A Functional Dynamic Factor Model

by
Spencer Eric Hays

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2011

Approved by:

Haipeng Shen, Advisor

Chuanshu Ji, Committee Member

Young K. Truong, Committee Member

Jianhua Huang, Committee Member

Harry Hurd, Committee Member

© 2011
Spencer Eric Hays
ALL RIGHTS RESERVED

ABSTRACT

SPENCER ERIC HAYS: A Functional Dynamic Factor Model.
(Under the direction of Haipeng Shen.)

Functional data analysis is a burgeoning area in statistics. However, much of the literature to date deals primarily with methods for collections of independent functional observations, which are not well suited to application with time series of curves. In this paper, a functional time series model is proposed for the purpose of curve forecasting. The model is a synthesis of ideas stemming from traditional functional data analysis and from dynamic factor analysis. The primary contribution of the model is that it accounts for both smooth functional behavior and dynamic correlation in time series of curves. Specifically, it is hypothesized that observed data represents a discrete sampling of an underlying smooth time series of curves. These curves themselves are functions of unobserved dynamic factors with corresponding factor loadings that take the form of a functional curve; the model is thusly named the functional dynamic factor model (FDFM). Based on distributional assumptions regarding the observed data and unobserved factors, maximum likelihood estimation is proposed. To ensure that the estimated factor loading curves do represent smooth curves, roughness penalties are added to the likelihood, resulting in a penalized likelihood. The unobserved time series factors are considered as a problem of missing data for which the Expectation Maximization algorithm (EM) is well suited as the tool of estimation. As part of the EM, generalized cross validation (GCV) is used to select the optimal smoothing parameter corresponding to each smooth factor loading curve. As an iterative estimation procedure, the EM in this context can be computationally intensive. To this end, several computational efficiencies are derived to expedite estimation. Model performance is illustrated through simulation and through rather varied applications, including industrial, climatological and financial settings. Based on the simulation studies, the FDFM results in accurate parameter estimation as compared to those of benchmark models. For both simulated and applied data, forecast results for the FDFM are comparable to results from other models used in their respective applied area. Finally, several extensions of the functional dynamic factor model and areas of future research are described.

Keywords: Functional data analysis, dynamic factor models, EM algorithm, functional time series, call center forecasting, yield curve forecasting.

ACKNOWLEDGMENTS

I would like to acknowledge the contributions of, and my gratitude to, my advisor Professor Haipeng Shen. It is an understatement to say I could not have done this without his help. Of course the research content itself stemmed from his own research interests. Yet in addition, the fulfillment of this dissertation not only required his expertise in the subject matter but also his ability to serve as an exemplary mentor; and having done so amidst a myriad other responsibilities. It has been a true honor to work with him, I am proud to have done so, and I only hope I represent him well in my own research going forward.

I would also like to thank my committee members: Professors Chuanshu Ji, Young Truong, Jianhua Huang and Harry Hurd. I was fortunate to have a committee with members of such varied research interests; and with each a command of knowledge in both breadth and depth of subject matter area. This dissertation greatly benefitted from the confluence of their input, and of course their support. Their approval of the research proposal and subsequent recommendations set the foundation for this dissertation. I am grateful for their assistance in the completion thereof.

I thank my employers at the Department of Psychiatry: first and foremost Professor Hongbin Gu and Professor Bob Hamer. I learned my own research area from my dissertation, but from them I learned what it is to be a collaborative researcher and a career statistician: for this I am in their debt, and I can not thank them enough for the opportunity to have worked with them. I would also like to thank Janet Spear for her help and support; both with my work in Psychiatry, and for her support in all other regards. Finally I thank Ann VonHolle, Sandra Woolson, Jackie Johnson, Abby Scheer and Junghee Choi: their knowledge, support and camaraderie is unmatched; and I hope we can work together again in the future.

For my family, there is neither text allotment nor words for conveying my gratitude. My

parents, John Hays and Patricia Hays, have been constant inspiration and support. For this I thank them. Likewise to my brothers Justin Hays and Zachary Hays, I have appreciated all of their help and guidance. Also with special thanks to Zachary's wife and children; Brie Hays; Fletcher Hays and Kate Hays, respectively.

Finally, to Rebekah: This dissertation is a far second to you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Consider the problem of properly staffing a call center for, say, a major East coast credit card company. Call volumes fluctuate week to week, day to day, and even hour to hour. The danger of over-staffing is obvious: an overstaffed call center loses as much money as the idle employees are paid in hourly wages to (not) answer phones. The costs of an understaffed call center are more subtle but perhaps even more expensive in terms of customer affinity and attrition. By under-staffing, wait times for customers in queue are increased. In fact, call times are increased as representatives must take additional time to explain and apologize for the wait. As a result, customers can become dissatisfied with their product and cancel. Further increased demand on the fewer representatives leads to poor quality control. It is clear then, that non-optimal staffing is rather costly in either case. The solution is a forecast model that meets the specific needs of the call center environment.

An optimal staffing forecast model must take into account not only weekly call volume fluctuations, but also how volumes vary throughout the course of any given business day. The most mechanical way to develop a forecast model is to use the standard univariate auto-regressive, moving-average framework (ARMA). Supposing the data consist of quarter hourly volume measurements over dozens of weeks, an ARMA model could then forecast future call volumes by the hour, week, month, etc.

However, using the ARMA framework, the developer would no doubt encounter multiple periodic effects within the data that would need to be accounted for. It has already been alluded to that volumes fluctuate in a regular pattern throughout the day, but data like this also exhibit a day of the week effect. Exceptionally long time series like this might also display the multiple seasonal components. In fact this is typical with this type of data (Taylor, 2008). Further, with the smallest data unit being a quarter hourly call volume, an ARMA model may be expected to forecast tomorrow's volumes reasonably well. But for "longer“ forecast horizons like even a few days ahead, the ARMA forecast would exhibit the usual mean-reversion seen in these models. Resulting in either over or under-staffing and the aforementioned costs associated with each of these. A better method to account for multiple periodicity would be to consider periodic auto-regressive models (PAR) (Hurd and Miamee, 2007). A connection to those types of models will be made in Chapter 6, however here, the method proposed is of a functional nature. It is a method capable of forecasting both within day (intra-day) call volumes and inter-day call volumes. A method that, similar to a PAR model, accounts for the multiple periodic components evident in the data. Consider the following proposed model, beginning with the actual data that motivated it.

