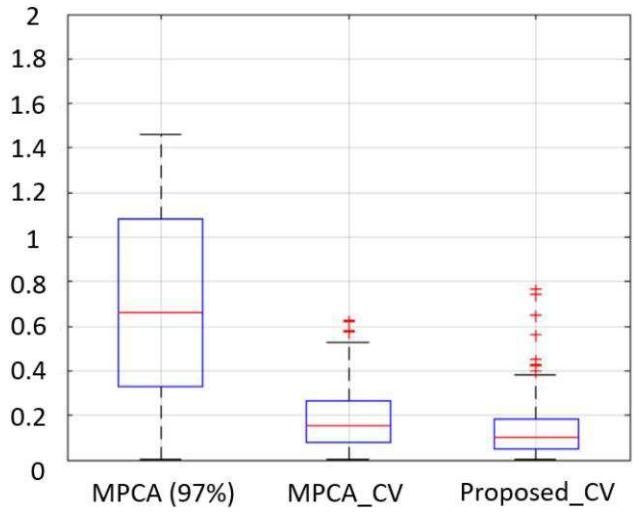
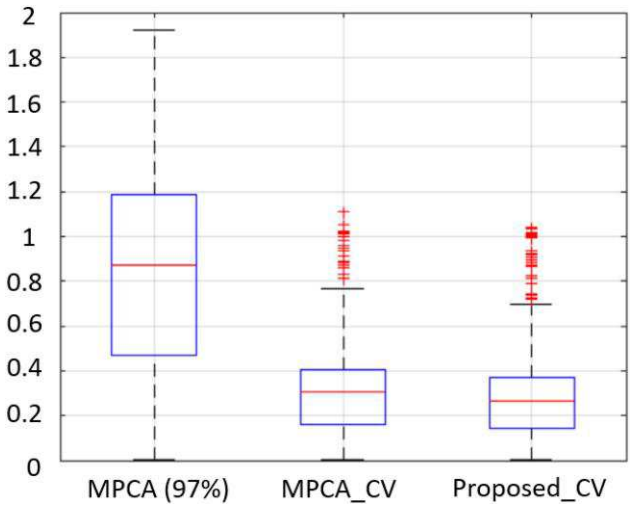


Figure 11. (Color online) Prediction Errors When 90% Data are Missing in Case Study



rates. For example, when the missing rates are 0%, 10%, 50%, and 90%, the median absolute prediction errors (and IQRs) of "MPCA (97%)" are 0.3 (0.24), 0.49 (0.42), 0.62 (0.65), and 0.83 (0.78), respectively, whereas they are, respectively, 0.09 (0.11), 0.1 (0.16), 0.15 (0.19), and 0.31 (0.22) for "MPCA_CV", and 0.03 (0.04), 0.05 (0.08), 0.09 (0.17), and 0.29 (0.21) for our proposed method. This is reasonable because a higher data missing rate means less useful degradation information and thus a worse model performance. In addition, we observe that the superiority of our proposed method over the two benchmarks decreases with the increase of data missing rates. For example, the prediction accuracy of our method and "MPCA (97%)" are comparable when the data missing rate is 90%. Again, we believe that this is because not much useful information is available when data are highly incomplete, and neither of the two models perform well with such limited data.

We also observe that "MPCA_CV" always outperforms "MPCA (97%)," and the superiority of "MPCA_CV" is augmented with the increase of the data missing rate. For example, when 10% images are missing, the median absolute prediction errors (and IQR) of "MPCA (97%)" and MPCA_CV are 0.49 (0.42) and 0.1 (0.16), respectively; when the missing rate is 50%, they are 0.62 (0.65) and 0.15 (0.19). Again, we believe that this is because "MPCA (97%)" determines the dimension of the tensor subspace by setting the "FVE" as 97%, which results in relatively high-dimensional features because of the insufficient dimension reduction. Also, it results in a parameter estimation challenge because the number of parameters to be estimated in the prognostic model is relatively large compared to the limited number of historical samples for model training. This suggests that cross-validation is a better method to determine the dimension of the tensor subspace, especially when we do not have enough number of samples for model training.

7. Conclusions

This paper proposed a supervised tensor dimension reduction-based prognostic model for applications with incomplete degradation imaging data. This is achieved by first developing a new supervised tensor dimension reduction method that reduces the dimension of incomplete high-dimensional degradation image streams and provides low-dimensional tensor features, which are then used to build a prognostic model based on (log)-location-scale regression.

The supervised tensor dimension reduction method uses historical TTFs to supervise the detection of a low-dimensional tensor subspace to reduce the dimension of incomplete high-dimensional image streams. Mathematically, it is formulated as an optimization criterion that combines a feature extraction term and a regression term. The feature extraction term focuses on identifying a tensor space to extract low-dimensional tensor features from high-dimensional image streams. The regression term regresses failure times against the features extracted by the first term using LLS regression. By jointly optimizing the two terms, it is expected to detect an appropriate tensor subspace such that the extracted features are effective for TTF prediction. To estimate the parameters of the supervised dimension reduction method, we developed a block updating algorithm for applications where TTFs follow distributions in the (log)-location-scale family. The algorithm works by splitting the parameters into several blocks and cyclically optimizing one block of parameters while keeping other blocks fixed until convergence. In addition, we showed that if TTFs follow normal or lognormal distributions, there is a closed-form solution when optimizing each block of the parameters,

no matter whether the imaging data are complete or incomplete.

Simulated data as well as a data set from rotating machinery were used to validate the effectiveness of our proposed method. The results showed that our proposed prognostic method consistently outperforms the unsupervised tensor reduction-based benchmarks under various data missing rates. This validated the benefits and importance of using failure time information to supervise the dimension reduction of high-dimensional degradation image streams when building prognostic models.

The proposed prognostic model assumes that the TTFs of assets in the training data set are known. In many real-world applications, the historical failure times might be right censored. This is because a component might be replaced before failure, so the exact TTF is unknown, and we know only that it is larger than the replacement time. How to incorporate censored TTFs into the proposed method could be an interesting future research topic.

Appendix

Proof of Proposition 1

The original optimization problem is

$$\arg\min_{\boldsymbol{U}_1} \ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is equivalent to the following problem when data are complete:

$$\arg\min_{\boldsymbol{U}_1} \ \alpha\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2 + (1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is convex. Thus, it can be solved by setting the derivatives to be zeros, that is $\frac{d\Psi}{d\boldsymbol{U}_1}=\mathbf{0}$, where $\Psi=\alpha\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2+(1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2$. This implies $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2)=\mathbf{0}$. According to the communication law of tensor mode multiplication, we have $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X}-(\mathcal{S}\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\times_1\boldsymbol{U}_1^\top\|_F^2)=\mathbf{0}$. Thus, $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X}-\mathcal{S}_{U_1}\times_1\boldsymbol{U}_1^\top\|_F^2)=\mathbf{0}$, where $\mathcal{S}_{U_1}=\mathcal{S}\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top$. Furthermore, we have $\frac{d}{d\boldsymbol{U}_1}(\|\boldsymbol{X}_{(1)}-\boldsymbol{U}_1^\top\cdot\boldsymbol{S}_{U_1(1)}\|_F^2)=\mathbf{0}$ because of the fact that $\|\mathcal{S}\|_F^2=\|\boldsymbol{S}_{(n)}\|_F^2$ and the property of tensor mode multiplication $\mathcal{S}\times_n\boldsymbol{U}=\boldsymbol{U}\cdot\boldsymbol{S}_{(n)}$. By taking the derivative of the Frobenius norm, we have $2(\boldsymbol{X}_{(1)}-\boldsymbol{U}_1^\top\cdot\boldsymbol{S}_{U_1(1)})\cdot(-\boldsymbol{S}_{U_1(1)}^\top)=\mathbf{0}$. Thus, $\boldsymbol{U}_1^\top\cdot\boldsymbol{S}_{U_1(1)}\cdot\boldsymbol{S}_{U_1(1)}^\top=\boldsymbol{X}_{(1)}\cdot\boldsymbol{S}_{U_1(1)}^\top$, which gives that $\boldsymbol{U}_1^\top=\boldsymbol{X}_{(1)}\cdot\boldsymbol{S}_{U_1(1)}^\top\cdot(\boldsymbol{S}_{U_1(1)}\cdot\boldsymbol{S}_{U_1(1)}^\top)^{-1}$. Finally, we have $\boldsymbol{U}_1=(\boldsymbol{X}_{(1)}\cdot\boldsymbol{S}_{U_1(1)}^\top\cdot(\boldsymbol{S}_{U_1(1)}\cdot\boldsymbol{S}_{U_1(1)}^\top)^{-1})^\top$.

