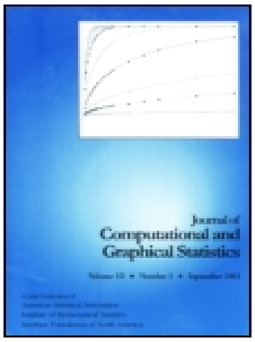

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: http://amstat.tandfonline.com/loi/ucgs20

Supervised Sparse and Functional Principal Component Analysis

Gen Li, Haipeng Shen & Jianhua Z. Huang

To cite this article: Gen Li, Haipeng Shen & Jianhua Z. Huang (2015): Supervised Sparse and Functional Principal Component Analysis, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2015.1064434

To link to this article: http://dx.doi.org/10.1080/10618600.2015.1064434

Supervised Sparse and Functional Principal Component Analysis

Gen Li, Haipeng Shen
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
Jianhua Z. Huang
Department of Statistics, Texas A&M University

Abstract

Principal component analysis (PCA) is an important tool for dimension reduction in multivariate analysis. Regularized PCA methods, such as sparse PCA and functional PCA, have been developed to incorporate special features in many real applications. Sometimes additional variables (referred to as supervision) are measured on the same set of samples, which can potentially drive low-rank structures of the primary data of interest. Classical PCA methods cannot make use of such supervision data. In this paper, we propose a supervised sparse and functional principal component (SupSFPC) framework that can incorporate supervision information to recover underlying structures that are more interpretable. The framework unifies and generalizes several existing methods and flexibly adapts to the practical scenarios at hand. The SupSFPC model is formulated in a hierarchical fashion using latent variables. We develop an efficient modified expectation-maximization algorithm for parameter estimation. We also implement fast data-driven procedures for tuning parameter selection. Our comprehensive simulation and real data examples demonstrate the advantages of SupSFPC. Supplementary materials for this article are available online.

Keywords: Regularized PCA; Supervised dimension reduction; Penalized likelihood; Low rank approximation; Latent variable; SupSFPC.

1 Introduction

Principal component analysis (PCA) has been widely used in multivariate analysis to extract important features in data. Principal component (PC) loadings usually provide useful interpretation of major variations, while PC scores facilitate follow-up statistical analyses such as clustering and regression. It is a powerful tool for dimension reduction, pattern recognition, and visualization for big data.

This paper concerns regularized PCA methods, which impose useful structural regularization on PCA, and have been extensively studied in the literature. Special structures like sparsity and smoothness are imposed on the loading vectors to model high-dimensional data with complex structure. For example, functional PCA is used to model functional observations such as temporal data or spatial data (cf. Rice and Silverman (1991), Silverman et al. (1996), Huang et al. (2008), and references therein). In high dimensional situations where most variables are noise and only a few variables are important, sparse PCA is used to simultaneously select variables and capture major variations (cf. Zou et al. (2006), Shen and Huang (2008), d'Aspremont et al. (2008), and references therein). More recently, some researchers studied two-way extensions of the above one-way regularized PCA methods (see Allen, 2013; Huang et al., 2009; Lee et al., 2010, for example).

Although powerful, the above regularized PCA methods have one limitation in common: they only make use of a single data set, and by default ignore any other measurements collected on the same set of samples. It is now increasingly common that multiple related data sets are available on the same set of samples. In such cases, borrowing information across data sets may lead to recovery of a more interpretable low rank structure. This is especially relevant when the additional measurements, referred to as *supervision information*, can potentially drive underlying patterns within the primary data. For example, in Section 5, we are interested in studying expression patterns of a number of yeast genes over two cell cycles. In addition to the gene expression data, we have extra binding information of transcription factors (TFs) for each gene. Since TFs regulate gene expressions biologically (Lee and Young, 2000; Nikolov and Burley, 1997), using TF binding information as supervision when studying expression patterns can lead to a more inherent and meaningful discovery. Another motivating example considered in Section D of the online supplement concerns daily arrival rates of patients to a hospital emergency room. It is of interest to understand patient arrival patterns to better allocate medical resources. In addition to the primary data, i.e., the arrival rates at different time of day over many days, we want to use the day-of-week index as *supervision* to extract day-of-week specific arrival patterns.

Motivated by these applications, in this paper, we develop a supervised regularized PCA framework that makes use of extra supervision information when doing regularized PCA. We name it the *supervised sparse and functional PCA*, or *SupSFPC*. Supervision, subject to variable selection, directly affects the PC scores, while smooth and sparse structures are imposed on the PC loadings. The SupSFPC framework is very general and flexible. It unifies and generalizes many variants of PCA. In particular, without the supervision, it encompasses regularized PCA methods such as functional PCA and sparse PCA as special cases. Supervision and regularization complement each other under SupSFPC. By smoothing the loading vectors, our method can borrow strength across neighboring variables to reduce noise; with sparsity, the variation of the functional estimate is reduced; supervision indirectly affects the loading vectors to make them more interpretable. Overall, the proposed SupSFPC method can recover an interpretable and accurate low-rank approximation of a primary data set with potential guidance from supervision data.

Recently, Li et al. (2015) studied a supervised version of the standard PCA. However, their method cannot accommodate special features of functional or high dimensional data. Incorporating smoothness and sparsity in such data reduces estimation variability and improves interpretability. Furthermore, their method cannot achieve variable selection of the supervision set: when auxiliary data contain irrelevant information to the low-rank structure of the primary data, it is desirable to eliminate unimportant variables and identify crucial driving factors. For example, in the yeast gene expression application of Section 5, researchers are also interested in identifying TFs that regulate cell cycles. The SupSFPC method addresses the above problems through regularization. The regularization brings new challenges for model fitting, such as optimization of a non-differentiable object function and selection of multiple tuning parameters. We develop an innovative algorithm for SupSFPC by combining an expectation-maximization algorithm with several ascent algorithms. We further alleviate the computational burden by embedding tuning parameter selection in the iterative scheme. Numerical results show high computational efficiency and improvement in interpretability over existing methods.

The rest of the paper is organized as follows. In Section 2, after reviewing the functional PCA model, we propose our new SupSFPC model, followed by the penalized likelihood framework. We then elaborate on connections of the SupSFPC framework to various regularized PCA and supervised PCA methods. In Section 3, we develop a computationally efficient algorithm to estimate the model parameters, and briefly discuss tuning parameter selection. We then demonstrate our method using comprehensive simulation studies in Section 4 and a real data example in Section 5. Additional technical details and numerical studies can be found in the online supplement.

2 Model and Likelihood

In this section, we first review the functional PCA model, and then develop the SupSFPC model and introduce a regularized likelihood approach for the model fitting.

2.1 Functional PCA Model

We assume that $Z_i(s)$ ($i = 1, \cdots, n$) are independent realizations of a smooth random function $Z(s)$ with mean function $\mathbb{E}(Z(s)) = \mu(s)$ and covariance function $\mathrm{cov}(Z(s), Z(s')) = G(s, s')$. The index variable s can represent any continuous measure such as time, spatial location, and so on. Its domain $\mathcal{S}$ is assumed bounded. The covariance function can be decomposed as

$$G(s, s') = \sum_{k=1}^{\infty} d_k V_k(s) V_k(s')$$

where $d_1 \geq d_2 \geq \cdots \geq 0$ are the eigenvalues and $V_k(s)$ ($k = 1, 2, \cdots$) are the corresponding orthogonal unit-norm eigenfunctions. Consequently, by the Karhunen-Loève theorem, the random function $Z(s)$ can be expressed as a linear combination of the eigenfunctions as $Z(s) = \mu(s) + \sum_{k=1}^{\infty} u_k V_k(s)$ where $u_k = \int_{\mathcal{S}} Z(s) V_k(s) \mathrm{d}s$ ($k = 1, 2, \cdots$) are uncorrelated random variables with mean zero and variance d_k. In particular, $Z_i(s)$ has the expression

$$Z_i(s) = \mu(s) + \sum_{k=1}^{\infty} u_{ik} V_k(s), \tag{1}$$

where u_{ik} $(i = 1, \cdots, n)$ are independent realizations of u_k. This is the classical functional PCA model where $(u_{1k}, \cdots, u_{nk})^T$ is the kth score vector, corresponding to the kth loading function $V_k(s)$, $k \geq 1$.

Researchers usually consider the above functional model with measurement errors added, as in for example Yao et al. (2005). Namely, each observed trajectory, denoted by $X_i(s)$, is expressed as

$$X_i(s) = Z_i(s) + e_i(s),$$

where $Z_i(s)$ is the latent random function given in (1), and $e_i(s)$ is a measurement error process that is assumed to be uncorrelated at each point with mean zero and variance $\sigma^2_{\mathbf{e}}$, independently identically distributed (i.i.d.) for different observations.

In practice, the majority of variations in data is contained in the subspace spanned by the first few PC loadings in (1). Namely, the first few layers of the latent function $Z(s)$ dominate and the rest are negligible. Hereafter, we consider the following rank-r functional PCA model:

$$X_i(s) = \mu(s) + \sum_{k=1}^{r} u_{ik} V_k(s) + e_i(s) = \mu(s) + \mathbf{u}_{(i)}^T \mathbf{V}(s) + e_i(s), \tag{2}$$

where $\mathbf{u}_{(i)} = (u_{i1}, \cdots, u_{ir})^T$ is the $r\times1$ score vector for the ith observation, and $\mathbf{V}(s) = (V_1(s), \cdots, V_r(s))^T$ is the collection of r loading functions. In particular, the finite linear combination $\mathbf{u}_{(i)}^T\mathbf{V}(s)$ is referred to as the low-rank approximation of the ith demeaned observation $X_i(s) - \mu(s)$.

2.2 SupSFPC Model

Let $X_i(s)$ be the ith functional observation from Model (2). Let $\mathbf{y}_{(i)}$ be an $q \times 1$ vector containing q auxiliary variables for the ith observation. We assume that $\mathbf{y}_{(i)}$, the supervision data, drives the low-rank structure of $X_i(s)$, the primary data, by directly affecting its PC score vector $\mathbf{u}_{(i)}$ in Model (2). In particular, we propose the following multivariate linear model for the scores:

$$\mathbf{u}_{(i)} = \boldsymbol{\beta}_0 + \mathbf{B}^T\mathbf{y}_{(i)} + \mathbf{f}_{(i)} \tag{3}$$

where $\boldsymbol{\beta}_0$ is an $r \times 1$ intercept vector, $\mathbf{B}$ is a $q \times r$ coefficient matrix with the rows corresponding to the supervision variables and the columns corresponding to the PC scores, and $\mathbf{f}_{(i)}$ is an independent realization of an $r \times 1$ random vector with mean zero and unknown covariance $\boldsymbol{\Sigma}_{\mathbf{f}}$. For example, in the genetic application of Section 5, $X_i(s)$ denotes the gene expression profile of the ith sample, while $\mathbf{y}_{(i)}$ are the corresponding transcription factors.

Model (3) consists of a fixed term $\boldsymbol{\beta}_0 + \mathbf{B}^T\mathbf{y}_{(i)}$ and a random term $\mathbf{f}_{(i)}$. The fixed term captures the variations in $\mathbf{u}_{(i)}$ that can be explained by the supervision data $\mathbf{y}_{(i)}$. The random term effectively collects the leftover variations driven by other (unknown) factors. Model (3) is flexible enough to adapt to different situations including those where the supervision information is indeed redundant, as we discuss later in Section 2.3.

Combining (2) and (3), we obtain the *supervised functional PCA model*. In particular, we substitute $\mathbf{u}_{(i)}$ in (2) with (3) and get the following equivalent expression of the model:

$$\begin{aligned} X_i(s) &= \mu(s) + (\boldsymbol{\beta}_0 + \mathbf{B}^T\mathbf{y}_{(i)} + \mathbf{f}_{(i)})^T\mathbf{V}(s) + e_i(s) \\ &= \left[\mu(s) + \boldsymbol{\beta}_0^T\mathbf{V}(s)\right] + \mathbf{y}_{(i)}^T\mathbf{B}\mathbf{V}(s) + \left[\mathbf{f}_{(i)}^T\mathbf{V}(s) + e_i(s)\right]. \end{aligned} \tag{4}$$

The first term, $\mu(s) + \boldsymbol{\beta}_0^T\mathbf{V}(s)$, is an intercept term. Without loss of generality, we assume that $X_i(s)$ and $\mathbf{y}_{(i)}$ are centered at each variable so we can omit this intercept term. The second term, $\mathbf{y}_{(i)}^T\mathbf{B}\mathbf{V}(s)$, is a fixed term that incorporates the supervision information. The third term, $\mathbf{f}_{(i)}^T\mathbf{V}(s) + e_i(s)$, is a random term, with the covariance function being $\mathbf{V}(s)^T\boldsymbol{\Sigma}_{\mathbf{f}}\mathbf{V}(s') + \sigma_{\mathbf{e}}^2\delta(s - s')$, where $\delta(\cdot)$ is the Dirac delta function. The recovery of the low-rank structure $\mathbf{y}_{(i)}^T\mathbf{B}\mathbf{V}(s) + \mathbf{f}_{(i)}^T\mathbf{V}(s)$ is of primary interest in dimension reduction.