Call volumes are recorded every fifteen minutes throughout the business day, resulting in 68 intra-day intervals. This data is collected over the course of 210 days. Because of the high frequency of the intra day call volumes, it is of interest to model these as a discrete sampling of a continuous process. That is, to picture a functional relationship for intra day call volumes on a given day as some smooth underlying curve plus a noise or error component (to account for departures from smoothness). Figure 1.1, panel (a) displays a portion of the actual call volume data to illustrate the idea of modeling the data as a time series of curves. Panel (b) illustrates an example of the functional view of the data. However, for a given time interval on each day it is also rather plausible that the volumes from that interval are related from one day to the next, as depicted in panel (c) of 1.1.

From a statistical standpoint, this is straightforward to formulate; the goal then is estimate and forecast a time series of curves. Ramsay and Silverman (2002, 2005) provide a thorough treatment of functional data analysis (FDA) in both theory and through application. However, FDA in general is an area still nascent in development, and these applications deal primarily with collections of *independent* curves, with a few exceptions as described in Shen (2009). Therefore, pursuing the traditional FDA approach of independent curves ignores the rich cor-

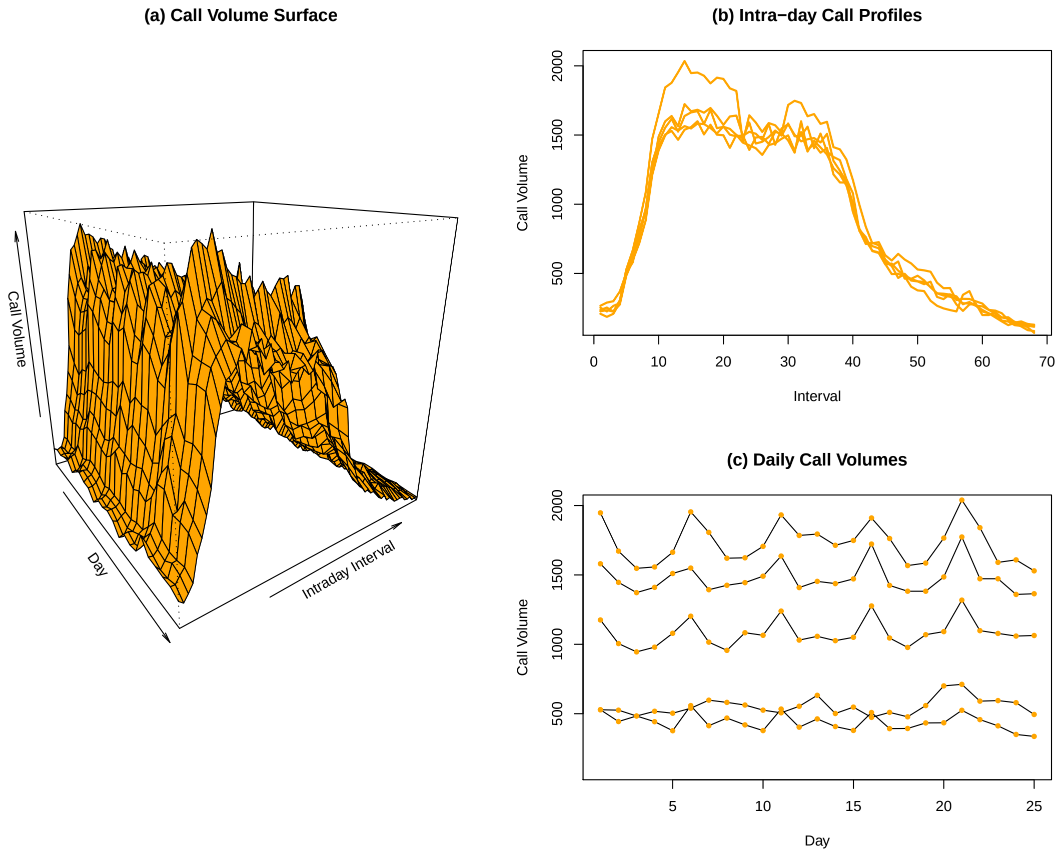


Figure 1.1: Example of Dynamic Functional Data. Three views of a functional time series. Panel (a) illustrates the surface created by plotting intraday call volumes by day. Panel (b) depicts cross sections of the data at five time points (days), resulting in a view of five intraday call volume profiles. These are hypothesized to consist of a smooth underlying curve and an error component that accounts for departures from smoothness. Panel (c) shows the daily time series of call volumes within a sample of five different 15 minute intervals.

relational structure inherent in data that occurs over time. An alternative is to simply treat the data as multivariate time series.

FDA provides a framework to work with the functional observations of curves. However, for each intraday interval there is also a daily time series of call volumes for that interval. This is essentially a multivariate time series, and a very large one: the motivating data set contains 210 days of data with 68 intervals each day. Modeling all 68 intervals jointly is intractable; even an order one unrestricted vector auto regression (VAR(1)) for 68 series would require a 68×68 coefficient matrix. In terms of a restricted VAR model, developing meaningful linear restrictions on 68 series may make a restricted VAR just as unwieldy. Aggregating the data into coarser intraday intervals would allow a method like vector auto-regressions directly, though at the expense of a loss of granularity in the intra day call volume profile. Further, the finer fifteen minute intervals are industry practice.

Therefore, consider an alternative approach. In the interest of dimension reduction, it would be helpful if the behavior of the 68 related time series could be explained via a smaller set of variables. Then that smaller set of variables could be modeled as a more manageable multivariate time series. This aspect of the problem lends itself to the realm of dynamic factor analysis (DFA) where the observed multivariate time series data can be explained via a smaller multivariate set of unobserved or latent factors, also following a time series process and related to the observed data through coefficients known as *factor loadings* (Basilevsky, 1994). So-called dynamic factor models (DFMs) first appeared in the literature independently by Geweke and Singleton (1981), Engle and Watson (1981) and Molenaar (1985), and have enjoyed some success in problems involving either economics or psychology.

Keeping in mind the desire to model the intra day profile as functional, the problem here is that there is nothing in the formulation of the typical dynamic factor model that constrains any part of it to represent a smooth function. Diebold and Li (2006) proposed a factor model with smooth parametric curves as loading coefficients for the purpose of yield curve forecasting. However, those curves were pre-specified and not part of the estimation process, so therefore the method is unlikely to be of use in other applied settings. Using the same yield data, Bowsher and Meeks (2008) formulated the curve forecasting as a cointegrated vector auto regression approach with natural cubic splines (NCS) to capture the functional behavior of the yield curve. The method was computationally intensive in that selection of the appropriate knot locations for the NCS's used a goodness-of-fit measure on every possible combination of 3 to 4 internal knots out of 34 possible choices to determine the best model. In the latter case of four knots this requires fitting of 46,376 candidate models.