Proof of Proposition 2

The original optimization problem is

$$\arg\min_{\boldsymbol{U}_2} \ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is equivalent to the following problem when data are complete:

$$\arg\min_{\boldsymbol{U}_2} \ \alpha\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2 + (1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is convex. Therefore, it can be solved by setting the derivatives to be zeros, that is $\frac{d\Psi}{d\boldsymbol{U}_2}=\mathbf{0}$, where $\Psi=\alpha\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2+(1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2$. This implies $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2)=\mathbf{0}$. According to the communication law of tensor mode multiplication, we have $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X}-(\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_3\boldsymbol{U}_3^\top)\times_2\boldsymbol{U}_2^\top\|_F^2)=\mathbf{0}$. Thus, $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X}-\mathcal{S}_{U_2}\times_2\boldsymbol{U}_2^\top\|_F^2)=\mathbf{0}$, where $\mathcal{S}_{U_2}=\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_3\boldsymbol{U}_3^\top$. Furthermore, we have $\frac{d}{d\boldsymbol{U}_2}(\|\boldsymbol{X}_{(2)}-\boldsymbol{U}_2^\top\cdot\boldsymbol{S}_{U_2(2)}\|_F^2)=\mathbf{0}$ because of the fact that $\|\mathcal{S}\|_F^2=\|\boldsymbol{S}_{(n)}\|_F^2$ and the property of tensor mode multiplication $\mathcal{S}\times_n\boldsymbol{U}=\boldsymbol{U}\cdot\boldsymbol{S}_{(n)}$. By taking the derivative of the Frobenius norm, we have $2(\boldsymbol{X}_{(2)}-\boldsymbol{U}_2^\top\cdot\boldsymbol{S}_{U_2(2)})\cdot(-\boldsymbol{S}_{U_2(2)}^\top)=\mathbf{0}$. Thus, $\boldsymbol{U}_2^\top\cdot\boldsymbol{S}_{U_2(2)}\cdot\boldsymbol{S}_{U_2(2)}^\top=\boldsymbol{X}_{(2)}\cdot\boldsymbol{S}_{U_2(2)}^\top$, which gives that $\boldsymbol{U}_2^\top=\boldsymbol{X}_{(2)}\cdot\boldsymbol{S}_{U_2(2)}^\top\cdot(\boldsymbol{S}_{U_2(2)}\cdot\boldsymbol{S}_{U_2(2)}^\top)^{-1}$. Finally, we have $\boldsymbol{U}_2=(\boldsymbol{X}_{(2)}\cdot\boldsymbol{S}_{U_2(2)}^\top\cdot(\boldsymbol{S}_{U_2(2)}\cdot\boldsymbol{S}_{U_2(2)}^\top)^{-1})^\top$.

Proof of Proposition 3

The original optimization problem is

$$\arg\min_{\boldsymbol{U}_3} \ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is equivalent to the following problem when data are complete:

$$\arg\min_{\boldsymbol{U}_3} \ \alpha\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2 + (1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2$$

which is convex. Therefore, it can be solved by setting the derivatives to be zeros, that is $\frac{d\Psi}{d\boldsymbol{U}_3}=\mathbf{0}$, where $\Psi=\alpha\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2+(1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2$. This implies that $\frac{d}{d\boldsymbol{U}_3}(\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2)=\mathbf{0}$. According to the communication law of tensor mode multiplication, we have $\frac{d}{d\boldsymbol{U}_3}(\|\mathcal{X}-(\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top)\times_3\boldsymbol{U}_3^\top\|_F^2)=\mathbf{0}$. Thus, $\frac{d}{d\boldsymbol{U}_3}(\|\mathcal{X}-\mathcal{S}_{U_3}\times_3\boldsymbol{U}_3^\top\|_F^2)=\mathbf{0}$, where $\mathcal{S}_{U_3}=\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top$. Furthermore, we have $\frac{d}{d\boldsymbol{U}_3}(\|\boldsymbol{X}_{(3)}-\boldsymbol{U}_3^\top\cdot\boldsymbol{S}_{U_3(3)}\|_F^2)=\mathbf{0}$ because of the fact that $\|\mathcal{S}\|_F^2=\|\boldsymbol{S}_{(n)}\|_F^2$ and the property of tensor mode multiplication $\mathcal{S}\times_n\boldsymbol{U}=\boldsymbol{U}\cdot\boldsymbol{S}_{(n)}$. By taking the derivative of the Frobenius norm, we have $2(\boldsymbol{X}_{(3)}-\boldsymbol{U}_3^\top\cdot\boldsymbol{S}_{U_3(3)})\cdot(-\boldsymbol{S}_{U_3(3)}^\top)=\mathbf{0}$. Thus, $\boldsymbol{U}_3^\top\cdot\boldsymbol{S}_{U_3(3)}\cdot\boldsymbol{S}_{U_3(3)}^\top=\boldsymbol{X}_{(3)}\cdot\boldsymbol{S}_{U_3(3)}^\top$, which gives that $\boldsymbol{U}_3^\top=\boldsymbol{X}_{(3)}\cdot\boldsymbol{S}_{U_3(3)}^\top\cdot(\boldsymbol{S}_{U_3(3)}\cdot\boldsymbol{S}_{U_3(3)}^\top)^{-1}$. Finally, we have $\boldsymbol{U}_3=(\boldsymbol{X}_{(3)}\cdot\boldsymbol{S}_{U_3(3)}^\top\cdot(\boldsymbol{S}_{U_3(3)}\cdot\boldsymbol{S}_{U_3(3)}^\top)^{-1})^\top$.

Proof of Proposition 4

The original optimization problem is

$$\arg\min_{S} \ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y}-\mathbf{1}_M\cdot\beta_0-\boldsymbol{S}_{(4)}\cdot\boldsymbol{\beta}_1\|_F^2,$$

which is equivalent to the following problem when data are complete:

$$\arg\min_{\hat{S}} \ \alpha\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2 + (1-\alpha)\|\boldsymbol{y}-\boldsymbol{1}_M\cdot\beta_0-\boldsymbol{S}_{(4)}\cdot\boldsymbol{\beta}_1\|_F^2,$$

which is convex. Thus, it can be solved by setting the derivatives to be zeros, that is $\frac{d\Psi}{d\hat{S}}=\boldsymbol{0}$, where $\Psi=\alpha\|\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top\|_F^2+(1-\alpha)\|\boldsymbol{y}-\boldsymbol{1}_M\cdot\beta_0-\boldsymbol{S}_{(4)}\cdot\boldsymbol{\beta}_1\|_F^2$. According to the connection between Kronecker product and tensor mode multiplication (Kolda 2006), we have $\frac{d}{d\hat{S}}(\alpha\|\boldsymbol{X}_{(4)}-\boldsymbol{S}_{(4)}\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)\|_F^2+(1-\alpha)\|\boldsymbol{y}-\boldsymbol{1}_M\cdot\beta_0-\boldsymbol{S}_{(4)}\cdot\boldsymbol{\beta}_1\|_F^2)=\boldsymbol{0}$. By taking the derivative of the Frobenius norm, we have $2\alpha\cdot[\boldsymbol{X}_{(4)}-\boldsymbol{S}_{(4)}\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)]\cdot[-(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^\top]+2(1-\alpha)\cdot[\boldsymbol{y}-\boldsymbol{1}_M\cdot\beta_0-\boldsymbol{S}_{(4)}\cdot\boldsymbol{\beta}_1]\cdot(-\boldsymbol{\beta}_1^\top)=\boldsymbol{0}$. Thus, $-2\alpha\cdot\boldsymbol{X}_{(4)}\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^\top+2\alpha\cdot\boldsymbol{S}_{(4)}\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^\top+2(1-\alpha)(\boldsymbol{y}-\boldsymbol{1}_M\cdot\beta_0)\cdot(-\boldsymbol{\beta}_1^\top)+2(1-\alpha)\cdot(\boldsymbol{S}_{(4)}\cdot\boldsymbol{\beta}_1\cdot\boldsymbol{\beta}_1^\top)=\boldsymbol{0}$. Finally, we have $\boldsymbol{S}_{(4)}=[\alpha\cdot\boldsymbol{X}_{(4)}\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^\top+(1-\alpha)\cdot(\boldsymbol{y}-\boldsymbol{1}_M\cdot\beta_0)\cdot\boldsymbol{\beta}_1^\top]\cdot[\alpha\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)\cdot(\boldsymbol{U}_3\otimes\boldsymbol{U}_2\otimes\boldsymbol{U}_1)^\top+(1-\alpha)\cdot\boldsymbol{\beta}_1\cdot\boldsymbol{\beta}_1^\top]^{-1}$.