We further generalize Model (4) by assuming that $\mathbf{B}$ and $\mathbf{V}(s)$ are potentially sparse. Consequently, we name Model (4) the *supervised sparse and functional PCA model*, or the SupSFPC model. Recall that $\mathbf{B}$ is a coefficient matrix to incorporate supervision. Sparsity on $\mathbf{B}$ can effectively identify auxiliary variables that do not provide supervision to the low-rank structure of the primary data. In particular, when $\mathbf{B}$ is a zero matrix, all auxiliary variables are irrelevant to the primary data, and the SupSFPC model reduces to the functional PCA model (2). The loading functions in $\mathbf{V}(s)$ can be sparse as well, in the sense that the support of each loading function may not be the entire domain $\mathcal{S}$. Similar to James et al. (2009) where the authors study a regression model with a (potentially sparse) functional predictor, we remark that sparse functions facilitate model interpretations by removing unnatural wiggles around zero. Overall, sparsity is usually a desirable (and sometimes necessary) feature in practice, especially when analyzing high-dimensional data.

As it stands, Model (4) is not identifiable. Because, for any $r \times r$ orthogonal matrix $\mathbf{Q}$, we have $\mathbf{B}\mathbf{Q}\mathbf{Q}^T\mathbf{V}(s) = \mathbf{B}\mathbf{V}(s)$ and $\mathbf{f}_{(i)}^T\mathbf{Q}\mathbf{Q}^T\mathbf{V}(s) = \mathbf{f}_{(i)}^T\mathbf{V}(s)$. Moreover, the columns of $\mathbf{B}$ and the entries of $\mathbf{V}(s)$ and $\mathbf{f}_{(i)}$ are subject to scale and order shifts. To rule out this kind of ambiguity, we impose the following identifiability constraints:

(1) The loading functions in $\mathbf{V}(s)$ form an orthonormal basis, i.e., $\int_{\mathcal{S}} V_i(s)V_j(s)\mathrm{d}s = \delta_{ij}$, where δ_{ij} is the Kronecker delta;

(2) The covariance matrix $\mathbf{\Sigma_f}$ is diagonal with distinct positive eigenvalues;

(3) The diagonal values of $\mathbf{\Sigma_f}$ are strictly decreasing.

The orthonormality constraint of the loading functions, also used in the functional PCA model (1), facilitates interpretation and rules out scale shift. The diagonality of the covariance matrix with distinct eigenvalues prevents random rotations. The order of the eigenvalues of $\mathbf{\Sigma_f}$ determines the order of the loading functions in $\mathbf{V}(s)$ and the columns in $\mathbf{B}$. We remark that under the above conditions, the loading functions carry explicit interpretations: the first loading captures the direction where variation in the data from unknown sources is maximized; subsequent loadings are orthogonal to the previous ones and sequentially maximize variations from unknown sources. This is similar with functional PCA where there is no supervision and all variations are from unknown sources.

2.3 Penalized Likelihood

In reality, typically we do not observe an entire function but rather at discrete sampling points. In particular, we assume that there are p sampling points in domain $\mathcal{S}$ indexed by $s_1, \cdots, s_p$, which may not be evenly spaced. For notational simplicity, without special notice we generally use $i = 1, \cdots, n$ to index samples, use $j = 1, \cdots, p$ to index discretized points, and use $k = 1, \cdots, r$ to index PC layers.

The discrete observations of the functional data $X_i(s)$ ($i = 1, \cdots, n$) are collected in an $n \times p$ matrix $\mathbf{X}$, where $x_{ij} = X_i(s_j)$. We discretize $\mathbf{V}(s)$ and $e_i(s)$ in Model (4) accordingly as a $p \times r$ loading matrix $\mathbf{V}$ with $v_{jk} = V_k(s_j)$ and an $n \times p$ error matrix $\mathbf{E}$ with $e_{ij} = e_i(s_j)$. We further denote $\mathbf{U} = (\mathbf{u}_{(1)}, \cdots, \mathbf{u}_{(n)})^T$ as an $n \times r$ score matrix, $\mathbf{Y} = (\mathbf{y}_{(1)}, \cdots, \mathbf{y}_{(n)})^T$ as an $n \times q$ supervision data matrix (viewed as fixed in the current context), and $\mathbf{F} = (\mathbf{f}_{(1)}, \cdots, \mathbf{f}_{(n)})^T$ as an $n \times r$ random error matrix. As a result, we obtain the following discretized version of the SupSFPC model (4):

$$\begin{cases} \mathbf{X} = \mathbf{U}\mathbf{V}^T + \mathbf{E} \\ \mathbf{U} = \mathbf{Y}\mathbf{B} + \mathbf{F} \end{cases}, \quad \text{or} \quad \mathbf{X} = \mathbf{Y}\mathbf{B}\mathbf{V}^T + \mathbf{F}\mathbf{V}^T + \mathbf{E}. \tag{5}$$

The identifiability conditions follow directly from those for the functional version of the model (4). Namely, $\mathbf{V}^T\mathbf{V}$ equals to an $r \times r$ identity matrix $\mathbf{I}_r$, and $\mathbf{\Sigma_f}$ is diagonal with positive decreasing eigenvalues.

To fit the SupSFPC model, we adopt a maximum likelihood approach. We assume normality for $\mathbf{E}$ and $\mathbf{F}$. In particular, we assume that e_{ij} is i.i.d. from an univariate normal distribution $\mathcal{N}(0, \sigma_{\mathbf{e}}^2)$, and $\mathbf{f}_{(i)}$ is i.i.d. from a multivariate normal distribution $\mathcal{N}_r(\mathbf{0}, \mathbf{\Sigma_f})$. In addition, e_{ij} is independent of $\mathbf{f}_{(i)}$. From (5), we see that the observation vector $\mathbf{x}_{(i)}$ follows $\mathcal{N}_p(\mathbf{V}\mathbf{B}^T\mathbf{y}_{(i)}, \mathbf{V}\mathbf{\Sigma_f}\mathbf{V}^T + \sigma_{\mathbf{e}}^2\mathbf{I})$, and different samples are independent. As a result, the log likelihood of the observed data matrix $\mathbf{X}$ is

$$\begin{aligned} \mathcal{L}(\mathbf{X}) = & -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log\det(\mathbf{V}\mathbf{\Sigma_f}\mathbf{V}^T + \sigma_{\mathbf{e}}^2\mathbf{I}_p) \\ & -\frac{1}{2}\mathrm{tr}\left((\mathbf{X} - \mathbf{Y}\mathbf{B}\mathbf{V}^T)(\mathbf{V}\mathbf{\Sigma_f}\mathbf{V}^T + \sigma_{\mathbf{e}}^2\mathbf{I}_p)^{-1}(\mathbf{X} - \mathbf{Y}\mathbf{B}\mathbf{V}^T)^T\right). \end{aligned}$$

To impose desirable structures (i.e., smoothness and sparsity) on $\mathbf{V}$ and $\mathbf{B}$, we optimize a regularized log likelihood function to estimate the model parameters. Let $\theta \triangleq (\mathbf{B}, \mathbf{V}, \sigma_{\mathbf{e}}^2, \mathbf{\Sigma_f})$ denote the model parameter set and Θ be the parameter space under the identifiability conditions. We propose to solve the following optimization problem:

$$\max_{\theta\in\Theta}\{\mathcal{L}(\mathbf{X}) - \mathcal{P}_f(\mathbf{V}) - \mathcal{P}_s(\mathbf{V}) - \mathcal{P}_s(\mathbf{B})\}, \tag{6}$$

where $\mathcal{P}_f(\mathbf{V})$ is the roughness penalty ("f" stands for functionality) on columns of $\mathbf{V}$, and $\mathcal{P}_s(\mathbf{V})$

and $\mathcal{P}_s(\mathbf{B})$ are the sparsity-inducing penalties ("s" stands for sparsity) on entries of $\mathbf{V}$ and $\mathbf{B}$ respectively. We remark by imposing sparsity on $\mathbf{B}$ we also avoid overfitting in the multivariate linear model (3) when the dimension of supervision data is high ($q > n$). Therefore, SupSFPC does not have any restrictions on the order of n, p, and q, and is suitable for high dimensional data.

For sparsity, numerous penalties have been proposed and studied in the literature (cf. Fan and Li, 2001; Tibshirani, 1996; Tibshirani et al., 2005; Yuan and Lin, 2006). In this paper, we present out method using the LASSO penalty (Tibshirani, 1996). It can be easily generalized to incorporate other penalties as well. The sparsity-inducing penalties in (6) take the following form:

$$\mathcal{P}_s(\mathbf{V}) = \sum_{k=1}^{r} \lambda_k \|\mathbf{v}_k\|_1, \quad \mathcal{P}_s(\mathbf{B}) = \sum_{k=1}^{r} \gamma_k \|\mathbf{b}_k\|_1, \tag{7}$$

where $\mathbf{v}_k$ and $\mathbf{b}_k$ are the kth columns of $\mathbf{V}$ and $\mathbf{B}$ corresponding to the kth layer of the low rank structure respectively, and λ_k and γ_k are the corresponding layer-specific tuning parameters.

For smoothness, generalized ℓ_2 penalties are widely used in the literature. Here we consider the elliptical ℓ_2 penalty:

$$\mathcal{P}_f(\mathbf{V}) = \sum_{k=1}^{r} \alpha_k \mathbf{v}_k^T \Omega \mathbf{v}_k, \tag{8}$$

where α_k are the layer-specific tuning parameters, and Ω is a fixed $p \times p$ positive semi-definite matrix depending on the sampling points, with the quadratic form $\mathbf{v}_k^T \Omega \mathbf{v}_k$ penalizing differences among adjacent values in $\mathbf{v}_k$. Here we use the same formulation of Ω as in Green and Silverman (1994) which connects nicely with smoothing splines.

The penalized likelihood framework (6) is very general and it subsumes many existing methods as we now discuss. If $\mathcal{P}_f(\mathbf{V}) = \mathcal{P}_s(\mathbf{V}) = \mathcal{P}_s(\mathbf{B}) = 0$, i.e., without any structural constraints, it reduces to the supervised SVD (SupSVD) method of Li et al. (2015). When $\mathcal{P}_s(\mathbf{B}) = \infty$, i.e., $\mathbf{B} = \mathbf{0}$, it reduces to regularized PCA methods: if $\mathcal{P}_f(\mathbf{V}) \neq 0$ and $\mathcal{P}_s(\mathbf{V}) = 0$, it corresponds to functional PCA of Huang et al. (2008); if $\mathcal{P}_f(\mathbf{V}) = 0$ and $\mathcal{P}_s(\mathbf{V}) \neq 0$, it results in sparse PCA of Shen and Huang (2008); if $\mathcal{P}_f(\mathbf{V}) \neq 0$ and $\mathcal{P}_s(\mathbf{V}) \neq 0$, one obtains the one-way situation of the sparse and functional PCA (SFPC) method of Allen (2013). We also note that the general framework includes many degenerated situations which have not been well studied before. For instance, when $\mathcal{P}_f(\mathbf{V}) = 0$ while $\mathcal{P}_s(\mathbf{V}) \neq 0$ and $\mathcal{P}_s(\mathbf{B}) \neq 0$, the framework reduces to a supervised PCA method with sparsity in **V** and **B**.

We want to comment on situations where the sampling points are different for different samples. For example, in longitudinal studies, patients may follow up at different times and also have distinct time domains. Similar situations have been referred to as sparsely-observed data in functional data analysis(see James et al., 2000; Yao et al., 2005, for example). In such situations, we can think of two possible approaches. For the first one, we can find a set of common grid points that are finer than the irregular sampling points, and treat the functional observations as missing on those grids where no data are observed. Our estimation algorithm can be extended to incorporate missing values. The second approach is to use basis expansion to interpolate the functional data, and then evaluate them on a set of common sampling points.