Ideally, a model to estimate and forecast functional time series like the call data should be both elegant in specification and estimation, and should further capture both the functional behavior and the time series behavior of the data. Therefore, what is proposed here is a synthesis of ideas stemming from both FDA and DFA to create a new model for the purpose of forecasting time series of curves. Presented in this dissertation is the specification of a *Functional* Dynamic Factor Model (FDFM). The FDFM retains the idea of dimension reduction of the observed data into a more manageable set of unobserved time series factors, but further specifies that the factor loadings for each factor form a smooth curve. The hypothesis is that the observed data on a given day is the sum of a smooth underlying curve plus noise. The former component is then the sum of unobserved factors and their corresponding smooth factor loading *curves*.

Outside of the call volume setting, another application of the FDFM model is in reference to yield curve forecasting from Diebold and Li (2006) and Bowsher and Meeks (2008). Specifically, given a time series of zero coupon bond yields of multiple maturities (3, 6, 12 months etc.), it is of interest to forecast yields not only for bonds of the observed maturities but also for the entire curve or spectrum of maturities. This type of data lends itself exactly to the FDFM formulation. It further illustrates the importance of viewing the data unit as a curve and not just a collection of discrete data points. In an applied sense, though a particular maturity is not observed or even exists, it is essential to investors to have a yield measure for effectively any maturity in order to evaluate rates of return on portfolios of bonds with varying maturities. The functional perspective is advantageous from a statistical point of view as well. For example, modeling high frequency data as smooth curves results in a method robust to outliers (not that extreme events are unimportant, rather just in this context there is not a particular interest in them).

Although the motivation for the development of the model was based upon the call data, it can just as easily be applied to other time series. Consider for example, strongly seasonal climatological data such as sea surface temperature (SST) in the South Pacific. The El Nino phenomenon is a well documented cycle that strongly influences sea surface temperatures. Given a time series of monthly average SSTs, a natural method for estimating and forecasting the data is to model it simply as a univariate seasonal ARIMA model with period twelve. A better method may be to consider a PAR formulation given the strong seasonal nature of the data. However in the present context, another method is to consider the seasonal pattern as a sampling of an underlying smooth curve. That is, to reconstruct the long series of monthly single observations as an annual series of multivariate data with twelve observations per year. Then for each multivariate observation treat those twelve observations as a realization of a smooth underlying seasonal cycle plus noise. Put another way, the twelve months represent the discrete sampling from the underlying curve and then the data can be viewed as an *annual* time series of curves that represent the cycle. Besse et al. (2000) used versions of functional autoregressive models (FARs) to forecast the data, but the data can just as easily be modeled within the FDFM framework.

Specification of the FDFM begins within the typical dynamic factor model framework. Errors are assumed to be normally distributed, and in the cases presented below, independent. The factors themselves are represented by low dimensional time series. In this dissertation, the cases presented are independent auto-regressive time series with no moving average components. This permits a straight forward derivation of a likelihood function. To ensure that the estimated factor loading curves are indeed curves, roughness penalties with smoothness parameters are added to the log-likelihood corresponding to each loading curve. By doing so, maximization of the penalized log-likelihood is a balance between goodness-of-fit and the smoothness of the loading curves. The smoothing parameters themselves are assumed as fixed but unknown, so a generalized cross validation approach is used to select those.

It is worth re-emphasizing that the factors are latent. Therefore, to estimate the model requires not only estimation of the factor loadings, but also of the unobserved factors themselves. This poses a difficult problem. One method is to approach this as a problem of missing data; that the latent factors are missing. A useful algorithm for working with missing data is the

Expectation Maximization algorithm (EM) first introduced by Dempster et al. (1977). Meng and Rubin (1993) further derived the theoretical and convergence properties of the EM. The EM is an iterative procedure that begins with the specification of initial values for the factors and factor loadings, then each iteration of the EM involves an *E-step* and an *M-step*. In the E-step, conditional expectations of the factors given the observed data are used in place of the latent factors. In the M-step these conditional expectations are used in the maximum likelihood solutions (MLEs) to solve for the factor loading curves and other model parameters.

The remainder of this dissertation is organized thusly: Chapter 2 presents and develops the specific model, and the method by which it is estimated. Chapter 3 details at length the mathematical and computational aspects of the concepts introduced in Chapter 2. Chapter 4 provides derivation of the explicit form for the cross-validation expression, and demonstrates that the factor loading curves form natural cubic splines. Chapter 5 illustrates model performance on simulated data including comparisons with competing methods. Chapter 6 introduces an innovative simulation design, and uses simulated data to illustrate methods for time series identification for the dynamic factors, determination of the number of factors, and bootstrap-based inference for forecasts and parameters. These methods are described therein. Chapter 7 presents a comprehensive analysis of true yield curve data. In Chapter 8, real data from two other applied areas are used to estimate the FDFM. Forecasts are calculated and compared with out of sample data to assess goodness-of-fit. Comparisons with corresponding models from their respective application are analyzed as well. Finally, Chapter 9 highlights the key components of the model and its performance and discusses possible extensions and future analyses regarding the FDFM.

Chapter 2

Functional Dynamic Factor Model

This chapter develops the model by beginning from a functional perspective of the process generating the data. Observed data is treated as a sampling of the underlying smooth functions and forms a data matrix. From this point the dynamic factor model framework is implemented hypothesizing the larger data matrix is composed of a smaller set of unobserved dynamic factors following independent time series processes, and corresponding factor loadings. Imposing structure on the factor loadings so that they form smooth curves is the defining feature of the FDFM and relates the typical DFM model back to the functional domain.

To estimate the model a likelihood function can be derived based on the error assumptions. Then conditions on the proper amount of smoothness in the factor loadings facilitate a penalized likelihood expression. A detailed description of the use of the Expectation Maximization algorithm to estimate the model follows, including specific solutions for the maximum likelihood estimates (MLEs). Next, certain computational efficiencies are highlighted in regards to implementation. Finally, some alternative models are discussed.

2.1 The Model

Development of the model begins with a description of the data and its associated notation. Consider a time series of curves $\{x_i(t) : t \in \mathrm{T}; i = 1, \ldots, n\}$, where T is some continuous interval and i indexes discrete time. It is hypothesized that each curve is composed of a smooth underlying curve, $y_i(t)$; plus an error component, $\epsilon_i(t)$:

$$x_i(t) = y_i(t) + \epsilon_i(t). \tag{2.1}$$

The purpose of this dissertation is to develop a viable model capable of forecasting the entire smooth curve for some future date: $y_{n+h}(t)$ with $h > 0$, of course.