Proof of Lemma 1

Let $\boldsymbol{a}_m\in\mathbb{R}^{1\times N}$ denote the mth row of matrix $\boldsymbol{A}\in\mathbb{R}^{M\times N}$ and $\boldsymbol{b}_m\in\mathbb{R}^{1\times P}$ denote the mth row of matrix $\boldsymbol{B}\in\mathbb{R}^{M\times P}$, $m=1,\dots,M$, and then we have

$$\boldsymbol{A}-\boldsymbol{BC}=[(\boldsymbol{a}_1-\boldsymbol{b}_1\boldsymbol{C})^\top,\dots,(\boldsymbol{a}_M-\boldsymbol{b}_M\boldsymbol{C})^\top]^\top.$$

Based on the definition of Frobenius norm, the original objective function in Lemma 1 can be transformed as follows:

$$\|\boldsymbol{A}-\boldsymbol{BC}\|_F^2=\|[(\boldsymbol{a}_1-\boldsymbol{b}_1\boldsymbol{C})^\top,\dots,(\boldsymbol{a}_M-\boldsymbol{b}_M\boldsymbol{C})^\top]^\top\|_F^2=\sum_{m=1}^M\|\boldsymbol{a}_m-\boldsymbol{b}_m\boldsymbol{C}\|_F^2.$$

Therefore, we have

$$\arg\min_{\boldsymbol{B}}\|\boldsymbol{A}-\boldsymbol{BC}\|_F^2=\arg\min_{\{\boldsymbol{b}_m\}_{m=1}^M}\sum_{m=1}^M\|\boldsymbol{a}_m-\boldsymbol{b}_m\boldsymbol{C}\|_F^2.,$$

where $\boldsymbol{B}=[\boldsymbol{b}_1^\top,\dots,\boldsymbol{b}_M^\top]^\top$. Therefore, to solve the original objective function, we can simply solve the following M sub problems:

$$\arg\min_{\boldsymbol{b}_m}\|\boldsymbol{a}_m-\boldsymbol{b}_m\boldsymbol{C}\|_F^2,\quad m=1,\dots,M.$$

Proof of Proposition 5

The original optimization problem is

$$\arg\min_{\boldsymbol{U}_1}\ \alpha\|\mathcal{P}_\Omega(\mathcal{X}-\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\|_F^2+(1-\alpha)\|\boldsymbol{y}-\boldsymbol{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is equivalent to the following problem when data are missing,

$$\arg\min_{\boldsymbol{U}_1}\ \alpha\|\mathcal{X}-(\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\odot\text{logic}(\mathcal{X})\|_F^2+(1-\alpha)\|\boldsymbol{y}-\boldsymbol{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

where $\odot$ is the inner product, and logic($\mathcal{X}$) denotes the logical value of $\mathcal{X}$, that is, if an entry is observed, its logical value is 1; otherwise, it is 0. Because the problem is convex, it can be solved by setting the derivatives to be zeros, that is, $\frac{d\Psi}{d\boldsymbol{U}_1}=\boldsymbol{0}$, where $\Psi=\alpha\|\mathcal{X}-(\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\odot\text{logic}(\mathcal{X})\|_F^2+(1-\alpha)\|\boldsymbol{y}-\boldsymbol{1}_M\cdot\tilde{\beta}_0-\boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2$. This implies that $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X}-(\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\odot\text{logic}(\mathcal{X})\|_F^2)=\boldsymbol{0}$. According to the communication law of tensor mode multiplication, we have $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X}-[(\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top)\times_3\boldsymbol{U}_3^\top]\odot\text{logic}(\mathcal{X})\|_F^2)=\boldsymbol{0}$. Thus, $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X}-(\mathcal{S}_{U_1}\times_1\boldsymbol{U}_1^\top)\odot\text{logic}(\mathcal{X})\|_F^2)=\boldsymbol{0}$, where $\mathcal{S}_{U_1}=\mathcal{S}\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top$. Furthermore, we have $\frac{d}{d\boldsymbol{U}_1}(\|\boldsymbol{X}_{(1)}-(\boldsymbol{U}_1^\top\cdot\boldsymbol{S}_{U_1(1)})\odot\text{logic}(\boldsymbol{X}_{(1)})\|_F^2)=\boldsymbol{0}$ because $\|\mathcal{S}\|_F^2=\|\boldsymbol{S}_{(n)}\|_F^2$ and $\mathcal{S}\times_n\boldsymbol{U}=\boldsymbol{U}\cdot\boldsymbol{S}_{(n)}$.

Figure A.1 shows the pattern of mode-1 matricization of the 4D tensor $\mathcal{X}$ when it has missing entries whose indices can be denoted by a set $\Omega\subseteq\{(i_1,i_2,i_3,m),1\le i_1\le I_1,1\le i_2\le I_2,1\le i_3\le I_3,1\le m\le M\}$. Based on Lemma 1, we can sequentially optimize each column of $\boldsymbol{U}_1$.

Figure A.1. (Color online) An Illustration of the Data Missing Pattern in Proposition 5 (Stripes Representing Available Entries)

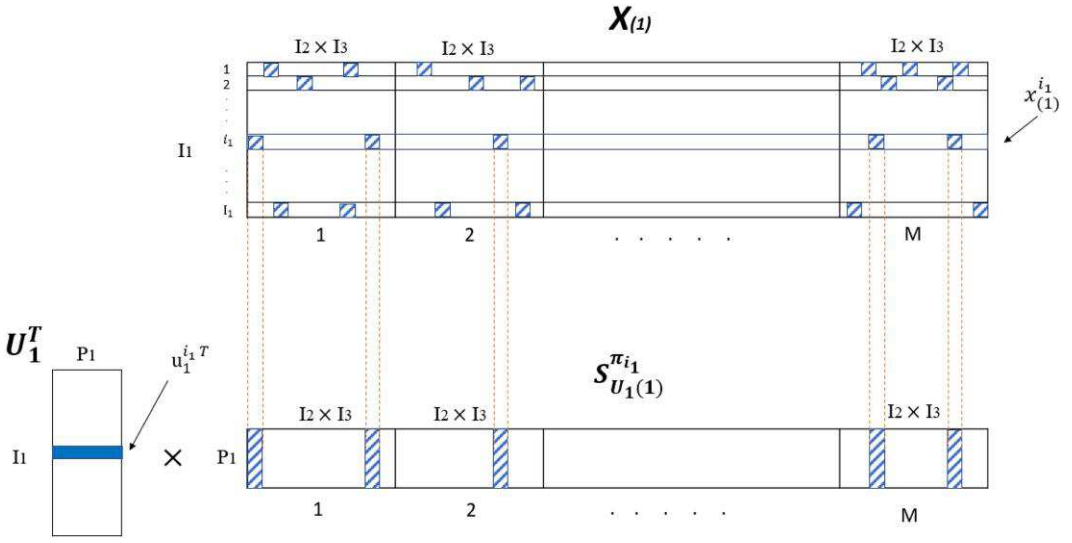
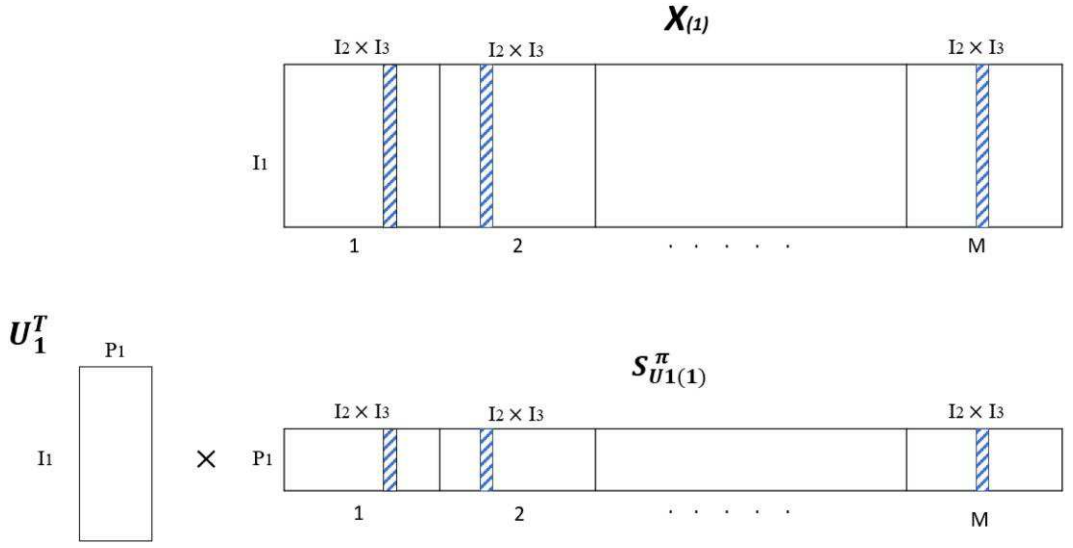