3 Computational Algorithm

In this section, we propose an algorithm for parameter estimation of the SupSFPC model. For the sake of clarity in describing the estimation algorithm, we first assume that all the tuning parameters, including the rank of the model, are given. We motivate and summarize the algorithm in Section 3.1, and derive the algorithm in more detail in Section 3.2. Then we briefly discuss the data-driven selection of tuning parameters in Section 3.3. Detailed derivation of the tuning parameter selection can be found in Section A of the online supplement.

3.1 EM Algorithm

Directly optimizing the penalized log likelihood (6) with respect to the identifiability constraints is non-trivial. The model parameters are intertwined in the log likelihood $\mathcal{L}(\mathbf{X})$: both the mean and the covariance terms share the parameter matrix **V**. In addition, the sparsity-inducing penalties are non-differentiable; the feasible region determined by the identifiability conditions is non-convex. We propose an algorithm that effectively combines the expectation-maximization (EM) algorithm with proximal gradient ascent (Beck and Teboulle, 2009; Nesterov, 2005) and block coordinate descent (Ortega and Rheinboldt, 2000) to overcome these computational difficulties.

To motivate the EM formulation, we first note that the hierarchical Model (5) contains the PC scores $\mathbf{U}$ as latent variables. It is easily seen that $\mathbf{x}_{(i)}$ and $\mathbf{u}_{(i)}$ are jointly normally distributed, and different samples are independent. The joint log likelihood of the observed data $\mathbf{X}$ and the latent data $\mathbf{U}$ can be decomposed as:

$$\mathcal{L}(\mathbf{X}, \mathbf{U}) = \mathcal{L}(\mathbf{X}|\mathbf{U}) + \mathcal{L}(\mathbf{U}),$$

where the conditional log likelihood of $\mathbf{X}$ given $\mathbf{U}$ is

$$\mathcal{L}(\mathbf{X}|\mathbf{U}) \propto -np \log \sigma_{\mathbf{e}}^2 - \sigma_{\mathbf{e}}^{-2} \mathrm{tr} \left[(\mathbf{X} - \mathbf{U}\mathbf{V}^T)(\mathbf{X} - \mathbf{U}\mathbf{V}^T)^T \right], \tag{9}$$

which only depends on $\mathbf{V}$ and $\sigma_{\mathbf{e}}^2$, while the marginal log likelihood of $\mathbf{U}$ is

$$\mathcal{L}(\mathbf{U}) \propto -n \log \det \mathbf{\Sigma_f} - \mathrm{tr} \left[(\mathbf{U} - \mathbf{YB})\mathbf{\Sigma_f}^{-1}(\mathbf{U} - \mathbf{YB})^T \right], \tag{10}$$

only depending on $\mathbf{B}$ and $\mathbf{\Sigma_f}$. Therefore, an EM algorithm can effectively separate the parameter estimation into two parts to simplify the optimization.

The EM algorithm iterates between an E step and an M step. In the $(t + 1)$th iteration, the E step is to calculate $\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[\mathcal{L}(\mathbf{X}, \mathbf{U})]$, where the expectation is taken with respect to $\mathbf{U}$ given $\mathbf{X}$ and $\theta^{(t)} = (\mathbf{B}^{(t)}, \mathbf{V}^{(t)}, \sigma_{\mathbf{e}}^{2(t)}, \mathbf{\Sigma_f}^{(t)})$, the estimated parameter set obtained in the tth iteration. The M step is to maximize $\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[\mathcal{L}(\mathbf{X}, \mathbf{U})] - \mathcal{P}_f(\mathbf{V}) - \mathcal{P}_s(\mathbf{V}) - \mathcal{P}_s(\mathbf{B})$ with respect to $\theta \in \Theta$, with the penalty terms as in (7) and (8). We denote the corresponding optimizer as $\theta^{(t+1)}$. After convergence, we obtain a local optimal solution for optimizing the regularized log likelihood (6).

Algorithm Summary: Before the detailed technical derivation, we summarize the algorithm with fixed tuning parameters below in Algorithm 1.

Algorithm 1 EM Algorithm for Fitting SupSFPC

```
Initialize model parameters $\theta^{(0)} = (\mathbf{B}^{(0)}, \mathbf{V}^{(0)}, \mathbf{\Sigma}_{\mathbf{f}}^{(0)}, \sigma_{\mathbf{e}}^{2(0)})$;
Repeat until convergence:
    (a) E Step:
        – Get critical conditional expectations (13), (14), and (15);
    (b) M Step:
        – Estimate $\mathbf{v}_k^{(t+1)}$ for $k = 1, \cdots, r$ from (20);
        – Estimate $\sigma_{\mathbf{e}}^{2(t+1)}$ from (18);
        – Estimate $\mathbf{b}_k^{(t+1)}$ for $k = 1, \cdots, r$ from (22);
        – Estimate $\mathbf{\Sigma}_{\mathbf{f}}^{(t+1)}$ from (19);
```

3.2 Derivation of the EM Algorithm

Since $\mathbf{u}_{(i)}$ and $\mathbf{x}_{(i)}$ are jointly normally distributed, the conditional distribution of $\mathbf{u}_{(i)}$ given $\mathbf{x}_{(i)}$ and $\theta^{(t)}$ is easily derived as $\mathcal{N}_r\left(\boldsymbol{\mu}_{(i)}^{(t)}, \mathbf{\Psi}^{(t)}\right)$ where

$$\boldsymbol{\mu}_{(i)}^{(t)} = \left(\mathbf{I}_r + \sigma_{\mathbf{e}}^{2(t)} \mathbf{\Sigma}_{\mathbf{f}}^{(t)-1}\right)^{-1} \left[\left(\sigma_{\mathbf{e}}^{2(t)} \mathbf{\Sigma}_{\mathbf{f}}^{(t)-1}\right) \mathbf{B}^{(t)^T} \mathbf{y}_{(i)} + \mathbf{V}^{(t)^T} \mathbf{x}_{(i)}\right], \tag{11}$$

$$\mathbf{\Psi}^{(t)} = \sigma_{\mathbf{e}}^{2(t)} \left(\mathbf{I}_r + \sigma_{\mathbf{e}}^{2(t)} \mathbf{\Sigma}_{\mathbf{f}}^{(t)-1}\right)^{-1}. \tag{12}$$

We remark that the conditional expectation of the PC scores for the ith sample is a weighted average of $\mathbf{B}^{(t)^T} \mathbf{y}_{(i)}$ and $\mathbf{V}^{(t)^T} \mathbf{x}_{(i)}$, where the weight is determined by $\mathbf{\Sigma}_{\mathbf{f}}^{(t)}$ and $\sigma_{\mathbf{e}}^{2(t)}$. Namely, in SupSFPC, the PC scores are partially driven by the supervision effect $\mathbf{y}_{(i)}$, and partially affected by the observation $\mathbf{x}_{(i)}$ as in the ordinary PCA.

In the E step, given (11) and (12), we can derive the explicit expression of $\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}(\mathcal{L}(\mathbf{X}, \mathbf{U}))$. As a matter of fact, we do not need to calculate the expectation of the entire joint log likelihood, but rather only the following three terms:

$$\text{first order term:} \quad \mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}(\mathbf{U}) \triangleq \mathbf{\Gamma}^{(t)} = \left(\boldsymbol{\mu}_{(1)}^{(t)}, \cdots, \boldsymbol{\mu}_{(n)}^{(t)}\right)^T, \tag{13}$$

$$\text{second order term:} \quad \mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}(\mathbf{U}^T\mathbf{U}) = n\mathbf{\Psi}^{(t)} + \mathbf{\Gamma}^{(t)^T}\mathbf{\Gamma}^{(t)}, \tag{14}$$

$$\text{quadratic form:} \quad \mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}\left[\mathrm{tr}\left(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\right)\right] = n\mathrm{tr}\left(\mathbf{\Lambda}\mathbf{\Psi}^{(t)}\right) + \mathrm{tr}\left(\mathbf{\Gamma}^{(t)}\mathbf{\Lambda}\mathbf{\Gamma}^{(t)^T}\right), \tag{15}$$

where $\mathbf{\Lambda}$ is any $r \times r$ symmetric matrix.

In the M step, we optimize $\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[\mathcal{L}(\mathbf{X},\mathbf{U})] - \mathcal{P}_f(\mathbf{V}) - \mathcal{P}_s(\mathbf{V}) - \mathcal{P}_s(\mathbf{B})$ with respect to $\theta \in \Theta$. It is equivalent to the following two separate optimization problems

$$\max_{\mathbf{V},\sigma_{\mathbf{e}}^2:\ \mathbf{V}^T\mathbf{V}=\mathbf{I}} \quad \mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[\mathcal{L}(\mathbf{X}|\mathbf{U})] - \sum_{k=1}^{r} \lambda_k \|\mathbf{v}_k\|_1 - \sum_{k=1}^{r} \alpha_k \mathbf{v}_k^T \Omega \mathbf{v}_k, \tag{16}$$

$$\max_{\mathbf{B},\Sigma_{\mathbf{f}}:\ \Sigma_{\mathbf{f}}=\mathrm{diag}(\Sigma_{\mathbf{f}})} \quad \mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[\mathcal{L}(\mathbf{U})] - \sum_{k=1}^{r} \gamma_k \|\mathbf{b}_k\|_1, \tag{17}$$

where $\mathcal{L}(\mathbf{X}|\mathbf{U})$ is given by (9), and $\mathcal{L}(\mathbf{U})$ is given by (10). The notation, $\mathrm{diag}(\Sigma_{\mathbf{f}})$, represents a diagonal matrix whose diagonal entries are the diagonal entries of $\Sigma_{\mathbf{f}}$.

Estimation of $\sigma_{\mathbf{e}}^2$ and $\Sigma_{\mathbf{f}}$

We take the first order derivative of (16) (or (17)) with respect to $\sigma_{\mathbf{e}}^2$ (or the diagonal entries of $\Sigma_{\mathbf{f}}$) and set them to zero, and obtain the analytical expressions (see Section B of the online supplement for the derivation)

$$\sigma_{\mathbf{e}}^{2(t+1)} = \frac{1}{np}\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}\left\{\mathrm{tr}\left[(\mathbf{X} - \mathbf{U}\mathbf{V}^{(t+1)^T})(\mathbf{X}^T - \mathbf{V}^{(t+1)}\mathbf{U}^T)\right]\right\}, \tag{18}$$

$$\Sigma_{\mathbf{f}}^{(t+1)} = \frac{1}{n}\mathrm{diag}\left\{\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}\left[(\mathbf{U} - \mathbf{Y}\mathbf{B}^{(t+1)})^T(\mathbf{U} - \mathbf{Y}\mathbf{B}^{(t+1)})\right]\right\}, \tag{19}$$

where $\mathbf{V}^{(t+1)}$ and $\mathbf{B}^{(t+1)}$ are the optimizers of $\mathbf{V}$ and $\mathbf{B}$ for (16) and (17) respectively, to be discussed below. In particular, the conditional expectation terms of (18) and (19) can be obtained using (13) to (15) from the E step.

Estimation of $\mathbf{V}$

Optimizing (16) with respect to $\mathbf{V}$ under the orthogonality constraint is formidable. Instead, we propose to drop the orthogonality and optimize the criterion with respect to the columns of $\mathbf{V}$, one at a time while fixing the others, mimicking a block coordinate descent algorithm. Since the conditional distribution (9) of $\mathbf{X}$ given $\mathbf{U}$ is identifiable even without the orthogonality condition, the optimization problem is still well defined. The scheme is similar to the deflation method used in the regularized PCA literature (see Allen, 2013; Hays et al., 2012; Huang et al., 2008; Shen and Huang, 2008, for example). We remark that the greedy algorithm maintains orthogonality of the columns of $\mathbf{V}$ approximately throughout the EM iterations. In our simulation study, the angle between any two loading vectors is typically greater than 85 degrees; in the yeast cell cycle example studied in Section 5, the smallest angle between any of the first four loadings is 87.7 degrees. Therefore, the column-by-column optimizers serve as a reasonable surrogate of the global optimizer of (16).