A good example of this type of process is the yield data introduced in the first chapter. Here $x_i(t)$ would represent the yield to maturity as of date i for a (zero coupon) bond of maturity t. In financial terms, it is useful to investors to have information about the entire continuous yield curve. But of course in practice, yields are only observed for a discrete class of maturity horizons; 3 months, 6 months and so forth. This is also the case of the general problem in this dissertation: that only distinct data points are observed, yet the intent is to work with and forecast the entire curve over time.

Specifically, for $t \in \mathrm{T}$, consider a sample of discrete points $\{t_1, t_2, \ldots, t_m\}$ with $t_j \in \mathrm{T}$ for $j \in \{1, \ldots, m\}$. Then denote:

$$x_{ij} \equiv x_i(t_j).$$

In other words, from an FDA perspective, the observed data x_{ij} is a point sample from the process $x_i(t)$ at the specific value $t = t_j$. With the discrete values of the data, they can be collected into a data matrix which is the starting point for the Factor Analysis component of the model specification.

Specifically, for the observed data $\{x_{ij} : i = 1, \ldots, n; j = 1, \ldots, m\}$, let i index the row, and j index the column of a data matrix $\mathbf{X}_{n\times m}$ so that the i, jth element of $\mathbf{X}$ is x_{ij}. In reference to the yield example, the rows of $\mathbf{X}$ correspond to yield curves for at a fixed date; the columns are the time series of yield for a specific maturity. Figure 2.1 is a visual representation of a data matrix $\mathbf{X}$.

2.1.1 The Classical Dynamic Factor Model

To develop a forecasting model for the data, the (column) rank of $\mathbf{X}$ is too large to apply multivariate techniques, and the functional aspect of the data still must be accounted for. The

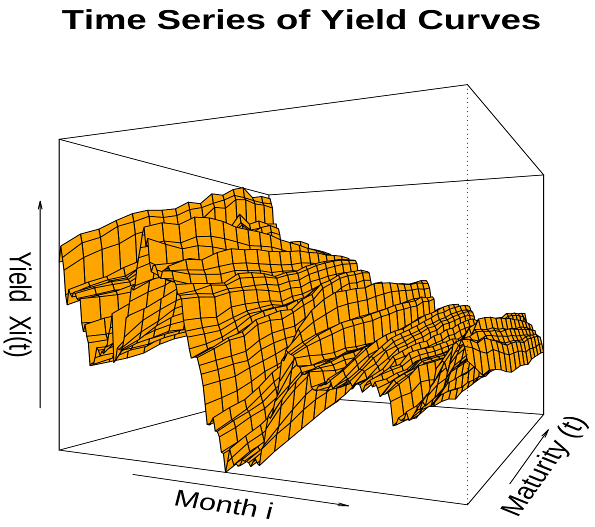


Figure 2.1: Example of Dynamic Functional Data. Data for yields x_{ij} on all observed maturities t_j at all dates i is plotted.

first step then is to reduce the dimensionality of the problem via the use of dynamic factor modeling. The idea is that the behavior of a set of m observed variables can be explained through the behavior of a much smaller, though unobserved, set of K variables or factors and their corresponding coefficients called factor loadings. To capture the dynamic nature of the data it is postulated that the latent factors themselves follow a stochastic process. Later constraints will be placed on the model so that the factor loadings form smooth curves.

Following the notation of Pena and Box (1987), denote the rows of $\mathbf{X}$ as $m \times 1$ *column* vectors $\mathbf{x}_i$, the general dynamic factor model can be represented in the following form:

$$\begin{aligned} \mathbf{x}_i &= \mathbf{F}'_{m\times K}\boldsymbol{\beta}_i + \epsilon_i, \qquad (2.2) \\ \epsilon_i &\sim N_m(0, \Sigma_\epsilon), \end{aligned}$$

where $\mathbf{x}_i$ is the vector of observed data at time i, $i = 1, \ldots, n$; $\mathbf{F}$ is a fixed but unknown $K \times m$ (rank K) matrix of factor loadings for the unobserved K dynamic factors $\boldsymbol{\beta}_i = [\beta_{i,1}, \ldots, \beta_{i,K}]'$;

and $\epsilon_i = [\epsilon_{i,1}, \ldots, \epsilon_{i,m}]'$ is a Gaussian error vector with a full rank covariance matrix Σ_ϵ.

The factors are hypothesized to follow a K-dimensional vector auto-regressive moving average (VARMA(P,Q)) process. Using the lag or back-shift operator $L^p \boldsymbol{\beta}_i = \boldsymbol{\beta}_{i-p}$ (Hamilton, 1994), the multivariate factor time series can be modeled as

$$\begin{aligned} \Phi(L)\boldsymbol{\beta}_i &= \Theta(L)v_i, \\ v_i &\sim N_K(0, \Sigma_v), \end{aligned} \tag{2.3}$$

with

$$\begin{aligned} \Phi(L) &= \mathbf{I}_K - \Phi_1 L - \ldots - \Phi_P L^P, \\ \Theta(L) &= \mathbf{I}_K - \Theta_1 L - \ldots - \Theta_Q L^Q. \end{aligned}$$

The $K \times K$ Φ and Θ coefficient matrices are assumed to be such that $\boldsymbol{\beta}_i$ is a covariance stationary vector time series. Finally, identification of the model requires further assumptions typically on either the structure of the covariance matrix Σ_v or on the properties of the factor loadings; the particular restriction employed here is discussed in the next section.

2.1.2 The Functional Dynamic Factor Model

Two types of additional assumptions are added to (6.1) and (2.3) to form the Functional Dynamic Factor Model; those that make this a functional model and those that are intended to simplify the development of the FDFM. The latter of these are discussed first.

These additional assumptions may be relaxed at a later time in order to provide a more general model framework. They are imposed here if only to provide a more parsimonious endeavor into this new, exciting class of models (see Chapter 9 for possible extensions). Specifically:

1. The K dynamic factors $\boldsymbol{\beta}_i$ have no moving average components ($\Theta(L) = \mathbf{I}_K$).

2. The factors are independent so that the coefficient matrices in $\Phi(L)$ are diagonal, reducing the VARMA(P,Q) process to K independent univariate covariance stationary AR(p) processes.

3. $\Sigma_\varepsilon = \sigma^2 \mathbf{I}_m$.

4. The innovations ϵ_i and v_{i+h} are uncorrelated at all leads and lags ($h = 0, \pm 1, \pm 2, \ldots$) for all $k \in \{1, \ldots, K\}$; all $j \in \{1, \ldots, m\}$.

5. For the purposes of the proposed model it will be assumed that the loading vectors in $\mathbf{F}$ are orthonormal: $\mathbf{F}\mathbf{F}' = \mathbf{I}_k$.