Figure A.2. (Color online) An Illustration of the Data Missing Pattern in Proposition 6 (Stripes Representing Available Columns)



Specifically, we denote the i_1th row in $\boldsymbol{U}_1^\top$ as $\boldsymbol{u}_1^{i_1\top}$ (blue solid row of $\boldsymbol{U}_1^\top$ in Figure A.1). The available entries in the i_1th row of $\boldsymbol{X}_{(1)}$ are denoted as $\boldsymbol{x}_{(1)}^{i_1,\pi_{i_1}}$ (blue striped squares of $\boldsymbol{x}_{(1)}^{i_1}$ in Figure A.1). In $\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}}$, we choose the columns whose indices are the same as those of the available entries of $\boldsymbol{x}_{(1)}^{i_1}$ (blue striped columns of $\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}}$ in Figure A.1). As a result, we have $\frac{d}{d\boldsymbol{u}_1^{i_1}}(\Sigma_{i_1=1}^{I_1}\|\boldsymbol{x}_{(1)}^{i_1,\pi_{i_1}} - (\boldsymbol{u}_1^{i_1\top}\cdot\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}})\|_F^2) = \boldsymbol{0}$. Because we only take the derivative of $\boldsymbol{u}_1^{i_1}$, we have $\frac{d}{d\boldsymbol{u}_1^{i_1}}(\|\boldsymbol{x}_{(1)}^{i_1,\pi_{i_1}} - (\boldsymbol{u}_1^{i_1\top}\cdot\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}})\|_F^2) = \boldsymbol{0}$. By taking the derivative of the Frobenius norm, we have $2(\boldsymbol{x}_{(1)}^{i_1,\pi_{i_1}} - \boldsymbol{u}_1^{i_1\top}\cdot\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}})\cdot(-\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}\top}) = \boldsymbol{0}$. Thus, $\boldsymbol{u}_1^{i_1\top}\cdot\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}}\cdot\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}\top} = \boldsymbol{x}_{(1)}^{i_1,\pi_{i_1}}\cdot\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}\top}$, which gives that $\boldsymbol{u}_1^{i_1} = (\boldsymbol{x}_{(1)}^{i_1,\pi_{i_1}}\cdot\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}\top}\cdot(\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}}\cdot\boldsymbol{S}_{U_1(1)}^{\pi_{i_1}\top})^{-1})^\top$.

Proof of Proposition 6

The original optimization problem is

$$\arg\min_{\boldsymbol{u}_1}\ \alpha\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M\cdot\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is equivalent to the following problem when data are incomplete,

$$\arg\min_{\boldsymbol{u}_1}\ \alpha\|\mathcal{X} - (\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\odot\mathrm{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M\cdot\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

Where $\odot$ is the inner product, and $\mathrm{logic}(\mathcal{X})$ denotes the logical value of $\mathcal{X}$. Because the problem is convex, it can be solved by setting the derivatives to be zeros, that is, $\frac{d\Psi}{d\boldsymbol{U}_1} = \boldsymbol{0}$, where $\Psi = \alpha\|\mathcal{X} - (\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\odot\mathrm{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M\cdot\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2$. This implies that $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X} - (\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\odot\mathrm{logic}(\mathcal{X})\|_F^2) = \boldsymbol{0}$. According to the communication law of tensor mode multiplication, we have $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X} - [(\mathcal{S}\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\times_1\boldsymbol{U}_1^\top]\odot\mathrm{logic}(\mathcal{X})\|_F^2) = \boldsymbol{0}$. Thus, $\frac{d}{d\boldsymbol{U}_1}(\|\mathcal{X} - (\mathcal{S}_{U_1}\times_1\boldsymbol{U}_1^\top)\odot\mathrm{logic}(\mathcal{X})\|_F^2) = \boldsymbol{0}$, where $\mathcal{S}_{U_1} = \mathcal{S}\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top$. Furthermore, we have $\frac{d}{d\boldsymbol{U}_1}(\|\boldsymbol{X}_{(1)} - (\boldsymbol{U}_1^\top\cdot\boldsymbol{S}_{U_1(1)})\odot\mathrm{logic}(\boldsymbol{X}_{(1)})\|_F^2) = \boldsymbol{0}$ because of the fact that $\|\mathcal{S}\|_F^2 = \|\boldsymbol{S}_{(n)}\|_F^2$ and $\mathcal{S}\times_n\boldsymbol{U} = \boldsymbol{U}\cdot\boldsymbol{S}_{(n)}$ (a property of tensor mode multiplication).

As discussed earlier, for applications with missing images, the indices of tensor $\mathcal{X}$'s missing entries can be denoted as $\Omega\subseteq\{(:,:,i_3,m), 1\le i_3\le I_3, 1\le m\le M\}$, where ":" denotes all of the indices in a dimension. As a result, it can be easily shown that $\mathcal{X}$'s mode-1 matricization $\boldsymbol{X}_{(1)}$ has missing columns (see Figure A.2 for an illustration). Let π be the set consisting of the indices of available columns in $\boldsymbol{X}_{(1)}$, and then we need to solve $\frac{d}{d\boldsymbol{U}_1}(\|\boldsymbol{X}_{(1)}^\pi - \boldsymbol{U}_1^\top\cdot\boldsymbol{S}_{U_1(1)}^\pi\|_F^2) = \boldsymbol{0}$, where $\boldsymbol{S}_{U_1(1)}^\pi$ denotes a matrix constituting the π columns of $\boldsymbol{S}_{U_1(1)}$. Thus, we have $2(\boldsymbol{X}_{(1)}^\pi - \boldsymbol{U}_1^\top\cdot\boldsymbol{S}_{U_1(1)}^\pi)\cdot(-\boldsymbol{S}_{U_1(1)}^{\pi\top}) = \boldsymbol{0}$. Thus, $\boldsymbol{U}_1^\top\cdot\boldsymbol{S}_{U_1(1)}^\pi\cdot\boldsymbol{S}_{U_1(1)}^{\pi\top} = \boldsymbol{X}_{(1)}^\pi\cdot\boldsymbol{S}_{U_1(1)}^{\pi\top}$, which gives that $\boldsymbol{U}_1^\top = \boldsymbol{X}_{(1)}^\pi\cdot\boldsymbol{S}_{U_1(1)}^{\pi\top}\cdot(\boldsymbol{S}_{U_1(1)}^\pi\cdot\boldsymbol{S}_{U_1(1)}^{\pi\top})^{-1}$. This yields the solution $\boldsymbol{U}_1 = (\boldsymbol{X}_{(1)}^\pi\cdot\boldsymbol{S}_{U_1(1)}^{\pi\top}\cdot(\boldsymbol{S}_{U_1(1)}^\pi\cdot\boldsymbol{S}_{U_1(1)}^{\pi\top})^{-1})^\top$.