Given all the parameters except the kth column of $\mathbf{V}$, we can estimate $\mathbf{v}_k^{(t+1)}$ as

$$\mathbf{v}_k^{(t+1)} = \underset{\mathbf{v}_k:\|\mathbf{v}_k\|_2=1}{\arg\min}\ \frac{1}{2}\|\mathbf{v}_k - \boldsymbol{\beta}_k^{(t)}\|_2^2 + \lambda_k^{(t)}\|\mathbf{v}_k\|_1 + \frac{1}{2}\alpha_k^{(t)}\mathbf{v}_k^T\boldsymbol{\Omega}\mathbf{v}_k, \tag{20}$$

where $\boldsymbol{\beta}_k^{(t)} = \mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}\left[(\mathbf{X}^T - \mathbf{V}_{-k}^{(t)}\mathbf{U}_{-k}^T)\mathbf{u}_k\right]/c_k^{(t)}$, $\lambda_k^{(t)} = \sigma_{\mathbf{e}}^{2(t+1)}\lambda_k/(2c_k^{(t)})$, $\alpha_k^{(t)} = \sigma_{\mathbf{e}}^{2(t+1)}\alpha_k/c_k^{(t)}$, and $c_k^{(t)} = \mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}(\mathbf{u}_k^T\mathbf{u}_k)$. The matrices $\mathbf{U}_{-k}$ and $\mathbf{V}_{-k}^{(t)}$ are the submatrices of $\mathbf{U}$ and $\mathbf{V}^{(t)}$ leaving out the kth column $\mathbf{u}_k$ and $\mathbf{v}_k^{(t)}$, respectively. This setup facilitates parallel computing for the different columns in $\mathbf{V}$. The constants $\boldsymbol{\beta}_k^{(t)}$ and $c_k^{(t)}$ can be calculated from (13) and (14). The modified tuning parameters $\lambda_k^{(t)}$ and $\alpha_k^{(t)}$ can absorb the unknown constant $\sigma_{\mathbf{e}}^{2(t+1)}$, and be selected adaptively in a data-driven fashion in each iteration. For now, we treat them as known.

To solve (20), we adopt the *proximal gradient ascent* scheme studied in Nesterov (2005) and Beck and Teboulle (2009). We drop the subscripts and the superscripts in (20) for simplicity, and the optimization problem becomes $\min_{\mathbf{v}:\|\mathbf{v}\|_2=1} f(\mathbf{v}) + \lambda\|\mathbf{v}\|_1$ where $f(\mathbf{v}) \triangleq \frac{1}{2}\|\mathbf{v} - \boldsymbol{\beta}\|_2^2 + \frac{1}{2}\alpha\mathbf{v}^T\boldsymbol{\Omega}\mathbf{v}$. This optimization is solved by the iterative procedure

$$\mathbf{v}^{(l+1)} = \underset{\mathbf{v}:\|\mathbf{v}\|_2=1}{\arg\min}\left\{\frac{1}{2}\left\|\mathbf{v} - \left(\mathbf{v}^{(l)} - \frac{1}{L}\nabla f(\mathbf{v}^{(l)})\right)\right\|_2^2 + \frac{\lambda}{L}\|\mathbf{v}\|_1\right\}, \tag{21}$$

where ∇f is the gradient of f, and L is the Lipschitz constant of ∇f such that $\|\nabla f(\mathbf{a}) - \nabla f(\mathbf{b})\|_2 \leq L\|\mathbf{a}-\mathbf{b}\|_2$ for every $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$. Since $\nabla f(\mathbf{v}) = -\boldsymbol{\beta} + (\mathbf{I} + \alpha\Omega)\,\mathbf{v}$, L is the largest eigenvalue of $\mathbf{I} + \alpha\Omega$. Note that l is the proximal gradient ascent iteration index for estimating one column of $\mathbf{V}$, not to be confused with the EM iteration index t. In particular, we solve (21) approximately through the following two steps

$$\begin{aligned} \mathbf{v}^\star &= \mathbf{thres}\left(\mathbf{v}^{(l)} - \frac{1}{L}\nabla f(\mathbf{v}^{(l)})\,,\ \frac{\lambda}{L}\right), \\ \mathbf{v}^{(l+1)} &= \begin{cases} \dfrac{\mathbf{v}^\star}{\|\mathbf{v}^\star\|_2}, & \mathbf{v}^\star \neq \mathbf{0}, \\ \mathbf{0}, & \mathbf{v}^\star = \mathbf{0}, \end{cases} \end{aligned}$$

where **thres**() is a soft-thresholding function that $\mathbf{thres}(\boldsymbol{\beta}, \lambda) \triangleq \mathrm{sign}(\boldsymbol{\beta})(|\boldsymbol{\beta}| - \lambda)_+$.

Estimation of B

To estimate $\mathbf{B}$, we can rewrite (17) as r independent unconstrained optimization problems, and obtain each column of $\mathbf{B}$ as

$$\mathbf{b}_k^{(t+1)} = \min_{\mathbf{b}_k} \frac{1}{2}\|\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}(\mathbf{u}_k) - \mathbf{Y}\mathbf{b}_k\|_2^2 + \gamma_k^{(t)}\|\mathbf{b}_k\|_1, \tag{22}$$

where $\gamma_k^{(t)} = \sigma^{2\ (t+1)}_{\mathbf{f},k}\gamma_k/2$, with $\sigma^{2\ (t+1)}_{\mathbf{f},k}$ being the kth diagonal entry of $\boldsymbol{\Sigma}_\mathbf{f}^{(t+1)}$. The unknown constant $\sigma^{2\ (t+1)}_{\mathbf{f},k}$ is absorbed by the modified tuning parameter $\gamma_k^{(t)}$ that can be adaptively selected. The vector $\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}(\mathbf{u}_k)$ can be calculated from (13).

The optimization problem (22) is an univariate LASSO problem with $\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}(\mathbf{u}_k)$ being the response vector, $\mathbf{Y}$ being the $n \times q$ design matrix, and $\mathbf{b}_k$ being the coefficient vector. In addition, both the response vector and the design matrix are column centered, so there is no intercept term. Many methods have been developed to solve (22) (cf. Efron et al., 2004; Friedman et al., 2010). Here we use the default *coordinate descent* algorithm in Matlab (Friedman et al., 2010).

3.3 Tuning Parameter Selection

The tuning parameters in (6) play an important role in balancing the likelihood and the penalties. Note that there are $3r$ tuning parameters in the model. Searching over a $3r$-dimensional grid and refitting the model (potentially multiple times, if one uses cross validation) for each tuning set can be a huge computational burden. Instead, we adopt a nested procedure of selecting tuning parameters introduced by Huang et al. (2009). In each iteration, we find the optimal tuning parameters $\lambda_k^{(t)}$ and $\alpha_k^{(t)}$ for $\mathbf{v}_k^{(t+1)}$ while solving (20), and find the best $\gamma_k^{(t)}$ for $\mathbf{b}_k^{(t+1)}$ while solving (22). Numerical results illustrate this nested procedure always converges. Theoretical justification of the convergence property of the scheme is an open question.

In particular, when selecting $\lambda_k^{(t)}$ and $\alpha_k^{(t)}$, we assume that they do not interfere with each other and select one while fixing the other as zero. As a result, the selection of $\alpha_k^{(t)}$ is equivalent to selecting the smoothing parameter in a smoothing spline problem. For this selection task, we use leave-one-out cross validation (LOOCV), since the LOOCV score has an analytical form from Green and Silverman (1994) that facilitates fast computation. To select $\lambda_k^{(t)}$, we set it at an asymptotical value since the problem is equivalent to a filtering problem studied in Yang et al. (2013). The asymptotical value can induce appropriate amount of sparsity in $\mathbf{v}_k^{(t+1)}$. The tuning parameter $\gamma_k^{(t)}$ in (22) is selected using BIC, which is a popular choice in LASSO problems (Chand, 2012; Wang et al., 2009). The degree of freedom is determined in the same way as in Tibshirani et al. (2012). As a result, the algorithm is computationally efficient and scalable for high dimensional data. Numerical results in Sections 4 and 5 suggest that the scheme performs well. A more detailed derivation of tuning parameter selection can be found in Section A of the online supplement.

So far we have assumed that the rank r of the model is known. In practice, the rank needs to be determined from data. It is reasonable to assume that the rank of the underlying signal of the primary data matrix is inherent. Therefore, all rank selection methods studied in the PCA literature may be used in our framework. In this paper, we adopt a popular approach of using the scree plot of the primary data matrix to determine a proper rank. One can also consider other methods, such as the permutation assessment method in Buja and Eyuboglu (1992) and the bi-cross-validation method in Owen and Perry (2009). More sophisticated rank selection methods for functional data and high dimensional data need further investigation and are beyond the scope of the current paper.

4 Simulations

In this section, we compare SupSFPC with SupSVD proposed by Li et al. (2015), one-way SFPC proposed by Allen (2013), and the PCA using comprehensive simulations.

Simulation Settings

Data are generated from the low rank model: $\mathbf{X} = \mathbf{Y}\mathbf{B}\mathbf{V}^T + \mathbf{F}\mathbf{V}^T + \mathbf{E}$, which connects to the SupSFPC, SupSVD, one-way SFPC, and PCA models respectively through specific choices of $\mathbf{B}, \mathbf{\Sigma_f}$ and $\mathbf{V}$. Throughout the section, we assume that each entry of $\mathbf{E}$ is i.i.d. standard normal (i.e., $\sigma^2_{\mathbf{e}} = 1$).

Study I: We first consider a **unit-rank** setup where $n = 200$, $p = 100$, $q = 4$, $r = 1$. The 200×4 supervision matrix $\mathbf{Y}$ is filled with standard normal random numbers and then column centered. The 200×100 primary data matrix $\mathbf{X}$ is also column-centered after being generated. We focus on 4 settings where data are generated from each model respectively:

- **Case 1 (SupSFPC):** The loading vector $\mathbf{V}$ is shown in the left panel of Figure 1; the coefficient vector $\mathbf{B}$ is $(3, -3, 5, 0)^T$; $\mathbf{F}$ is a 200×1 random vector where each entry is i.i.d. standard normal (i.e., $\mathbf{\Sigma_f} = 1$).

- **Case 2 (SupSVD):** The parameters $\mathbf{B}$ and $\mathbf{\Sigma_f}$ are the same as in Case 1; the loading vector $\mathbf{V}$ is filled with standard normal random numbers and scaled to have norm one. Namely, there is no smoothness or sparsity in the loading.

- **Case 3 (SFPC):** The vector $\mathbf{V}$ is the same as in Case 1; the coefficient $\mathbf{B} = \mathbf{0}$, which eliminates the supervision effect; each entry of $\mathbf{F}$ is i.i.d. $\mathcal{N}(0, 9)$ (i.e., $\mathbf{\Sigma_f} = 9$).

- **Case 4 (PCA):** The parameters $\mathbf{B}$ and $\mathbf{\Sigma_f}$ are the same as in Case 3, and the loading vector $\mathbf{V}$ is obtained in the same way as in Case 2.

Study II: We then consider a **multi-rank** setup, where $n = 100$, $p = 120$, $q = 10$, $r = 3$. Again, the 100×10 supervision matrix $\mathbf{Y}$ contains standard normal random numbers and column centered. The 100×120 primary data $\mathbf{X}$ is also column-centered after being generated. Similarly to the unit-rank setup, we consider the following 4 settings:

- **Case 5 (SupSFPC):** The loading vectors in $\mathbf{V}$ are shown in the right panel of Figure 1; the 10×3 coefficient matrix $\mathbf{B} = [3, -4, 2, -1, \mathrm{rep}(0, 6); \mathrm{rep}(0, 3), 2, -3, 1, 1, \mathrm{rep}(0, 3); \mathrm{rep}(0, 6), -1, 1, 1, 2]^T$, where rep(a,b) means repeat a b times; the 3×3 covariance matrix $\mathbf{\Sigma_f}$ is a diagonal matrix with diagonal values $(1, 3, 4)$.
- **Case 6 (SupSVD):** The parameters $\mathbf{B}$ and $\mathbf{\Sigma_f}$ are the same as in Case 5; the 120×3 loading matrix $\mathbf{V}$ is filled with standard normal random numbers and normalized to have orthonormal columns.
- **Case 7 (SFPC):** The loading matrix $\mathbf{V}$ is the same as in Case 5; the coefficient matrix $\mathbf{B} = \mathbf{0}$, which eliminates the supervision effect; the covariance matrix $\mathbf{\Sigma_f}$ is diagonal with diagonal values $(16, 9, 4)$.
- **Case 8 (PCA):** The parameters $\mathbf{B}$ and $\mathbf{\Sigma_f}$ are the same as in Case 7, and the loading vectors are obtained in the same way as in Case 6.