6. (Optional) Due to the simplified univariate AR structure and for additional flexibility, in place of a constant, non-stochastic regressors can be considered in the factor time series.

With these simplifying assumptions the FDFM can be expressed in the following scalar manner. Denote f_{kj} as the kjth element of $\mathbf{F}$. Then implementing these assumptions, together with (6.1) and (2.3), yields the model:

$$\begin{cases} x_{ij} = \sum_{k=1}^{K} \beta_{ik} f_{kj} + \epsilon_{ij}, \quad \epsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma^2) \\ \beta_{ik} - A_{ik}\mu_k = \sum_{r=1}^{p_k} \varphi_{rk}(\beta_{i-r,k} - A_{i-r,k}\mu_k) + v_{ik}, \quad v_{ik} \overset{i.i.d.}{\sim} N(0, \sigma_k^2) \\ E v_{tk}\epsilon_{sj} = 0 \quad \text{for} \quad t, s = 1, \ldots, n; \quad k = 1, \ldots, K; \quad j = 1, \ldots, m, \end{cases} \tag{2.4}$$

where for date i, the $1 \times d_k$ regressor vector for the kth factor is A_{ik}, with $d_k \times 1$ coefficient vector μ_k. Going forward, this will be denoted as an *FDFM(K,p)* model, which refers to the K factors and the order $p = \max\{p_1, \ldots, p_K\}$ of the auto-regressive factors.

The next class of assumptions are the ones critical to converting the traditional dynamic factor model to a functional dynamic factor model; those regarding the smoothness of the factor loadings. In the classical model, outside of the normalizing condition the loadings are otherwise unconstrained and represent distinct coefficients for the dynamic factors. However in the present setting, the loadings are further hypothesized to be a sampling or “discretization” of a deterministic, continuous, yet unobserved smooth function/curve. That is, the functional dynamic factor framework provides another means by which to model time series of curves.

Specifically, let

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_K \end{bmatrix} \tag{2.5}$$

with $\mathbf{f}_k = [f_{k1} \dots f_{km}]$ so that $\mathbf{f}_k$ is the sequence of factor coefficients corresponding to the kth factor. Then based on the fifth assumption above, this implies

$$\mathbf{f}_k \mathbf{f}_l' = \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

Recall the original formulation of the observed data as a sample of a continuous process; that the observed data x_{ij} is a point sample from the process $x_i(t)$ at the specific value $t = t_j$. Since $x_{ij} = \sum_{k=1}^{K} \beta_{ik} f_{kj} + \epsilon_{ij}$, it is proposed that the factor loadings are themselves discrete samples from continuous, unobserved *factor loading curves.* That is, that $f_{kj} \equiv f_k(t_j)$ and that each $\mathbf{f}_k$ represents the sampled curve corresponding to the kth factor. Thus, while the dynamic factors represent a time series at discrete time points $i \in \{1, \dots, n\}$ the factor loadings are a realization of a continuous function evaluated at m distinct points. It is precisely here where the synthesis of functional data analysis and dynamic factor models occurs.

$$x_i(t_j) = \sum_{k=1}^{K} \beta_{ik} f_k(t_j) + \epsilon_i(t_j). \tag{2.7}$$

This closely resembles the formulation of a classical dynamic factor model if not for the functional assumption on the factor loadings.

2.2 The Joint Distribution

To use the Expectation Maximization algorithm to estimate the model requires first the specification of the joint distribution in order to derive a likelihood expression to maximize. A modified likelihood is derived to include conditions for smoothness of the factor loading curves $\mathbf{f}_k$ which is discussed in the next section, followed by a rigorous development of the EM implementation.

The remainder of the dissertation includes many mathematical derivations and thus the introduction of some matrix notation will be useful hereafter. The model $x_{ij} = \sum_{k=1}^{K} \beta_{ik} f_{kj} + \epsilon_{ij}$ is represented in matrix form as

$$\mathbf{X}_{n\times m} = \mathbf{B}_{n\times K}\mathbf{F}_{K\times m} + \epsilon_{n\times m}, \tag{2.8}$$

where $\mathbf{B} = \{_m\beta_{ik}\} = [\boldsymbol{\beta}_1 \ldots \boldsymbol{\beta}_K]$ and $\boldsymbol{\beta}_k = [\beta_{1k} \ldots \beta_{nk}]'$.

For the moment, suppose the values of the factor time series are known. If this is the case then finding the joint distribution of the observed data and the factors is a fairly straightforward exercise and consists of the distribution for the factor time series and the distribution for the observed data. That is, the distribution of **X** and **B** is found by finding the conditional distribution of **X** given **B**, and the unconditional distribution of **B**. The latter is determined using the familiar properties of univariate autoregressive time series which aids in the derivation of the former.

2.2.1 Distribution for the factor time series

With the independence assumption for the factor time series from Section 2.1.2, the joint distribution for the factor time series is just the product of the univariate time series. Each univariate distribution is then the product of an unconditional distribution for the first p values of the factor and the distribution conditioned on those p values.

For notational convenience and without loss of generality, for the following derivation it is assumed that:

1. no regressors are included in the time series other than an intercept for each of the K factors.

2. It is further assumed that the order of the AR(p_k) processes are the same for all factors. That is, $p_k \equiv p \; \forall k$.

Either assumption does not detract from the derivation. For the latter, admittedly it may be unrealistic to hypothesize p should be the same for all k. However, if the order of the processes is defined as $p = \max\{p_1, \ldots, p_k\}$ then it is easy to imagine that corresponding coefficients φ_{rk} are simply set to 0 for $r \leq p_k < p$. For the former assumption, the model equation for $\boldsymbol{\beta}_k$ in (2.3) simplifies to

$$\beta_{ik} = c_k + \sum_{r=1}^{p} \varphi_{rk}\beta_{i-r,k} + v_{ik} \text{ ; for } k = 1, \ldots, K.$$

Since the K factor time series are assumed to be independent, their joint distribution factors to the product of the univariate distributions for the time series:

$$f(\mathbf{B}) = f(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K) = \prod_{k=1}^{K} f(\boldsymbol{\beta}_k). \tag{2.9}$$

Each univariate distribution in the expression above can be further decomposed via conditioning on the first p observations (Hamilton, 1994). Consider the likelihood for the factor time series $\boldsymbol{\beta}_k$ as

$$\begin{aligned} f(\boldsymbol{\beta}_k) &= f(\beta_{1k}, \ldots, \beta_{nk}) \\ &= f(\beta_{1k}, \ldots, \beta_{pk}) f(\beta_{p+1,k}, \ldots, \beta_{nk} | \beta_{1k}, \ldots, \beta_{pk}) \\ &= f(\beta_{1k}, \ldots, \beta_{pk}) \left\{ \prod_{i=p+1}^{n} \frac{1}{\sqrt{2\pi\sigma_k^2}} exp\left[-\frac{(\beta_{ik} - c_k - \sum_{r=1}^{p} \varphi_{rk}\beta_{i-r,k})^2}{2\sigma_k^2} \right] \right\}, \end{aligned}$$

which can be substituted into (2.9) to obtain the likelihood for the factor time series.