Proof of Proposition 7

The original optimization problem is

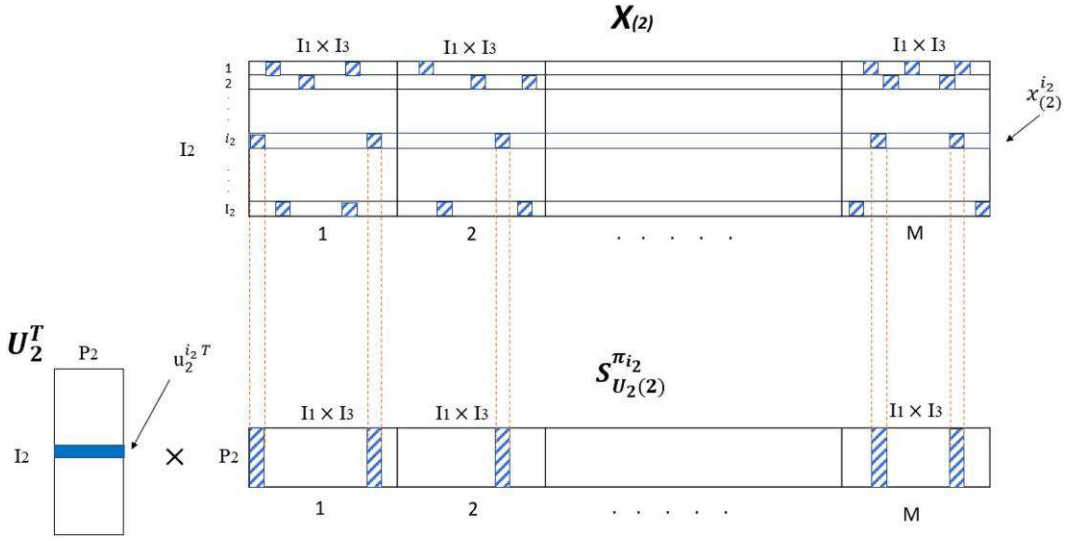
$$\arg\min_{\boldsymbol{u}_2}\ \alpha\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M\cdot\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is equivalent to the following problem when data are missing,

$$\arg\min_{\boldsymbol{u}_2}\ \alpha\|\mathcal{X} - (\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\odot\mathrm{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M\cdot\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2,$$

where $\odot$ is the inner product, and $\mathrm{logic}(\mathcal{X})$ denotes the logical value of $\mathcal{X}$. Because the problem is convex, it can be solved by setting the derivatives to be zeros, i.e., $\frac{d\Psi}{d\boldsymbol{U}_2} = \boldsymbol{0}$, where $\Psi = \alpha\|\mathcal{X} - (\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)\odot\mathrm{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M\cdot\tilde{\beta}_0 - \boldsymbol{S}_{(4)}\cdot\tilde{\boldsymbol{\beta}}_1\|_F^2$. This implies that $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X} - (\mathcal{S}\times_1\boldsymbol{U}_1^\top\times_2\boldsymbol{U}_2^\top\times_3\boldsymbol{U}_3^\top)$

Figure A.3. (Color online) An Illustration of the Data Missing Pattern in Proposition 7 (Stripes Representing Available Entries)



$\odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$. According to the communication law of tensor mode multiplication, we have $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X} - [(\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_3 \boldsymbol{U}_3^\top) \times_2 \boldsymbol{U}_2^\top] \odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$. Thus, $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X} - (\mathcal{S}_{U_2} \times_2 \boldsymbol{U}_2^\top) \odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$, where $\mathcal{S}_{U_2} = \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_3 \boldsymbol{U}_3^\top$. Furthermore, we have $\frac{d}{d\boldsymbol{U}_2}(\|\boldsymbol{X}_{(2)} - (\boldsymbol{U}_2^\top \cdot \boldsymbol{S}_{U_2(2)}) \odot \text{logic}(\boldsymbol{X}_{(2)})\|_F^2) = \mathbf{0}$ because $\|\mathcal{S}\|_F^2 = \|\boldsymbol{S}_{(n)}\|_F^2$ and $\mathcal{S} \times_n \boldsymbol{U} = \boldsymbol{U} \cdot \boldsymbol{S}_{(n)}$.

Figure A.3 shows the pattern of mode-2 matricization of the 4D tensor $\mathcal{X}$ when it has missing entries whose indices can be denoted by a set $\Omega \subseteq \{(i_1, i_2, i_3, m), 1 \le i_1 \le I_1, 1 \le i_2 \le I_2, 1 \le i_3 \le I_3, 1 \le m \le M\}$. Based on Lemma 1, we can sequentially optimize each column of $\boldsymbol{U}_2$.

Specifically, we denote the i_2th row in $\boldsymbol{U}_2^\top$ as $\boldsymbol{u}_2^{i_2\top}$ (blue solid row of $\boldsymbol{U}_2^\top$ in Figure A.3). The available entries in the i_2th row of $\boldsymbol{X}_{(2)}$ are denoted as $\boldsymbol{x}_{(2)}^{i_2,\pi_{i_2}}$ (blue striped squares of $\boldsymbol{x}_{(2)}^{i_2}$ in Figure A.3). In $\boldsymbol{S}_{U_2(2)}^{\pi_{i_2}}$, we choose the columns whose indices are the same as those of the available entries in $\boldsymbol{x}_{(2)}^{i_2}$ (blue striped columns of $\boldsymbol{S}_{U_2(2)}^{\pi_{i_2}}$ in Figure A.3). As a result, we have $\frac{d}{d\boldsymbol{u}_2^{i_2}}(\sum_{i_2=1}^{I_2} \|\boldsymbol{x}_{(2)}^{i_2,\pi_{i_2}} - (\boldsymbol{u}_2^{i_2\top} \cdot \boldsymbol{S}_{U_2(2)}^{\pi_{i_2}})\|_F^2) = \mathbf{0}$. Because on the derivative of $\boldsymbol{u}_2^{i_2}$ is taken, we have $\frac{d}{d\boldsymbol{u}_2^{i_2}}(\|\boldsymbol{x}_{(2)}^{i_2,\pi_{i_2}} - (\boldsymbol{u}_2^{i_2\top} \cdot \boldsymbol{S}_{U_2(2)}^{\pi_{i_2}})\|_F^2) = \mathbf{0}$. By taking the derivative of the Frobenius norm, we have $2(\boldsymbol{x}_{(2)}^{i_2,\pi_{i_2}} - \boldsymbol{u}_2^{i_2\top} \cdot \boldsymbol{S}_{U_2(2)}^{\pi_{i_2}}) \cdot (-\boldsymbol{S}_{U_2(2)}^{\pi_{i_2}\top}) = \mathbf{0}$. Thus, $\boldsymbol{u}_2^{i_2\top} \cdot \boldsymbol{S}_{U_2(2)}^{\pi_{i_2}} \cdot \boldsymbol{S}_{U_2(2)}^{\pi_{i_2}\top} = \boldsymbol{x}_{(2)}^{i_2,\pi_{i_2}} \cdot \boldsymbol{S}_{U_2(2)}^{\pi_{i_2}\top}$, which gives that $\boldsymbol{u}_2^{i_2} = (\boldsymbol{x}_{(2)}^{i_2,\pi_{i_2}} \cdot \boldsymbol{S}_{U_2(2)}^{\pi_{i_2}\top} \cdot (\boldsymbol{S}_{U_2(2)}^{\pi_{i_2}} \cdot \boldsymbol{S}_{U_2(2)}^{\pi_{i_2}\top})^{-1})^\top$.