Performance Measures

We compare the methods in three aspects, *loading estimation*, *score prediction*, and *low-rank structure recovery*. To evaluate the loading estimation accuracy, we use two criteria, the *mean square error* and the *largest principal angle* (Golub and Van Loan, 2012):

$$MSE_{\mathbf{V}} = \frac{1}{pr}\|\mathbf{V} - \widehat{\mathbf{V}}\|_{\mathbb{F}}^2, \qquad Angle_{\mathbf{V}} = \frac{180}{\pi}\arccos(\min\mathrm{eig}(\mathbf{V}^T\widehat{\mathbf{V}})),$$

where $\|\cdot\|_{\mathbb{F}}$ denotes the Frobenius norm, and $\min\mathrm{eig}(\cdot)$ denotes the minimal eigenvalue. The former characterizes the entry-wise accuracy, and the latter captures the subspace-wise accuracy which is invariant to rotations. For evaluating score prediction and low-rank structure recovery, we use *mean squared prediction errors* defined as:

$$MSPE_{\mathbf{U}} = \frac{1}{nr}\|\mathbf{U} - \widehat{\mathbf{U}}\|_{\mathbb{F}}^2, \qquad MSPE_{\mathbf{UV}^T} = \frac{1}{np}\|\mathbf{UV}^T - \widehat{\mathbf{U}}\widehat{\mathbf{V}}^T\|_{\mathbb{F}}^2,$$

where the true scores $\mathbf{U} = \mathbf{YB} + \mathbf{F}$, and the predicted $\widehat{\mathbf{U}}$ have different formulas for different methods. For SupSFPC and SupSVD, $\widehat{\mathbf{U}} = \mathbb{E}_{\mathbf{U}|\mathbf{X},\widehat{\theta}}(\mathbf{U})$ where $\widehat{\theta}$ is specific to respective methods; for SFPC and PCA, $\widehat{\mathbf{U}} = \mathbf{X}\widehat{\mathbf{V}}$ where $\widehat{\mathbf{V}}$ is method specific.

Results

For each case, we repeat the simulation 100 times and present the median and the median absolute deviation (MAD) of each performance measurement for all methods in Table 1. The results show that SupSFPC outperforms the other methods in all cases, in terms of the considered aspects. One explanation for the superior performance is that SupSFPC is a general framework unifying many existing methods. It automatically adapts to a wide range of practical situations.

There are several interesting observations in Table 1. First, in Case 3 and Case 7 , SFPC surprisingly performs badly in all aspects. This is likely due to an inadequate tuning parameter selection procedure. The original SFPC paper did not provide any guidance on how to set tuning grids for BIC, which is a crucial issue in practice. We consulted with the author and used a suggested tuning grid here. Second, in Case 4 and Case 8, SupSFPC and SupSVD outperform SFPC and PCA in terms of score prediction and low-rank structure recovery. Since the auxiliary data are irrelevant in both cases, the improvement in score prediction must come from the shrinkage effect imposed by $\left(\mathbf{I} + \sigma_{\mathbf{e}}^2 \mathbf{\Sigma}_{\mathbf{f}}^{-1}\right)^{-1}$ in (13). This has been studied from a random matrix point of view by Shabalin and Nobel (2013). Third, in Case 6 where the generating model is SupSVD, the medians of $MSPE_{\mathbf{U}}$ and $MSE_{\mathbf{V}}$ for SupSVD are larger than SupSFPC. In this case the only difference between SupSFPC and SupSVD is that the former does not require strict orthogonality in loading estimation, so we think the improvement comes from this extra flexibility. Nevertheless, both SupSVD and SupSFPC have similar medians of $MSPE_{\mathbf{U}\mathbf{V}^T}$ that are superior to SFPC and PCA. This suggests that the recovery of low-rank structures actually benefits from incorporating auxiliary data.

5 Real Data Example: Yeast Cell Cycle Data

In this section, we demonstrate the advantage of SupSFPC using a yeast cell cycle data set.Two additional real data examples, a government bond yield data set and a hospital emergency room visit data set, are considered in Sections C and D of the online supplement.

We consider microarray expression measurements ($\mathbf{X}$) of yeast genes over a certain time period. About 800 cell cycle-related genes are identified in Spellman et al. (1998) through three independent synchronization methods. We consider the data from the α factor based experiment where mRNA levels were measured at every 7 minutes for 18 time points (about 2 hours) covering two cell cycles. In addition to the expression data, we also have ChIP-chip data (Lee et al., 2002) that contain binding information ($\mathbf{Y}$) of 106 TFs for the cell cycle-related genes. We exclude genes with missing values in either expression measurements or TF binding information as in Chen and Huang (2012) and Chun and Keleş (2010), and consider a subset of 542 genes. The data are publicly available in the R package "spls". Figure 2 shows the raw expression time series of the 542 cell cycle-related genes.

The goal of the yeast cell cycle data analysis is two-fold: 1) understanding the underlying expression patterns of cell cycle-related genes, and 2) identifying transcription factors (TFs) that regulate cell cycles. Below we address both topics simultaneously using SupSFPC. Zhao et al. (2004) primarily focus on the former by projecting the raw time series onto Fourier basis functions with even frequencies and carrying out principal component analysis of the projected data. Chun and Keleş (2010) and Chen and Huang (2012) study the latter by regressing the gene expression data onto TF data through sparse partial least square and sparse reduced rank regression respectively.

The primary data matrix $\mathbf{X}$ contains expression measurements of 542 genes at 18 time points. The supervision data matrix $\mathbf{Y}$ contains binding information of the same genes for 106 TFs. We mean center each time point in $\mathbf{X}$ and each TF in $\mathbf{Y}$. Based on the scree plot of singular values of the column-centered data matrix $\mathbf{X}$, we select the rank to be $r = 4$. The fitting procedure took about an hour (641 EM iterations) on a standard desktop (Intel Xeon CPU X5570 @ 2.93GHz dual processor) to reach relatively high accuracy (the ℓ_2 difference between consecutive estimates of the loading matrix below 10^{-3}).

Figure 3 compares the loading estimates from four methods: SupSFPC, SupSVD, SFPC and PCA. By taking into account the auxiliary binding information, the SupSFPC loadings are the most interpretable ones. The first and the forth loading vectors of SupSFPC effectively capture periodic patterns of cell cycles without referring to a priori knowledge of true cyclic information as in Zhao et al. (2004). In addition, the second loading mainly presents the variation in the first cell cycle, and the third loading reflects the contrast of the two cycles. The fourth loading also emphasizes the variation in the second cycle.

Figure 4 shows the clustering results of the 542 cell cycle-related genes based on SupSFPC scores. We apply a 5-mean clustering approach, where the number of clusters is suggested by Zhao et al. (2004). Different clusters contain genes with different periodic phases. In particular, the genes in the 2nd-5th clusters clearly exhibit different cyclic patterns, similar to the results in Zhao et al. (2004). The genes in the first cluster, on the other hand, do not show strong periodicity, which may need further investigation.

We also investigate the TF activities. Active TFs correspond to the nonzero rows of the estimated supervision coefficient matrix $\widehat{\mathbf{B}}$. Out of the 106 TFs, we identify 32 to be active, with 13 of them being among the 21 experimentally confirmed TFs in Wang et al. (2007). The TF activities for those discovered by SupSFPC are shown in Figure 5. Most of the confirmed TFs have clear periodic behavior; among the unconfirmed ones, DOT6, MET4, SFL1, and YAP5 have the most significant cyclic patterns, which may provide useful guidance to further investigate the regulation effect of TFs on yeast cell cycle.

Supplementary Materials

Appendices: Technical derivations of the algorithm in A and B; additional real data examples in C and D. (Appendix.pdf; pdf file)

Code and Data: Matlab codes for different methods (i.e., SupSFPC, SFPC and SupSVD) and numerical analyses, along with the yeast cell cycle data, bond yield data and emergency room data. (codedata.zip; zip file)

Acknowledgements

This research was partially supported by NSF Grants DMS-1106912 (HS and GL), DMS-1407655 (HS and GL), and DMS-1208952 (JH), and the Xerox UAC Foundation (HS).

References

Allen, G. I. (2013). Sparse and functional principal components analysis. *arXiv preprint arXiv:1309.2895*.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202.

Buja, A. and Eyuboglu, N. (1992). Remarks on parallel analysis. *Multivariate Behavioral Research*, 27(4):509–540.

Chand, S. (2012). On tuning parameter selection of lasso-type methods-a monte carlo study. In *Applied Sciences and Technology (IBCAST), 2012 9th International Bhurban Conference on*, pages 120–129. IEEE.

Chen, L. and Huang, J. Z. (2012). Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, 107(500):1533–1545.

Chun, H. and Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25.

d'Aspremont, A., Bach, F., and Ghaoui, L. E. (2008). Optimal solutions for sparse principal component analysis. *The Journal of Machine Learning Research*, 9:1269–1294.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

Green, P. J. and Silverman, B. W. (1994). *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press.

Hays, S., Shen, H., Huang, J. Z., et al. (2012). Functional dynamic factor models with application to yield curve forecasting. *The Annals of Applied Statistics*, 6(3):870–894.

Huang, J. Z., Shen, H., and Buja, A. (2008). Functional principal components analysis via penalized rank one approximation. *Electronic Journal of Statistics*, 2:678–695.

Huang, J. Z., Shen, H., and Buja, A. (2009). The analysis of two-way functional data using two-way regularized singular value decompositions. *Journal of the American Statistical Association*, 104(488).

James, G. M., Hastie, T. J., and Sugar, C. A. (2000). Principal component models for sparse functional data. *Biometrika*, 87(3):587–602.

James, G. M., Wang, J., and Zhu, J. (2009). Functional linear regression that's interpretable. *The Annals of Statistics*, pages 2083–2108.

Lee, M., Shen, H., Huang, J. Z., and Marron, J. (2010). Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095.

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., et al. (2002). Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804.

Lee, T. I. and Young, R. A. (2000). Transcription of eukaryotic protein-coding genes. *Annual Review of Genetics*, 34(1):77–137.

Li, G., Yang, D., Nobel, A. B., and Shen, H. (2015). Supervised singular value decomposition and its asymptotic properties. *Journal of Multivariate Analysis*, In Press.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152.

Nikolov, D. and Burley, S. (1997). Rna polymerase ii transcription initiation: a structural view. *Proceedings of the National Academy of Sciences*, 94(1):15–22.

Ortega, J. M. and Rheinboldt, W. C. (2000). *Iterative solution of nonlinear equations in several variables*, volume 30. SIAM.

Owen, A. B. and Perry, P. O. (2009). Bi-cross-validation of the svd and the nonnegative matrix factorization. *The Annals of Applied Statistics*, pages 564–594.

Rice, J. A. and Silverman, B. W. (1991). Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 233–243.

Shabalin, A. and Nobel, A. (2013). Reconstruction of a low-rank matrix in the presence of Gaussian noise. *Journal of Multivariate Analysis*, 118:67–76.

Shen, H. and Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034.

Silverman, B. W. et al. (1996). Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1–24.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Tibshirani, R. J., Taylor, J., et al. (2012). Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232.

Wang, H., Li, B., and Leng, C. (2009). Shrinkage tuning parameter selection with a diverging number of parameters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):671–683.

Wang, L., Chen, G., and Li, H. (2007). Group scad regression analysis for microarray time course gene expression data. *Bioinformatics*, 23(12):1486–1494.

Yang, D., Ma, Z., and Buja, A. (2013). A sparse svd method for high-dimensional data. *Journal of Computational and Graphical Statistics*, (Forthcoming).

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Zhao, X., Marron, J. S., and Wells, M. T. (2004). The functional data analysis view of longitudinal data. *Statistica Sinica*, 14(3):789–808.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286.

Supplement: Technical Details and Additional Numerical Studies for “Supervised Sparse and Functional Principal Component Analysis”

A Tuning Parameter Selection

In this section, we elaborate on the tuning parameter selection procedures for SupSFPC, which are briefly discussed in Section 3.3 of the main paper. For computational efficiency, we embed the selection procedures in each EM iteration, as in Huang et al. (2009) and Allen (2013). Before presenting more technical details, we summarize the comprehensive SupSFPC algorithm in Algorithm 2.