2.2.2 The Joint distribution and Likelihood of X and B

The joint distribution of $\mathbf{X}$ and $\mathbf{B}$ can similarly be simplified via successive conditioning. Using the property that $\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$ and the prior result on the joint distribution of the factor time series yields the following:

Proposition 2.2.1. *Joint Distribution of* $\mathbf{X}$ *and* $\mathbf{B}$

$$\begin{aligned} f(\mathbf{X},\mathbf{B}) &= \prod_{k=1}^{K} f(\boldsymbol{\beta}_k) \times \prod_{i=1}^{n}\prod_{j=1}^{m} f(x_{ij}|\beta_{i1},\ldots,\beta_{iK}) \\ &= \prod_{k=1}^{K} f(\beta_{1k},\ldots,\beta_{pk}) \qquad (2.10) \\ &\times \prod_{k=1}^{K}\left\{\prod_{i=p+1}^{n} \frac{1}{\sqrt{2\pi\sigma_k^2}} exp\left[-\frac{(\beta_{ik}-c_k-\sum_{r=1}^{p}\varphi_{rk}\beta_{i-r,k})^2}{2\sigma_k^2}\right]\right\} \\ &\times \prod_{i=1}^{n}\prod_{j=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[-\frac{(x_{ij}-\sum_{k=1}^{K}\beta_{ik}f_{jk})^2}{2\sigma^2}\right]. \end{aligned}$$

Finally, the log-likelihood expression is achieved by applying the natural logarithm to the above joint distribution, then multiplying by -2,

$$\begin{aligned} (-2\times)\ln L &= -2\sum_{k=1}^{K}\ln[f(\beta_{1k},\ldots,\beta_{pk})] + (n-p)\sum_{k=1}^{K}\ln(2\pi\sigma_k^2) + nm\ln(2\pi\sigma^2) \qquad (2.11) \\ &+ \frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij}-\sum_{k=1}^{K}\beta_{ik}f_{kj})^2 \\ &+ \sum_{i=p+1}^{n}\sum_{k=1}^{K}\frac{1}{\sigma_k^2}(\beta_{ik}-c_k-\sum_{r=1}^{p}\varphi_{rk}\beta_{i-r,k})^2. \end{aligned}$$

For the auto regressive parameters, the independence assumption on the time series factors allows K distinct optimization problems for each of the K factors. However, note the appearance of the term

$$\sum_{k=1}^{K}\ln[f(\beta_{1k},\ldots,\beta_{pk})].$$

This is the log sum of the joint distributions for the first p time points for each factor. Optimization including this term would require numerical methods, so for ease of computation a conditional likelihood approach is employed *where* $[\beta_{1,k},\ldots,\beta_{p,k}]$ *are assumed as known/given for all* k.

Notwithstanding some miracle, optimization of the likelihood (2.11) would generally not result with estimates of the $\{\mathbf{f}_k\}$ resembling smooth curves. Therefore, some additional work must be performed to assure that this will be the case for the estimated factor loading curves;

this is addressed in the next section.

2.3 The Penalized Likelihood Expression

Simply finding the optimal solutions to the likelihood expression (2.11) falls short of fully estimating the functional DFM on two counts. First, the factor time series are unobserved. Second, there is no reason to expect that by finding the values of the factor loadings and other parameters that maximizing the expression above will in any way satisfy the prior assumption that the factor loadings represent smooth curves. The latter is discussed presently; the former in the section on estimation.

From the likelihood (2.11), a maximum likelihood solution for the factor loading curves is equivalent to a minimization of the sum of squares

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left[x_{ij}-\sum_{k=1}^{K}\beta_{ik}f_{kj}\right]^2=\sum_{i=1}^{n}\sum_{j=1}^{m}\left[x_i(t_j)-\sum_{k=1}^{K}\beta_{ik}f_k(t_j)\right]^2,$$

with respect to the functions $\{f_k(\cdot)\}$.

It is proposed to follow the roughness penalty approach of Green and Silverman (1994), where a roughness penalty with a smoothing parameter is added to the sum of squares. In the functional dynamic factor model, for each of the K factor loading curves, K roughness penalty/smoothing parameter terms are added to the sum of squares. Consider the following penalty criteria for each of the factor loading curves: $\lambda_k \int [f_k''(t)]^2\, dt$. Then the *penalized* sum of squares becomes:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij}-\sum_{k=1}^{K}\beta_{ik}f_{kj})^2+\sum_{k=1}^{K}\lambda_k\int\left[f_k''(t)\right]^2 dt.$$

These terms place a condition on the second derivative of each function $\mathbf{f}_k$. In this context this is equivalent to a condition on the curvature of the function, which specifies that on the domain of the function, it is not too "rough." The coefficient λ_k controls how strictly this condition is enforced. Put another way, each penalty term imposes smoothness, or a roughness penalty on the resulting estimate so that the discrete estimated points $\mathbf{f}_{kj}$ reasonably resemble those that would lie along the smooth underlying curve.

In practice, first and second differences are used to approximate the first and second derivatives of $f_k(\cdot)$. Green and Silverman (1994) showed these are actually equivalent in the setting of the natural cubic spline (NCS). This result is yet to be shown here, but is addressed in Chapter 9. In the meantime, just as a matter of practical implementation, first differences can reasonably approximate first derivatives and second differences can likewise approximate second derivatives. With Δ representing the difference operator $\Delta f_{kj} = f_{kj} - f_{k,j-1}$:

$$\lambda_k \int \left[f_k''(t)\right]^2 dt \quad \approx \quad \lambda_k \sum_{j=2}^{m-1} \left[\frac{\Delta^2 f_{kj}}{\Delta^2 t_j}\right]^2 .$$

So, for example, if the points t_j are evenly spaced, then $\Delta t_j = 1$, and

$$\lambda_k \int \left[f_k''(t)\right]^2 dt \quad \approx \quad \lambda_k \sum_{j=2}^{m-1} \left[f_{k,j-1} - 2f_{k,j} + f_{k,j+1}\right]^2 .$$