Proof of Proposition 8

The original optimization problem is

$$\arg\min_{\boldsymbol{U}_2} \ \alpha\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \mathbf{1}_M \cdot \tilde{\beta}_0 - \boldsymbol{S}_{(4)} \cdot \tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is equivalent to the following problem when data are incomplete,

$$\arg\min_{\boldsymbol{U}_2} \ \alpha\|\mathcal{X} - (\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \mathbf{1}_M \cdot \tilde{\beta}_0 - \boldsymbol{S}_{(4)} \cdot \tilde{\boldsymbol{\beta}}_1\|_F^2,$$

where $\odot$ is the inner product, and $\text{logic}(\mathcal{X})$ denotes the logical value of $\mathcal{X}$. Because the optimization criterion is convex, it can be solved by setting the derivatives to be zeros, that is, $\frac{d\Psi}{d\boldsymbol{U}_2} = \mathbf{0}$, where $\Psi = \alpha\|\mathcal{X} - (\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \mathbf{1}_M \cdot \tilde{\beta}_0 - \boldsymbol{S}_{(4)} \cdot \tilde{\boldsymbol{\beta}}_1\|_F^2$. This implies that $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X} - (\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$. According to the communication law of tensor mode multiplication, we have $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X} - [(\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_3 \boldsymbol{U}_3^\top) \times_2 \boldsymbol{U}_2^\top] \odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$. As a result, $\frac{d}{d\boldsymbol{U}_2}(\|\mathcal{X} - (\mathcal{S}_{U_2} \times_2 \boldsymbol{U}_2^\top) \odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$, where $\mathcal{S}_{U_2} = \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_3 \boldsymbol{U}_3^\top$. Furthermore, we have $\frac{d}{d\boldsymbol{U}_2}(\|\boldsymbol{X}_{(2)} - (\boldsymbol{U}_2^\top \cdot \boldsymbol{S}_{U_2(2)}) \odot \text{logic}(\boldsymbol{X}_{(2)})\|_F^2) = \mathbf{0}$ because $\|\mathcal{S}\|_F^2 = \|\boldsymbol{S}_{(n)}\|_F^2$ and $\mathcal{S} \times_n \boldsymbol{U} = \boldsymbol{U} \cdot \boldsymbol{S}_{(n)}$.

It can be easily shown that $\mathcal{X}$'s mode-2 matricization $\boldsymbol{X}_{(2)}$ has missing columns as well (see Figure A.4 for an illustration). Therefore, similar to the proof of Proposition 9, we have $\frac{d}{d\boldsymbol{U}_2}(\|\boldsymbol{X}_{(2)}^\pi - \boldsymbol{U}_2^\top \cdot \boldsymbol{S}_{U_2(2)}^\pi\|_F^2) = \mathbf{0}$, where π denotes the indices of available columns in $\boldsymbol{X}_{(2)}$, $\boldsymbol{S}_{U_2(2)}^\pi$ denotes a matrix constituting the π columns of $\boldsymbol{S}_{U_2(2)}$. As a result, we have $2(\boldsymbol{X}_{(2)}^\pi - \boldsymbol{U}_2^\top \cdot \boldsymbol{S}_{U_2(2)}^\pi) \cdot (-\boldsymbol{S}_{U_2(2)}^{\pi\top}) = \mathbf{0}$. Thus, $\boldsymbol{U}_2^\top \cdot \boldsymbol{S}_{U_2(2)}^\pi \cdot \boldsymbol{S}_{U_2(2)}^{\pi\top} = \boldsymbol{X}_{(2)}^\pi \cdot \boldsymbol{S}_{U_2(2)}^{\pi\top}$, which gives that $\boldsymbol{U}_2^\top = \boldsymbol{X}_{(2)}^\pi \cdot \boldsymbol{S}_{U_2(2)}^{\pi\top} \cdot (\boldsymbol{S}_{U_2(2)}^\pi \cdot \boldsymbol{S}_{U_2(2)}^{\pi\top})^{-1}$. This yields the analytical solution $\boldsymbol{U}_2 = (\boldsymbol{X}_{(2)}^\pi \cdot \boldsymbol{S}_{U_2(2)}^{\pi\top} \cdot (\boldsymbol{S}_{U_2(2)}^\pi \cdot \boldsymbol{S}_{U_2(2)}^{\pi\top})^{-1})^\top$.

Proof of Proposition 9

The original optimization problem is

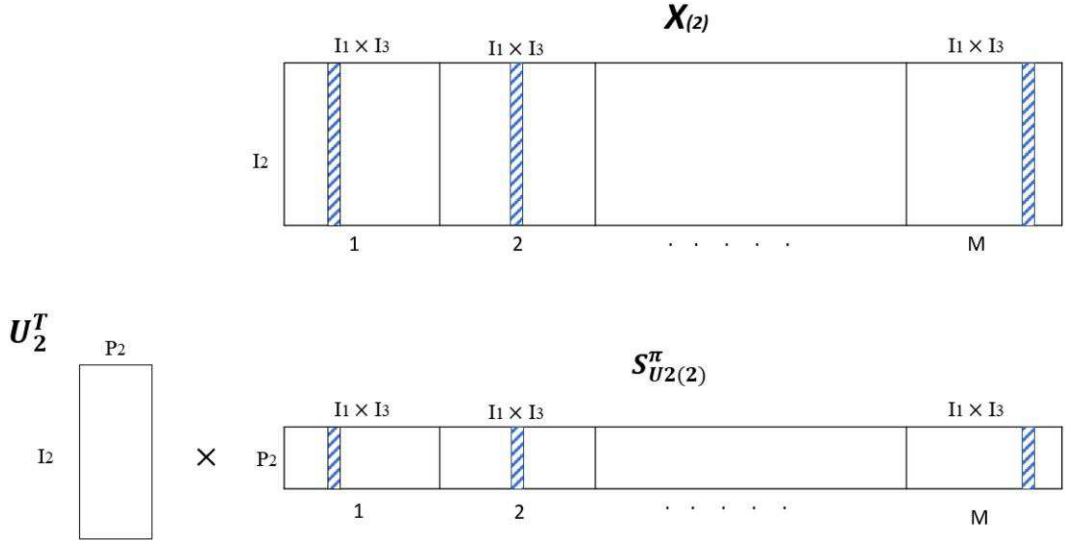
$$\arg\min_{\boldsymbol{U}_3} \ \alpha\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \mathbf{1}_M \cdot \tilde{\beta}_0 - \boldsymbol{S}_{(4)} \cdot \tilde{\boldsymbol{\beta}}_1\|_F^2,$$

which is equivalent to the following problem when data are missing,

$$\arg\min_{\boldsymbol{U}_3} \ \alpha\|\mathcal{X} - (\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \mathbf{1}_M \cdot \tilde{\beta}_0 - \boldsymbol{S}_{(4)} \cdot \tilde{\boldsymbol{\beta}}_1\|_F^2,$$

Where $\odot$ is the inner product, and $\text{logic}(\mathcal{X})$ denotes the logical value of $\mathcal{X}$. Because the problem is convex, it can be solved by

Figure A.4. (Color online) An Illustration of the Data Missing Pattern in Proposition 8 (Stripes Representing Available Columns)



setting the derivatives to be zeros, i.e., $\frac{d\Psi}{d\boldsymbol{U}_3} = \mathbf{0}$, where $\Psi = \alpha\|\mathcal{X} - (\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \mathbf{1}_M \cdot \tilde{\beta}_0 - \boldsymbol{S}_{(4)} \cdot \tilde{\boldsymbol{\beta}}_1\|_F^2$. This implies that $\frac{d}{d\boldsymbol{U}_3}(\|\mathcal{X} - (\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$. According to the communication law of tensor mode multiplication, we have $\frac{d}{d\boldsymbol{U}_3}(\|\mathcal{X} - [(\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top) \times_3 \boldsymbol{U}_3^\top] \odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$. Thus, $\frac{d}{d\boldsymbol{U}_3}(\|\mathcal{X} - (\mathcal{S}_{U_3} \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2) = \mathbf{0}$, where $\mathcal{S}_{U_3} = \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top$. Furthermore, we have $\frac{d}{d\boldsymbol{U}_3}(\|\boldsymbol{X}_{(3)} - (\boldsymbol{U}_3^\top \cdot \boldsymbol{S}_{U_3(3)}) \odot \text{logic}(\boldsymbol{X}_{(3)})\|_F^2) = \mathbf{0}$ because $\|\mathcal{S}\|_F^2 = \|\boldsymbol{S}_{(n)}\|_F^2$ and $\mathcal{S} \times_n \boldsymbol{U} = \boldsymbol{U} \cdot \boldsymbol{S}_{(n)}$.