Algorithm 2 EM Algorithm for SupSFPC with Adaptive Tuning Selection

```
Initialize model parameters $\theta^{(0)} = (\mathbf{B}^{(0)}, \mathbf{V}^{(0)}, \mathbf{\Sigma}_{\mathbf{f}}^{(0)}, \sigma_{\mathbf{e}}^{2(0)})$;
Repeat until convergence:
    (a) E Step:
        * Get critical conditional expectations (13), (14), and (15);
    (b) M Step:
        * for $k = 1 \cdots r$ do
            · Select $\alpha_k^{(t)}$ from (24);
            · Set $\lambda_k^{(t)}$ to be (25);
            · Estimate $\mathbf{v}_k^{(t+1)}$ from (20);
        * end for
        * Estimate $\sigma_{\mathbf{e}}^{2(t+1)}$ from (18);
        * for $k = 1 \cdots r$ do
            · Select $\gamma_k^{(t)}$ from (26);
            · Estimate $\mathbf{b}_k^{(t+1)}$ from (22);
        * end for
        * Estimate $\mathbf{\Sigma}_{\mathbf{f}}{}^{(t+1)}$ from (19);
```

A.1 Select α and λ

The optimization (20) involves two tuning parameters: $\alpha_k^{(t)}$ and $\lambda_k^{(t)}$. They control the smoothness and the sparsity of the kth estimated loading vector $\mathbf{v}_k^{(t+1)}$, respectively. To select the best values for both tuning parameters simultaneously, one may search over a 2-dimensional tuning grid and use cross validation methods (Zou and Hastie, 2005) or information theoretic criteria (Allen, 2013). However, the searching procedure is computationally intensive, especially when we do not have a good knowledge of the range of different tuning parameters and have to search over a large grid. Moreover, we need to repeat the procedure for different PC layers in every EM iteration. The overall computational cost can be huge.

As a remedy, we propose to select $\alpha_k^{(t)}$ and $\lambda_k^{(t)}$ separately. In particular, we omit the sparsity penalty (i.e., set $\lambda_k^{(t)} = 0$) when selecting the smoothness parameter $\alpha_k^{(t)}$, and vice versa. An advantage of this approach is that the optimization (20) reduces to two well-studied problems: a smoothing spline problem (when $\lambda_k^{(t)} = 0$) and a penalized least square problem (when $\alpha_k^{(t)} = 0$). For each respective problem, the other tuning parameter can be selected adaptively using some computationally efficient methods. We drop the subscripts and the superscripts in (20) for simplicity and discuss in more detail below.

When $\lambda = 0$, (20) becomes

$$\min_{\mathbf{v}} \ \|\boldsymbol{\beta} - \mathbf{v}\|_2^2 + \alpha \mathbf{v}^T \Omega \mathbf{v} \tag{23}$$

where Ω has an expression that is the same as that in smoothing splines (Green and Silverman, 1994). Therefore, (23) is a smoothing spline problem. For a given $\alpha > 0$, the closed-form solution of (23) is $\widehat{\mathbf{v}}_\alpha = \mathbf{H}_\alpha \boldsymbol{\beta}$, where $\mathbf{H}_\alpha = (\mathbf{I} + \alpha\Omega)^{-1}$ is a $p \times p$ hat matrix. Leave-one-out cross validation (LOOCV) is commonly used to select the smoothing parameter α in smoothing splines. Given an α, we leave out one entry of $\boldsymbol{\beta}$ at a time, and solve (23) to get a smooth estimate of $\mathbf{v}$; then we calculate the squared difference between the left-out value in $\boldsymbol{\beta}$ and the corresponding interpolated value in $\mathbf{v}$; we repeat the procedure for all entries of $\boldsymbol{\beta}$ and sum up the squared differences as the LOOCV score for this tuning parameter α. In a candidate tuning set, the one that has the smallest LOOCV score is the optimal tuning parameter.

Solving (23) multiple times for each α can be computationally expensive. However, Green and Silverman (1994) show that the LOOCV score for smoothing spline problems can be obtained analytically by solving the full optimization problem once as

$$\text{LOOCV}(\alpha) = \frac{1}{p}\sum_{j=1}^{p}\left(\frac{\beta_j - \widehat{v}_{\alpha,j}}{1 - h_{\alpha,jj}}\right)^2, \tag{24}$$

where β_j and $\widehat{v}_{\alpha,j}$ are the jth entry of $\boldsymbol{\beta}$ and $\widehat{\mathbf{v}}_\alpha$, and $h_{\alpha,jj}$ is the jth diagonal entry of $\mathbf{H}_\alpha$. Therefore, LOOCV is an efficient method for tuning parameter selection in smoothing spline. We adopt LOOCV for selecting α in our algorithm. In practice, we can search over a wide range of candidate values at rather low cost.

Given $\alpha = 0$, (20) reduces to a penalized least square problem:

$$\min_{\mathbf{v}} \frac{1}{2}||\boldsymbol{\beta} - \mathbf{v}||_2^2 + \lambda||\mathbf{v}||_1,$$

which has an explicit solution $\widehat{\mathbf{v}}_\lambda = \textbf{thres}(\boldsymbol{\beta}, \lambda)$. Namely, $\lambda > 0$ is the shrinkage amount imposed on $\boldsymbol{\beta}$. Given $\mathbf{U}$, we know from the definition that the vector $\boldsymbol{\beta} = (\mathbf{E}^T + \mathbf{v}\mathbf{u}^T)\mathbf{u}/||\mathbf{u}||_2^2 = \mathbf{v} + \mathbf{E}^T\mathbf{u}/||\mathbf{u}||_2^2$, where $\mathbf{E}$ is the measurement error matrix in Model (5) with i.i.d. entries from $\mathcal{N}(0, \sigma_{\mathbf{e}}^2)$. Namely, $\boldsymbol{\beta}$ can be viewed as the true sparse vector $\mathbf{v}$ plus a noise vector with i.i.d. entries from $\mathcal{N}(0, \sigma_{\mathbf{e}}^2/||\mathbf{u}||_2^2)$. To accurately estimate the zero entries in $\mathbf{v}$, a proper threshold is the asymptotically tight upper bound of the expectation of infinity norm of the noise vector, which is $\sqrt{2\log(p)\sigma_{\mathbf{e}}^2/||\mathbf{u}||_2^2}$ (Yang et al., 2013). In practice, since both $||\mathbf{u}||_2^2$ and $\sigma_{\mathbf{e}}^2$ are unknown, we substitute them with estimates from the previous EM iteration. In particular, the approximate optimal value for $\lambda_k^{(t)}$ is

$$\lambda_k^{(t)} = \sqrt{2\log(p)\sigma_{\mathbf{e}}^{2(t)}/c_k^{(t)}}, \tag{25}$$

where $c_k^{(t)} = \mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}(\mathbf{u}_k^T\mathbf{u}_k)$. Numerical studies indicate this constant works well.

A.2 Select γ

The tuning parameter $\gamma_k^{(t)}$ in (22) is a LASSO sparsity parameter as we formulate and solve (22) as a LASSO problem. Selection of sparsity parameter in a LASSO problem has been well studied in the literature. See, for example, Wang et al. (2009) and Chand (2012). Among data-driven approaches, BIC is a favorable method due to its theoretical merit and fast computation. In particular, since the coordinate descent algorithm can recover the entire solution path efficiently, using BIC to tune LASSO roughly has the same cost as fitting LASSO with a known parameter. Therefore, the BIC procedure for selecting $\gamma_k^{(t)}$ is suitable to be embedded in the EM iteration. For simplicity, we drop the subscripts and the superscripts in the discussion below.

In (22), the BIC score for a given tuning parameter γ is defined as

$$\mathrm{BIC}(\gamma) = n\log(\mathrm{MSE}_\gamma) + \mathrm{df}_\gamma \log(n), \tag{26}$$

where MSE_γ is the mean residual sum of squares, and df_γ is the degree of freedom of the fitted model corresponding to the tuning parameter γ. The degree of freedom of a LASSO fit has been studied in Zou et al. (2007) for a full-column-rank design matrix, and in Tibshirani et al. (2012) for general design matrices. In our case, the design matrix is $\mathbf{Y}$ where columns are potentially linearly dependent. Therefore, we estimate df_γ according to Tibshirani et al. (2012) as

$$\widehat{\mathrm{df}}_\lambda = \mathrm{rank}(\mathbf{Y}_{\mathcal{A}(\gamma)}),$$

where $\mathcal{A}(\gamma)$ is a column index set corresponding to nonzero LASSO estimates at γ, and $\mathbf{Y}_{\mathcal{A}(\gamma)}$ is a submatrix of $\mathbf{Y}$ with columns in $\mathcal{A}(\gamma)$. The value that leads to the smallest BIC score is the selected tuning parameter.

B Derivation of (19)

Note that we require $\boldsymbol{\Sigma}_{\mathbf{f}}$ to be a diagonal matrix, which is denoted by $\boldsymbol{\Sigma}_{\mathbf{f}} = \text{diag}(\sigma^2_{\mathbf{f},1}, \cdots, \sigma^2_{\mathbf{f},r})$. We can rewrite the first term of the object function in (17) as

$$\begin{aligned}\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[\mathcal{L}(\mathbf{U})] &\propto -\log\det\boldsymbol{\Sigma}_{\mathbf{f}} - \frac{1}{n}\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}\{\text{tr}[(\mathbf{U}-\mathbf{YB})\boldsymbol{\Sigma}_{\mathbf{f}}^{-1}(\mathbf{U}-\mathbf{YB})^T]\} \\ &= -\sum_{k=1}^{r}\log\sigma^2_{\mathbf{f},k} - \frac{1}{n}\sum_{k=1}^{r}\sigma^{-2}_{\mathbf{f},k}\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[(\mathbf{u}_k-\mathbf{Y}\mathbf{b}_k)^T(\mathbf{u}_k-\mathbf{Y}\mathbf{b}_k)].\end{aligned}$$

Therefore, the maximization over the diagonal entries of the diagonal matrix $\boldsymbol{\Sigma}_{\mathbf{f}}$ can be separated into r univariate problems, each being

$$\max_{\sigma^2_{\mathbf{f},k}} \quad -\log\sigma^2_{\mathbf{f},k} - \frac{1}{n}\sigma^{-2}_{\mathbf{f},k}\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[(\mathbf{u}_k-\mathbf{Y}\mathbf{b}_k)^T(\mathbf{u}_k-\mathbf{Y}\mathbf{b}_k)].$$

It then follows that the optimal solution is

$$\widehat{\sigma^2_{\mathbf{f},k}} = \frac{1}{n}\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}[(\mathbf{u}_k-\mathbf{Y}\mathbf{b}_k)^T(\mathbf{u}_k-\mathbf{Y}\mathbf{b}_k)].$$

Therefore, $\widehat{\boldsymbol{\Sigma}}_{\mathbf{f}} = \frac{1}{n}\text{diag}\left\{\mathbb{E}_{\mathbf{U}|\mathbf{X},\theta^{(t)}}\left[(\mathbf{U}-\mathbf{YB}^{(t+1)})^T(\mathbf{U}-\mathbf{YB}^{(t+1)})\right]\right\}$.

C Government Bond Yield Data

In this section, we consider the application of SupSFPC to the government bond yield data also studied in Diebold and Li (2006) and Hays et al. (2012). We use the example to illustrate that when auxiliary data are *irrelevant* to the primary data of interest, SupSFPC can adaptively ignore the supervision effect and perform as well as an unsupervised method.

The primary data contain the end-of-month price quotes for U.S. Treasuries, from January 1985 to December 2000 (192 months). For each month, we consider yields on zero coupon bonds of 18 fixed maturities (imputed if missing) of 1.5, 3, 6, 9, 12, 15, 18, 21, 24, 30, 36, 48, 60, 72, 84, 96, 108, 120 months. Each month is a sample ($n = 192$) and each maturity is a variable ($p = 18$), resulting in a 192×18 primary data matrix $\mathbf{X}$. The 192 raw yield curves of different maturities are shown in Figure 6, with random coloring. For each sample, we also have the auxiliary monthly index information, which may or may not influence the underlying structure of $\mathbf{X}$. In particular, we treat the monthly indices (converted to dummy variables) for the 192 months as supervision data $\mathbf{Y}$.