Coefficients from this sum can be collected in the banded matrix

$$\omega_k' \quad \equiv \quad \begin{bmatrix} 0 & & & \dots & & & 0 \\ 1 & -2 & 1 & 0 & \dots & & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & & \dots & 0 & 1 & -2 & 1 \\ 0 & & & \dots & & & 0 \end{bmatrix} .$$

Let $\mathbf{\Omega}_k \equiv \omega_k \omega_k'$; then

$$\lambda_k \sum_{j=2}^{m-1} \left[f_{k,j-1} - 2f_{k,j} + f_{k,j+1}\right]^2 = \lambda_k \mathbf{f}_k \Omega_k \mathbf{f}_k'.$$

Next, using the $vec(\cdot)$ operator, which stacks the columns of a matrix, yields

$$\begin{aligned}\sum_{k=1}^{K}\lambda_k\int\left[f_k''(t)\right]^2dt &\approx \sum_{k=1}^{K}\lambda_k\mathbf{f}_k\Omega_k\mathbf{f}_k' \\ &= vec(\mathbf{F}')'\cdot\mathbf{S}\cdot vec(\mathbf{F}'),\end{aligned}$$

where $\mathbf{S}_{mK\times mK}$ is the block diagonal matrix with $m\times m$ blocks $\lambda_k\mathbf{\Omega}_k$. Adding this term to the portion of the log-likelihood creates a roughness penalty for each of the factor loading curves so that optimal solutions for the curves will reflect the dual objectives of both finding estimates that fit the data and ensuring those estimates exhibit an appropriate level of smoothness.

Combining the log-likelihood (2.11) with the K additional penalty terms results in the Penalized Log Likelihood expression

$$\begin{aligned}PL &= -2\sum_{k=1}^{K}\ln[f(\beta_{1k},\ldots,\beta_{pk})]+(n-p)\sum_{k=1}^{K}\ln(2\pi\sigma_k^2)+nm\ln(2\pi\sigma^2) \\ &+ \frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij}-\sum_{k=1}^{K}\beta_{ik}f_{kj})^2 \\ &+ \sum_{i=p+1}^{n}\sum_{k=1}^{K}\frac{1}{\sigma_k^2}(\beta_{ik}-c_k-\sum_{r=1}^{p}\varphi_{rk}\beta_{i-r,k})^2 \\ &+ \sum_{k=1}^{K}\lambda_k\mathbf{f}_k\Omega_k\mathbf{f}_k'.\end{aligned} \tag{2.12}$$

2.4 Maximum Likelihood Estimation with EM

The next step in the development of the FDFM is the method by which to estimate the model. There are two main points in this section. The first is that with Expression (2.12) maximum likelihood (ML) can be used to estimate model parameters. The second point is that the factor time series are unobserved; thus the Expectation Maximization algorithm (EM) is used in conjunction with maximum likelihood. The following subsections first present an executive summary of the EM, next develop the the use of the EM with ML solutions and finally detail smoothing parameter selection and othe computational aspects. ML parameter solutions will be indicated by a hat ($\hat{\theta}$) over the symbol for the parameter.

2.4.1 A Brief Overview of Maximum Likelihood with the EM

Expression (2.12) facilitates the use of ML to estimate the model parameters. These are the time series slopes, intercepts, and variances for each factor, the factor loading curves, and the overall model variance; collectively these will be referred to in the set Θ:

$$\Theta \equiv \left\{ \sigma^2, \{ \bigcup_{k=1}^{K} \mathbf{f}_k, \sigma_k^2, c_k, \varphi_{1,k}, \ldots, \varphi_{p,k} \} \right\} . \tag{2.13}$$

Obviously, direct maximum likelihood estimation is only of use if there is data available with which to compute the estimates. In the present context of the functional dynamic factor model this is only partially the case as it is hypothesized the observed data $\mathbf{X}$ is a function of unobserved explanatory factors $\mathbf{B}$. Thus despite having a theoretical solution for the parameter and factor loading estimates, the problem is as yet intractable due to the latent data. Therefore, the method is to treat this as a problem of missing data; enter the EM algorithm.

First introduced by Dempster et al. (1977), the EM is a method by which to impute missing data with values based on a conditional expectation using the observed, non-missing data. Further work by Meng and Rubin (1993) showed the theoretical properties of EM estimates, including the desirable properties regarding convergence. The current setting differs due to the inclusion of the penalty terms in the likelihood and so it is as yet undetermined if those results hold here. Further discussion on this point is reserved for future work in Chapter 9.

EM estimation is an iterative procedure; inaugurated with initial values, the algorithm then oscillates between the so-called E-step and M-step. It proceeds thusly:

Step 0: Initial Values: To initialize the EM algorithm, some form of starting values are required. Many possibilities exist, but here it is proposed that the right and left singular vectors extracted via the singular value decomposition (SVD) of the data matrix $\mathbf{X}$ will provide adequate initial estimates for the factors and loadings. Details will follow, but for now from these primordial time series and curves, parameter estimates may be calculated with which to inaugurate the E-step. Section 2.4.2 briefly covers this.

The E-step: In the E-step, new values for the factor time series are calculated as conditional expectations given the observed data using the parameter estimates from either Step 0 or the previous M-step. The conditional expectations then take the place of the factor time series in the calculation of the factor loading curves and the next iteration of parameter estimates. See Section 2.4.3 for a thorough derivation.

The M-step: Based on the factor scores from the conditional expectation in the E-step, MLEs are calculated for the factor loading curves and other parameters using the ML solutions in the subsequent sections. The optimal solution for the set of $\{\mathbf{f}_k\}$ is dependent on the smoothing parameters $\{\lambda_k\}$, so as part of the M-Step the optimal solutions for the $\mathbf{f}_k$ are calculated based on several different values of λ_k. A Generalized Cross Validation (GCV) procedure is then used to select the optimal $\lambda_k/\mathbf{f}_k$ pair. See Sections 2.4.4 and 2.4.5 for details.

After the initial step, the E-step and the M-step are repeated until differences in the estimates from one iteration to the next are sufficiently small. See Chapter 9 for a discussion of convergence properties.

2.4.2 Step 0: Preliminary Estimates via SVD

Initial values for the factors and factor loadings are required to begin the EM. From those initial values, the variance and autoregressive parameters are calculated to inaugurate the E-step. Choices can be fairly arbitrary, but because of the connection of the functional dynamic factor model with other functional data analysis models (Shen and Huang (2005); Shen and Huang (2008); and Shen (2009)), singular value decomposition (SVD) is utilized on the original $\mathbf{X}$ data matrix for preliminary estimates of the factor time series and factor loadings.