Figure A.5 shows the pattern of mode-3 matricization of the 4D tensor $\mathcal{X}$ when it has missing entries whose indices can be denoted by a set $\Omega \subseteq \{(i_1, i_2, i_3, m), 1 \le i_1 \le I_1, 1 \le i_2 \le I_2, 1 \le i_3 \le I_3, 1 \le m \le M\}$. Based on Lemma 1, we can sequentially optimize each column of $\boldsymbol{U}_3$. The i_3th row in $\boldsymbol{U}_3^\top$ is denoted as $\boldsymbol{u}_3^{i_3\top}$ (blue solid row of $\boldsymbol{U}_3^\top$ in Figure A.5). The available entries in the i_3th row of $\boldsymbol{X}_{(3)}$ are denoted as $\boldsymbol{x}_{(3)}^{i_3,\pi_{i_3}}$ (blue striped squares of $\boldsymbol{x}_{(3)}^{i_3}$ in Figure A.5). In $\boldsymbol{S}_{U_3(3)}^{\pi_{i_3}}$, we choose the columns whose indices are the same as those of the available entries of $\boldsymbol{x}_{(3)}^{i_3}$ (blue striped columns of $\boldsymbol{S}_{U_3(3)}^{\pi_{i_3}}$ in Figure A.5). Thus, we have $\frac{d}{d\boldsymbol{u}_3^{i_3}}(\Sigma_{i_3=1}^{I_3}\|\boldsymbol{x}_{(3)}^{i_3,\pi_{i_3}} - (\boldsymbol{u}_3^{i_3\top} \cdot \boldsymbol{S}_{U_3(3)}^{\pi_{i_3}})\|_F^2) = \mathbf{0}$, which yields $\frac{d}{d\boldsymbol{u}_3^{i_3}}(\|\boldsymbol{x}_{(3)}^{i_3,\pi_{i_3}} - (\boldsymbol{u}_3^{i_3\top} \cdot \boldsymbol{S}_{U_3(3)}^{\pi_{i_3}})\|_F^2) = \mathbf{0}$. By taking the derivative of the Frobenius norm, we have $2(\boldsymbol{x}_{(3)}^{i_3,\pi_{i_3}} - \boldsymbol{u}_3^{i_3\top} \cdot \boldsymbol{S}_{U_3(3)}^{\pi_{i_3}}) \cdot (-\boldsymbol{S}_{U_3(3)}^{\pi_{i_3}\top}) = \mathbf{0}$. Thus, $\boldsymbol{u}_3^{i_3\top} \cdot \boldsymbol{S}_{U_3(3)}^{\pi_{i_3}} \cdot \boldsymbol{S}_{U_3(3)}^{\pi_{i_3}\top} = \boldsymbol{x}_{(3)}^{i_3,\pi_{i_3}} \cdot \boldsymbol{S}_{U_3(3)}^{\pi_{i_3}\top}$, which gives that $\boldsymbol{u}_3^{i_3} = (\boldsymbol{x}_{(3)}^{i_3,\pi_{i_3}} \cdot \boldsymbol{S}_{U_3(3)}^{\pi_{i_3}\top} \cdot (\boldsymbol{S}_{U_3(3)}^{\pi_{i_3}} \cdot \boldsymbol{S}_{U_3(3)}^{\pi_{i_3}\top})^{-1})^\top$.

Proof of Proposition 10

The original optimization problem is

$$\arg\min_{S} \ \alpha\|\mathcal{P}_\Omega(\mathcal{X} - \mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top)\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \mathbf{1}_M \cdot \beta_0 - \boldsymbol{S}_{(4)} \cdot \boldsymbol{\beta}_1\|_F^2,$$

Figure A.5. (Color online) An Illustration of the Data Missing Pattern in Proposition 9 (Stripes Representing Available Entries)

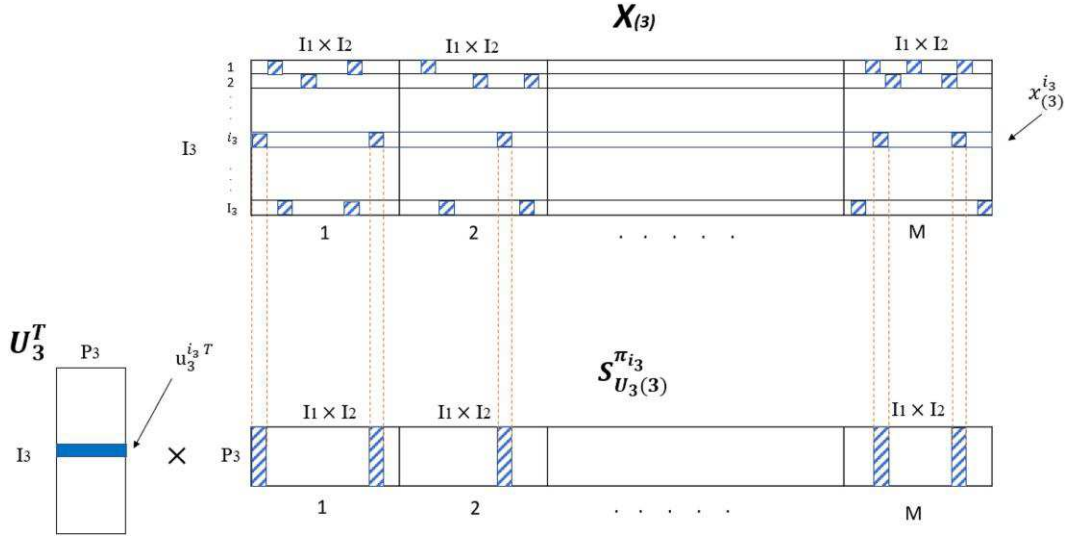
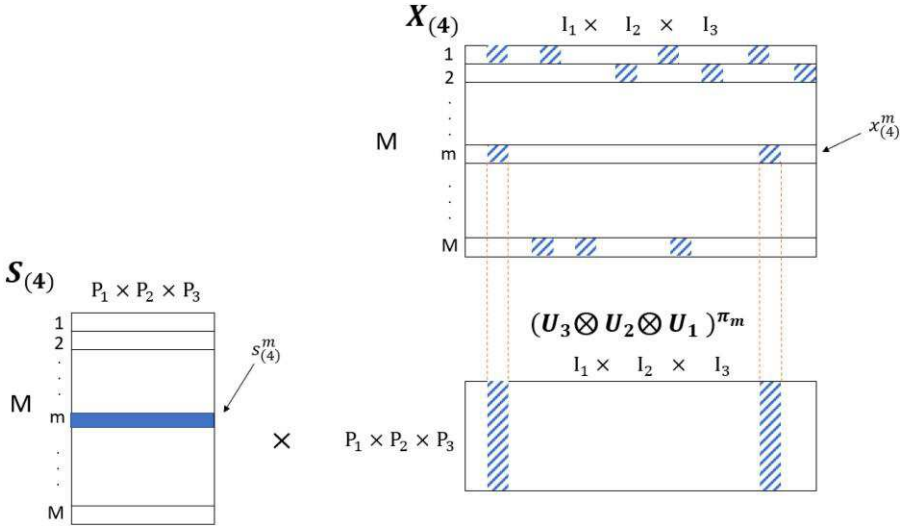


Figure A.6. (Color online) An Illustration of the Data Missing Pattern in Proposition 10 (Stripes Representing Available Entries)



which is equivalent to the following problem when data are missing,

$$\arg\min_{S} \ \alpha\|\mathcal{X} - (\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M \cdot \beta_0 - \boldsymbol{S}_{(4)} \cdot \boldsymbol{\beta}_1\|_F^2$$

where $\odot$ is the inner product, and logic($\mathcal{X}$) denotes the logical value of $\mathcal{X}$. Because the problem is convex, it can be solved by setting the derivative to be zeros–that is $\frac{d\Psi}{d\boldsymbol{S}} = \boldsymbol{0}$, where $\Psi = \alpha\|\mathcal{X} - (\mathcal{S} \times_1 \boldsymbol{U}_1^\top \times_2 \boldsymbol{U}_2^\top \times_3 \boldsymbol{U}_3^\top) \odot \text{logic}(\mathcal{X})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M \cdot \beta_0 - \boldsymbol{S}_{(4)} \cdot \boldsymbol{\beta}_1\|_F^2$. According to the connection between Kronecker product and tensor mode multiplication, we have $\frac{d}{d\boldsymbol{S}}(\alpha\|\boldsymbol{X}_{(4)} - [\boldsymbol{S}_{(4)} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)] \odot \text{logic}(\boldsymbol{X}_{(4)})\|_F^2 + (1-\alpha)\|\boldsymbol{y} - \boldsymbol{1}_M \odot \beta_0 - \boldsymbol{S}_{(4)} \odot \boldsymbol{\beta}_1\|_F^2) = \boldsymbol{0}$.