Each column of $\mathbf{X}$ and $\mathbf{Y}$ is centered before applying SupSFPC. We set the rank $r = 2$ as the first 2 principal components of the column-centered $\mathbf{X}$ explain over 99% of the total variation. Then we estimate the SupSFPC model parameters from the data. The fitting procedure took less than 1 second to converge (6 EM iterations). The estimated supervision coefficient matrix $\widehat{\mathbf{B}}$ is a zero matrix, meaning the auxiliary monthly index data are not relevant to the underlying structure of the yield data. Namely, the yield curves do not present any strong monthly patterns. This is concordant with our observation from the raw data in Figure 6.

We also compare the loading vectors estimated from SupSFPC with those obtained deterministically from the dynamic Nelson-Siegel (DNS) model (Diebold and Li, 2006), which is designed under prior economic theory guidance. Figure 7 shows the comparison results. The first panel shows the mean yield curve from the data versus the first loading from the DNS model, both representing a long-term factor. The deviance indicates that the constant loading of the DNS model may not be adequate to capture the overall yield trend at different maturities. The other two panels show the comparison between the 2nd and 3rd loading vectors between SupSFPC and DNS, respectively. From an economic point of view, the two pre-specified DNS loadings possess the interpretation of medium-term and short-term effects respectively. The two SupSFPC loadings have similar shapes with the respective DNS loadings, meaning that SupSFPC captures similar yield curve patterns as in the DNS model. However, we note that SupSFPC only uses information in the data without referring to any economic prior knowledge. Namely, SupSFPC is flexible enough to adapt to the dominant features in the data.

D Emergency Room Visit Data

We now analyze the patient arrival rate data from Armony et al. (2011), that contain hourly number of patients arriving to the emergency room (ER) of the Rambam Hospital, Israel for 417 consecutive days (from September 10th, 2006 to October 31th, 2007). The goal is to understand underlying patient arrival patterns to better allocate human and medical resources. The 417 raw arrival rate curves are shown in the 1st panel of Figure 8.

Other than the hourly arrival rates, we also know the day-of-week index of each day. In particular, the 2nd-8th panels in Figure 8 show arrival curves grouped by the day-of-week index. It can be seen that different days of a week have distinct arrival patterns. Namely, the day-of-week index may be treated as supervision information as it partially drives the underlying structure of the arrival rates.

Each row of the 417×24 primary data matrix contains hourly arrival rates of a day. We apply a square root transformation to the arrival rate data (i.e., $\sqrt{\text{arrival rate} + 1/4}$) to achieve approximate normality (Brown et al., 2005). Then we column center the data matrix and denote it as $\mathbf{X}$. The supervision data matrix $\mathbf{Y}$ contains 417 day-of-week indices (converted to dummy variables and column centered). The rank is set to be 4 based on the scree plot of the singular values of $\mathbf{X}$.

We apply different methods (SupSFPC, SupSVD, one-way SFPC and PCA) to the data. The fitting procedure of SupSFPC took about 1 minute to converge (166 EM iterations). Figure 9 shows the loading vectors estimated from the methods. By taking into account the auxiliary day-of-week information and allowing regularization, SupSFPC loadings have superior interpretability. The four loadings of SupSFPC capture major variabilities of arrival data from unknown sources after separating the day-of-week effect. They represent large variations of arrival rates at noon, in the evening, overnight, and in the morning, respectively. The day-of-week structure identified by SupSFPC is shown in Figure 10. To get the curves in the figure, we transform $\mathbf{Y}\widehat{\mathbf{B}}\widehat{\mathbf{V}}^T$ (where $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{V}}$ are the SupSFPC parameters estimated from the data) back into the original scale by adding the column mean of $\mathbf{X}$ and applying a square transformation. The recovered low-rank structures resemble the (smoothed) average arrival patterns for different weekdays in Figure 8.

References

Allen, G. I. (2013). Sparse and functional principal components analysis. *arXiv preprint arXiv:1309.2895*.

Armony, M., Israelit, S., Mandelbaum, A., Marmor, Y. N., Tseytlin, Y., and Yom-Tov, G. B. (2011). Patient flow in hospitals: A data-based queueing-science perspective. *New York University. Available at: http://www.stern.nyu.edu/om/faculty/armony/*.

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., and Zhao, L. (2005). Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American Statistical Association*, 100(469):36–50.

Chand, S. (2012). On tuning parameter selection of lasso-type methods-a monte carlo study. In *Applied Sciences and Technology (IBCAST), 2012 9th International Bhurban Conference on*, pages 120–129. IEEE.

Diebold, F. X. and Li, C. (2006). Forecasting the term structure of government bond yields. *Journal of Econometrics*, 130(2):337–364.

Green, P. J. and Silverman, B. W. (1994). *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press.

Hays, S., Shen, H., Huang, J. Z., et al. (2012). Functional dynamic factor models with application to yield curve forecasting. *The Annals of Applied Statistics*, 6(3):870–894.

Huang, J. Z., Shen, H., and Buja, A. (2009). The analysis of two-way functional data using two-way regularized singular value decompositions. *Journal of the American Statistical Association*, 104(488).

Tibshirani, R. J., Taylor, J., et al. (2012). Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232.

Wang, H., Li, B., and Leng, C. (2009). Shrinkage tuning parameter selection with a diverging number of parameters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):671–683.

Yang, D., Ma, Z., and Buja, A. (2013). A sparse svd method for high-dimensional data. *Journal of*

Computational and Graphical Statistics, (Forthcoming).

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Zou, H., Hastie, T., Tibshirani, R., et al. (2007). On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192.

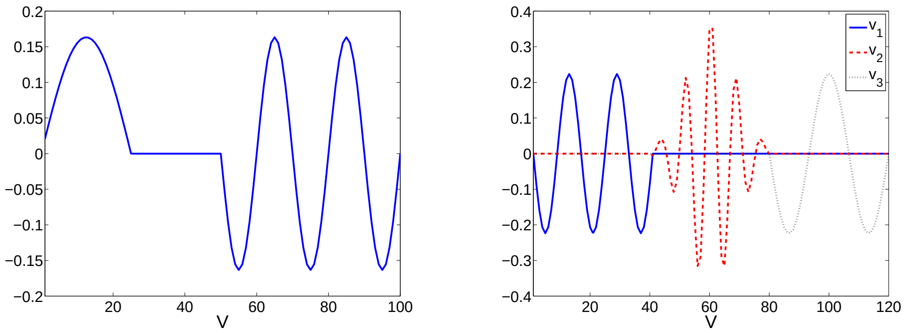

Figure 1: Smooth and sparse loading vectors. Left: the loading vector for Cases 1 and 3 in the unit-rank example; right: the loading vectors for Cases 5 and 7 in the rank-3 example.

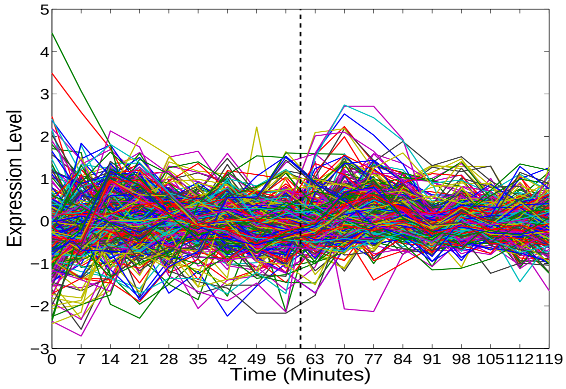

Figure 2: Raw expression curves for 542 cell-cycle related genes.

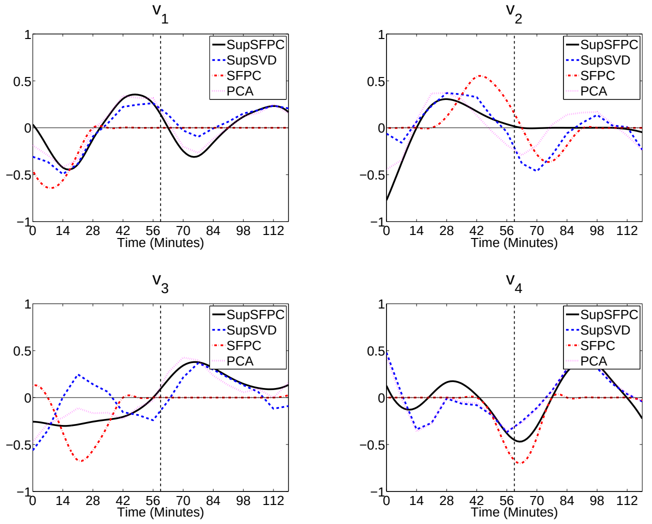

Figure 3: The first 4 loading vectors estimated from SupSFPC, SupSVD, SFPC, and PCA.

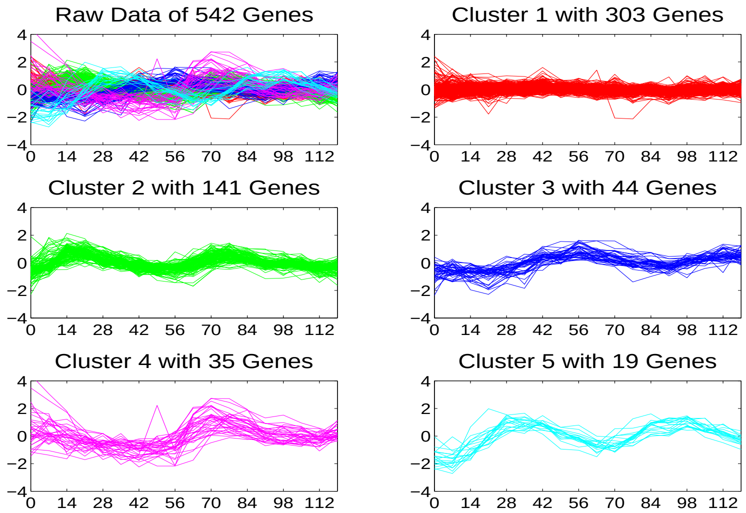

Figure 4: Raw gene expression curves clustered into 5 groups based on SupSFPC scores.

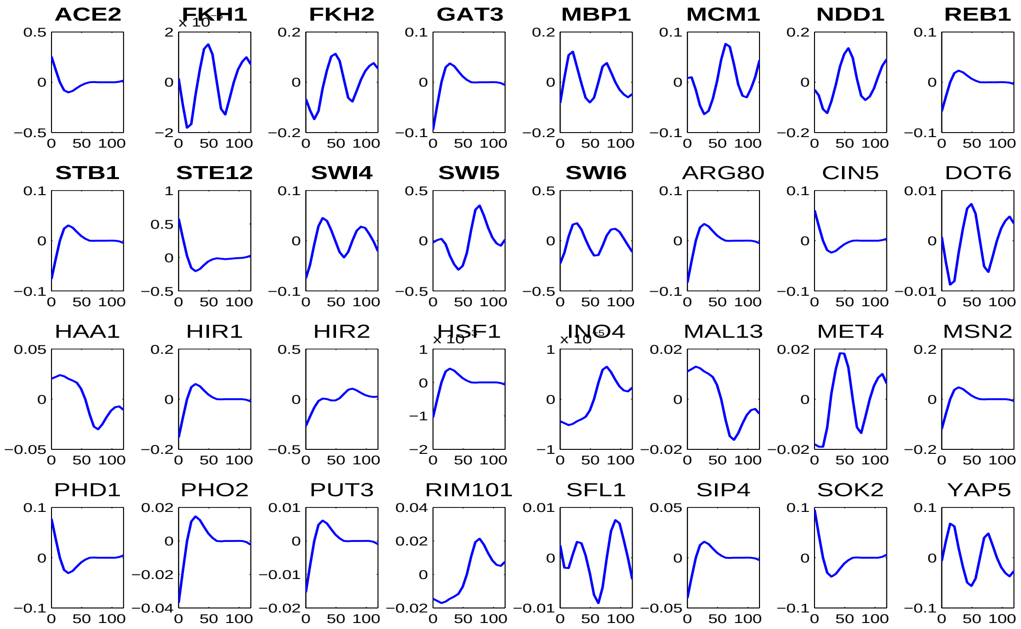

Figure 5: TF activities identified by SupSFPC that are related to yeast cell cycles. The first 13 (with bold titles) are experimentally confirmed TFs that are related to cell cycles.

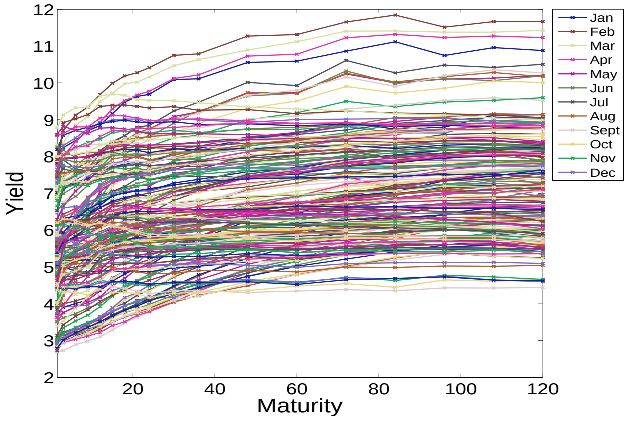

Figure 6: Raw yield curves of different maturities from January 1975 to December 2000.

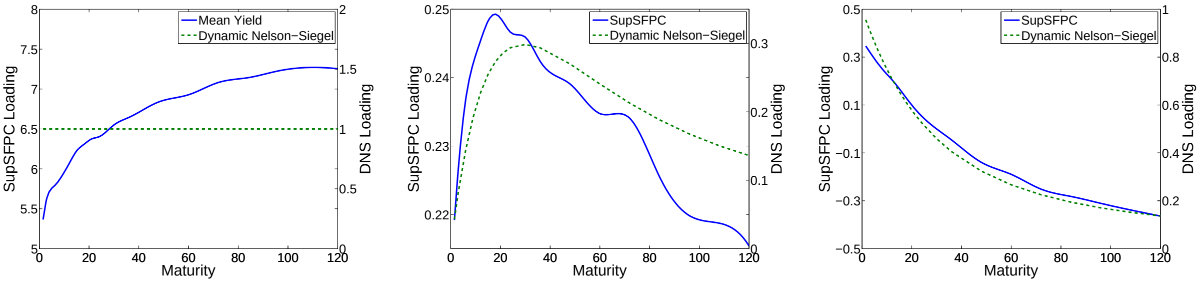

Figure 7: Loadings estimated from SupSFPC (solid line) and the pre-specified loadings from the dynamic Nelson-Siegel model (dashed line).

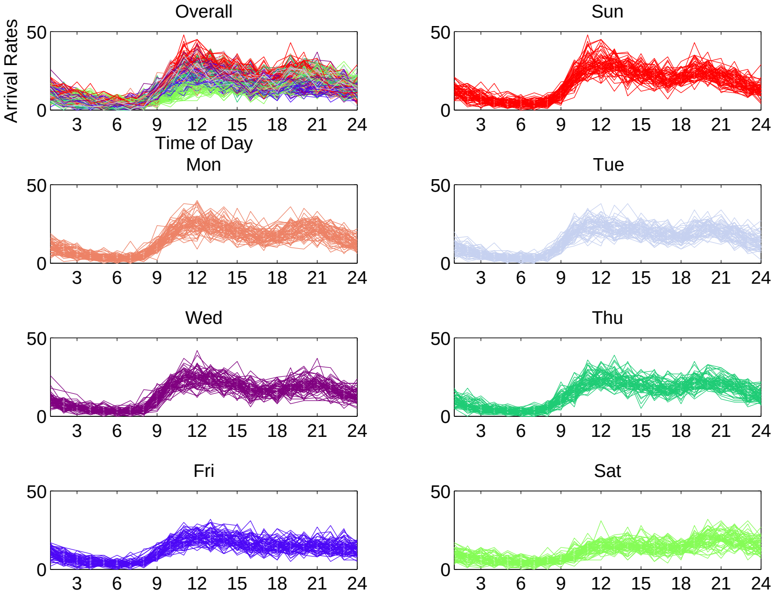

Figure 8: Raw arrive rate curves of the hospital ER visit data. The first panel shows the overall curves for 417 consecutive days; the other panels show arrival curves on different days of the week respectively.

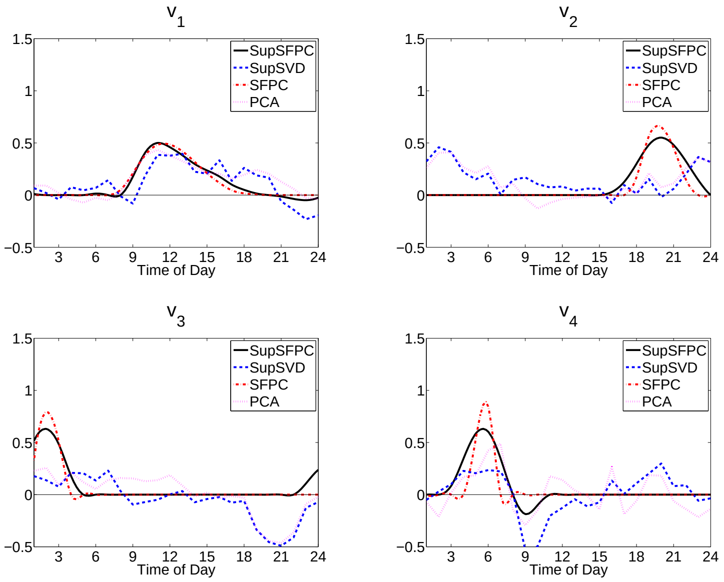

Figure 9: The first 4 loading vectors estimated from SupSFPC, SupSVD, SFPC, and PCA.

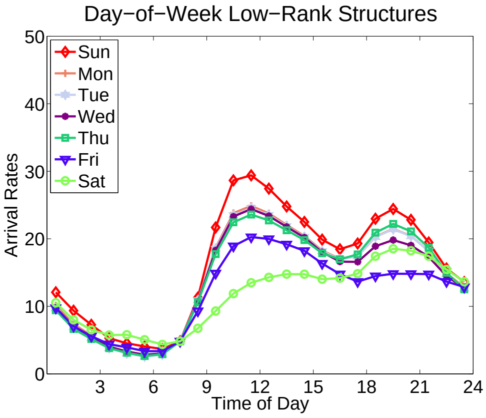

Figure 10: The day-of-week structure identified by SupSFPC.

Table 1: Median(MAD) of performance measurements for different settings based on 100 simulation runs. *Note: In Case 2, Case 4, Case 6 and Case 8, we deliberately set the sparsity and smoothness parameters in SupSFPC and SFPC to zero to improve performances. Practically, we usually know when smoothness and sparsity are needed.*

			SupSFPC	SupSVD	SFPC	PCA
$r = 1$	Case 1 (SupSFPC)	$MSE_{\mathbf{V}}$	**.44e-4** (1.0-5)	1.0e-4 (.86e-5)	2.8e-4 (3.3e-5)	1.0e-4 (.86e-5)
		$Angle_{\mathbf{V}}$	**3.8** (.45)	5.8 (.24)	9.6 (.55)	5.9 (.24)
		$MSPE_{\mathbf{U}}$	**1.0** (.07)	**1.0** (.06)	2.0 (.14)	2.1 (.14)
		$MSPE_{\mathbf{UV}^T}$	**.72e-2** (.60e-3)	.99e-2 (.60e-3)	2.3e-2 (1.5e-3)	1.5e-2 (.80e-3)
	Case 2 (SupSVD)	$MSE_{\mathbf{V}}$	**1.1e-4** (1.0e-5)	**1.1e-4** (1.0e-5)	**1.1e-4** (1.1e-5)	**1.1e-4** (1.1e-5)
		$Angle_{\mathbf{V}}$	**5.8** (.29)	**5.8** (.28)	5.9 (.29)	5.9 (.29)
		$MSPE_{\mathbf{U}}$	**1.0** (.07)	**1.0** (.07)	2.0 (.11)	2.0 (.11)
		$MSPE_{\mathbf{UV}^T}$	**1.0e-2** (5.6e-4)	**1.0e-2** (5.6e-4)	1.5e-2 (7.3e-4)	1.5e-2 (7.3e-4)
	Case 3 (SFPC)	$MSE_{\mathbf{V}}$	**.20e-3** (.69e-4)	.60e-3 (.81e-4)	7.4e-3 (9.2e-4)	.60e-3 (.82e-4)
		$Angle_{\mathbf{V}}$	**7.6** (1.5)	14 (0.9)	51 (3.5)	14 (1.0)
		$MSPE_{\mathbf{U}}$	**1.9** (.18)	2.0 (.15)	4.5 (.65)	2.3 (.16)
		$MSPE_{\mathbf{UV}^T}$	**1.3e-2** (1.2e-3)	1.5e-2 (.80e-3)	6.4e-2 (3.7e-3)	1.7e-2 (.90e-3)
	Case 4 (PCA)	$MSE_{\mathbf{V}}$	**5.8e-4** (7.1e-5)	**5.8e-4** (7.3e-5)	**5.8e-4** (7.1e-5)	**5.8e-4** (7.1e-5)
		$Angle_{\mathbf{V}}$	**14** (.88)	**14** (.88)	**14** (.87)	**14** (.87)
		$MSPE_{\mathbf{U}}$	**2.0** (.17)	**2.0** (.16)	2.3 (.17)	2.3 (.17)
		$MSPE_{\mathbf{UV}^T}$	**1.5e-2** (1.0e-3)	**1.5e-2** (1.1e-3)	1.7e-2 (1.2e-3)	1.7e-2 (1.2e-3)
$r = 3$	Case 5 (SupSFPC)	$MSE_{\mathbf{V}}$	**.60e-3** (.10e-3)	3.7e-3 (1.4e-3)	2.0e-3 (.40e-3)	3.3e-3 (.90e-3)
		$Angle_{\mathbf{V}}$	**15** (1.5)	18 (.87)	22 (1.5)	18 (.92)
		$MSPE_{\mathbf{U}}$	**1.9** (.12)	6.0 (2.5)	2.7 (.19)	6.0 (1.5)
		$MSPE_{\mathbf{UV}^T}$	**3.0e-2** (2.2e-3)	4.9e-2 (2.0e-3)	5.8e-2 (4.7e-3)	6.0e-2 (2.6e-3)
	Case 6 (SupSVD)	$MSE_{\mathbf{V}}$	**1.8e-3** (.20e-3)	3.9e-3 (1.6e-3)	3.6e-3 (1.0e-3)	3.6e-3 (1.0e-3)
		$Angle_{\mathbf{V}}$	**18** (1.1)	**18** (1.0)	**18** (1.0)	**18** (1.0)
		$MSPE_{\mathbf{U}}$	**2.2** (.23)	6.6 (3.3)	6.4 (1.6)	6.4 (1.6)
		$MSPE_{\mathbf{UV}^T}$	**4.9e-2** (1.8e-3)	**4.9e-2** (1.7e-3)	6.0e-2 (2.0e-3)	6.0e-2 (2.0e-3)
	Case 7 (SFPC)	$MSE_{\mathbf{V}}$	**1.4e-3** (.20e-3)	5.3e-3 (.70e-3)	4.5e-3 (1.7e-3)	5.3e-3 (.60e-3)
		$Angle_{\mathbf{V}}$	**19** (1.3)	32 (2.0)	35 (7.5)	32 (2.0)
		$MSPE_{\mathbf{U}}$	**2.5** (.17)	3.9 (.68)	3.2 (.49)	4.5 (.58)
		$MSPE_{\mathbf{UV}^T}$	**3.6e-2** (.23e-2)	5.9e-2 (.26e-2)	6.5e-2 (1.2e-2)	6.8e-2 (.32e-2)
	Case 8 (PCA)	$MSE_{\mathbf{V}}$	**5.3e-3** (7.2e-4)	**5.3e-3** (6.8e-4)	**5.3e-3** (7.1e-4)	**5.3e-3** (7.1e-4)
		$Angle_{\mathbf{V}}$	**33** (2.0)	**33** (2.2)	**33** (2.1)	**33** (2.1)
		$MSPE_{\mathbf{U}}$	**3.7** (.51)	3.9 (.55)	4.3 (.49)	4.3 (.49)
		$MSPE_{\mathbf{UV}^T}$	**5.8e-2** (2.9e-3)	**5.8e-2** (3.0e-3)	6.8e-2 (3.0e-3)	6.8e-2 (3.0e-3)