Figure A.6 shows the pattern of mode-4 matricization of the 4D tensor $\mathcal{X}$ when it has missing entries whose indices can be denoted by a set $\Omega \subseteq \{(i_1, i_2, i_3, m), 1 \le i_1 \le I_1, 1 \le i_2 \le I_2, 1 \le i_3 \le I_3, 1 \le m \le M\}$. Based on Lemma 1, we can sequentially optimize each column of $\boldsymbol{U}_3$.

The mth row in $\boldsymbol{S}_{(4)}$ is denoted as $\boldsymbol{s}^m_{(4)}$ (blue solid row of $\boldsymbol{S}_{(4)}$ in Figure A.6). The available entries in the mth row of $\boldsymbol{X}_{(4)}$ are denoted as $\boldsymbol{x}^{m,\pi_m}_{(4)}$ (blue striped squares of $\boldsymbol{x}^m_{(4)}$ in Figure A.6). In $(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)$, we choose the columns whose indices are the same as those of the available entries of $\boldsymbol{x}^m_{(4)}$ (blue striped columns of $(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m}$ in Figure A.6). Thus, we have $\frac{d}{d\boldsymbol{s}^m_{(4)}}(\alpha\{\Sigma_{m=1}^M\|\boldsymbol{x}^{m,\pi_m}_{(4)} - [\boldsymbol{s}^m_{(4)} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m}]\}\|_F^2 + (1-\alpha)\Sigma_{m=1}^M \|y_m - \beta_0 - \boldsymbol{s}^m_{(4)} \cdot \boldsymbol{\beta}_1\|_F^2) = \boldsymbol{0}$, which yields $\frac{d}{d\boldsymbol{s}^m_{(4)}}(\alpha\{\|\boldsymbol{x}^{m,\pi_m}_{(4)} - [\boldsymbol{s}^m_{(4)} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m}]\}\|_F^2 + (1-\alpha)\|y_m - \beta_0 - \boldsymbol{s}^m_{(4)} \cdot \boldsymbol{\beta}_1\|_F^2) = \boldsymbol{0}$. By taking the derivative of Frobenius norm, we have $2\alpha \cdot [\boldsymbol{x}^{m,\pi_m}_{(4)} - \boldsymbol{s}^m_{(4)} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m}] \cdot [-(\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m \top}] + 2(1-\alpha) \cdot [y_m - \beta_0 - \boldsymbol{s}^m_{(4)} \cdot \boldsymbol{\beta}_1] \cdot (-\boldsymbol{\beta}_1^\top) = \boldsymbol{0}$. Thus, $-2\alpha \cdot \boldsymbol{x}^{m,\pi_m}_{(4)} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m \top} + 2\alpha \cdot \boldsymbol{s}^m_{(4)} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m \top} + 2(1-\alpha)(y_m - \beta_0) \cdot (-\boldsymbol{\beta}_1^\top) + 2(1-\alpha) \cdot (\boldsymbol{s}^m_{(4)} \cdot \boldsymbol{\beta}_1 \cdot \boldsymbol{\beta}_1^\top) = \boldsymbol{0}$, which gives that $\boldsymbol{s}^m_{(4)} = [\alpha \cdot \boldsymbol{x}^{m,\pi_m}_{(4)} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m \top} + (1-\alpha) \cdot (y_m - \beta_0) \cdot \boldsymbol{\beta}_1^\top] \cdot [\alpha \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m} \cdot (\boldsymbol{U}_3 \otimes \boldsymbol{U}_2 \otimes \boldsymbol{U}_1)^{\pi_m \top} + (1-\alpha) \cdot \boldsymbol{\beta}_1 \cdot \boldsymbol{\beta}_1^\top]^{-1}$.

References

Abdi H, Williams LJ (2010) Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* 2(4):433–459.

Aydemir G, Paynabar K (2019) Image-based prognostics using deep learning approach. *IEEE Trans. Industr. Inform.* 16(9):5956–5964.

Bogdanoff JL, Kozin F (1985) *Probabilistic models of cumulative damage(book)* (Wiley-Interscience, New York).

Dong Y, Xia T, Wang D, Fang X, Xi L (2021) Infrared image stream based regressors for contactless machine prognostics. *Mech. Syst. Signal Process.* 154:107592.

Doray L (1994) IBNR reserve under a loglinear location-scale regression model. *Casualty Actuarial Society Forum*, vol. 2 (Casualty Actuarial Society, Arlington, VA), 607–652.

Fang X, Paynabar K, Gebraeel N (2019) Image-based prognostics using penalized tensor regression. *Technometrics.* 61(3):369–384.

Filipović M, Jukić A (2015) Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimens. Syst. Signal Process.* 26(3):677–692.

Gebraeel N, Elwany A, Pan J (2009) Residual life predictions in the absence of prior degradation knowledge. *IEEE Trans. Reliab.* 58(1):106–117.

Gebraeel NZ, Lawley MA, Li R, Ryan JK (2005) Residual-life distributions from component degradation signals: A bayesian approach. *IIE Trans.* 37(6):543–557.

Hastie T, Tibshirani R, Friedman JH (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York).

Hong Y, Meeker WQ (2010) Field-failure and warranty prediction based on auxiliary use-rate information. *Technometrics* 52(2):148–159.

Hong Y, Meeker WQ (2013) Field-failure predictions based on failure-time data with dynamic covariate information. *Technometrics* 55(2):135–149.

Jiang Y, Xia T, Wang D, Fang X, Xi L (2022a) Adversarial regressive domain adaptation approach for infrared thermography-based unsupervised remaining useful life prediction. *IEEE Trans. Indust. Informatics* 18(10):7219–7229.

Jiang Y, Xia T, Wang D, Fang X, Xi L (2022b) Spatiotemporal denoising wavelet network for infrared thermography-based machine

prognostics integrating ensemble uncertainty. *Mech. Syst. Signal Process* 173:109014.

Jiang Y, Xia T, Fang X, Wang D, Pan E, Xi L (2023) Sparse hierarchical parallel residual networks ensemble for infrared image stream-based remaining useful life prediction. *IEEE Trans. Industr. Inform.* Forthcoming.

Kolda TG (2006) Multilinear operators for higher-order decompositions. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM.

Kolda TG, Bader BW (2009) Tensor decompositions and applications. *SIAM Rev.* 51(3):455–500.

Liu K, Gebraeel NZ, Shi J (2013) A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. *IEEE Trans. Autom. Sci. Eng.* 10(3):652–664.

Liu J, Musialski P, Wonka P, Ye J (2012) Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(1):208–220.

Lu H, Plataniotis KN, Venetsanopoulos AN (2008) Mpca: Multilinear principal component analysis of tensor objects. *IEEE Trans. Neural Netw.* 19(1):18–39.

Prautzsch H, Boehm W, Paluszny M (2002) *Bézier and B-spline techniques*, vol. 6 (Springer, Berlin).

Ramsay J, Silverman B (2005) Principal components analysis for functional data. *Functional Data Analysis* (Springer, Berlin), 147–172.

Shu Y, Feng Q, Coit DW (2015) Life distribution analysis based on Lévy subordinators for degradation with random jumps. *Naval Res. Logist.* 62(6):483–492.

Wang F, Wang A, Tang T, Shi J (2022) Sgl-pca: Health index construction with sensor sparsity and temporal monotonicity for mixed high-dimensional signals. *IEEE Trans. Automation Sci. Eng.* 20(1):372–384.

Wang F, Gahrooei MR, Zhong Z, Tang T, Shi J (2021) An augmented regression model for tensors with missing values. *IEEE Trans. Automation Sci. Eng.* 19(4):2968–2984.

Xu Y, Yin W (2013) A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM J. Imaging Sci.* 6(3):1758–1789.

Xu Y, Hao R, Yin W, Su Z (2013) Parallel matrix factorization for low-rank tensor completion. Preprint, submitted March 24, https://arxiv.org/abs/1312.1254.

Yang Z, Baraldi P, Zio E (2021) A multi-branch deep neural network model for failure prognostics based on multimodal data. *J. Manuf. Syst.* 59:42–50.

Ye Z-S, Chen N (2014) The inverse gaussian process as a degradation model. *Technometrics.* 56(3):302–311.

Ye Z-S, Xie M, Tang L-C, Chen N (2014) Semiparametric estimation of gamma processes for deteriorating products. *Technometrics* 56(4):504–513.