First, $\mathbf{X}$ is decomposed by SVD into three matrices. Two of which are the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ which contain the left and right singular vectors respectively; $\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{I}_m$. The third matrix $\mathbf{D}$ is a diagonal matrix containing the m decreasing singular values on the diagonal. Hence, the SVD of $\mathbf{X}$ is as follows:

$$\begin{aligned}
\mathbf{X} &\overset{SVD}{=} \mathbf{U}_{n\times m}\mathbf{D}_{m\times m}\mathbf{V}'_{m\times m} \\
&= \begin{bmatrix} \mathbf{u}_1 & \ldots & \mathbf{u}_m \end{bmatrix} \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_m \end{bmatrix} \begin{bmatrix} \mathbf{v}'_1 \\ \vdots \\ \mathbf{v}'_m \end{bmatrix} = \sum_{j=1}^{m} d_j \mathbf{u}_j \mathbf{v}'_j.
\end{aligned}$$

Next, based on the model formulation that $\mathbf{X}$ is represented by K factor time series plus noise, the first K SVD components $\{d_k, \mathbf{u}_k, \mathbf{v}_k\}_{k=1}^{K}$ can be used to approximate $\mathbf{X}$ as in $\mathbf{X} \approx \sum_{k=1}^{K} d_k \mathbf{u}_k \mathbf{v}'_k$. Let the (0) subscript denote the step 0 EM values. Then the initial values for the factors and factor loading curves are designated as $\boldsymbol{\beta}_{(0);k} = d_k \mathbf{u}_k$ and $\mathbf{f}_{(0);k} = \mathbf{v}'_k$, for $k = 1, \ldots, K$. From these, initial parameter estimates are computed for σ^2 and the set of factor parameters $\{\sigma_k^2, c_k, \varphi_{1,k}, \ldots, \varphi_{p,k}\}$ as described below in Lemma 2.4.1 and Equations (2.14) and (2.15). Hence with starting values in hand, the true EM iterations can begin.

The next paragraphs briefly detail some of the maximum likelihood solutions to the penalized log-likelihood (2.12), beginning with the error variance, then the factor time series parameters. These are required to calculate the conditional mean and variance used in the E-step. For the moment, the smoothing parameters $\{\lambda_k\}$ are taken as given; Section 2.4.5 details a generalized cross validation approach for their selection.

The solution for the error variance, σ^2, is as follows. The penalized log-likelihood (2.12) is differentiated with respect to σ^2. Setting the resulting expression equal to zero, and solving for σ^2 results in the following MLE for σ^2:

$$\hat{\sigma}^2 = \frac{\sum_{i,j}(x_{ij} - \sum_{k=1}^{K} \beta_{ik} f_{kj})^2}{nm}. \tag{2.14}$$

Recall the assumption that the first p values for each of the factor time series are assumed as given. Then solutions for the AR(p) parameters reduce to an ordinary least squares problem. Holding the $\{\sigma_k^2\}$ fixed for the moment, maximization of (2.12) with respect to the $\{\varphi_k\}$ and intercepts $\{c_k\}$is equivalent to a minimization of a sum of squared errors:

Lemma 2.4.1. *Define* $SSE = \sum_{i=p+1}^{n} \sum_{k=1}^{K} (\beta_{ik} - c_k - \sum_{r=1}^{p} \varphi_{rk}\beta_{i-r,k})^2$*, and let*

$$
\begin{aligned}
\mathbf{y}_k &\equiv [\beta_{p+1,k}, \beta_{p+2,k}, \ldots, \beta_{nk}]', \\
\mathbf{W}_k &\equiv \begin{bmatrix} 1 & \beta_{p,k} & \beta_{p-1,k} & \cdots & \beta_{1,k} \\ 1 & \beta_{p+1,k} & \beta_{p,k} & \cdots & \beta_{2,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta_{n-1,k} & \beta_{n-2,k} & \cdots & \beta_{n-p,k} \end{bmatrix}_{(n-p)\times(p+1)}, \\
\textit{and } \phi_k &\equiv [c_k, \varphi_{1k}, \varphi_{2k}, \ldots, \varphi_{pk}]'.
\end{aligned}
$$

Then $SSE = \sum_{k=1}^{K} \|\mathbf{y}_k - \mathbf{W}_k\phi_k\|^2$ *and for each k the MLEs for the auto-regressive parameters are the OLS solutions* $\hat{\phi}_k = [\mathbf{W}_k'\mathbf{W}_k]^{-1}\mathbf{W}_k'\mathbf{y}_k$.

Alternatively, because of the independence assumption, all of the AR(p) parameters can be solved for simultaneously by posing the problem as one of multivariate regression.

Finally, with all of the estimates $\hat{\phi}$, the individual variances can be found by differentiating the penalized log-likelihood (2.12) with respect to σ_k^2, then setting the result equal to zero and finally solving for σ_k^2:

$$\hat{\sigma}_k^2 = \frac{\sum_{i=p+1}^{n} (\beta_{ik} - c_k - \sum_{r=1}^{p} \varphi_{rk}\beta_{i-r,k})^2}{n-p}. \tag{2.15}$$

With these initial parameters, the E-step can begin to update the values for the factors. Then the M-step is employed to update the values for the factor loading curves and other parameters.

2.4.3 The E-Step

During each E-step, the conditional expectations of the factor time series given the observed data are used to supplement the missing data of the factors themselves. These conditional expectations, in turn, are conditional on the current values for the estimated parameters and the factor loading curves. The first E-step uses the initial factor and factor loading values from step 0, and the parameter estimates generated from them.

Obviously, the E-Step of the EM requires the derivation of the conditional distribution of the factors with regards to the observed data. Restating the model terms of vector notation (as opposed to the matrix formulation), and due to the assumption of normality, the procedure is simplified. With the latter point the distribution is equivalent to the derivation of the first two moments; with the former, said derivation is forthright. To ease in the derivations, however, first consider the following lemma relating the $vec(\cdot)$ operator to the kronecker, or direct product:

Lemma 2.4.2. *(Magnus and Neudecker, 1999). Let* Γ *and* Δ *be two matrices such that the product* $\Gamma\Delta$ *is defined. Then*

$$vec(\mathbf{\Gamma}_{n\times K}\mathbf{\Delta}_{K\times m}) = \left(\mathbf{\Delta}'\otimes\mathbf{I}_n\right) vec(\mathbf{\Gamma}). \tag{2.16}$$

Based on the lemma, the model (4.3) can be rewritten in a vector-ized form, which facilitates the distributional derivations. Let $\boldsymbol{X}\equiv vec(\mathbf{X})$ and $\boldsymbol{\beta}\equiv vec(\mathbf{B})$. Then the model (4.3) $\mathbf{X}=\mathbf{BF}+\epsilon$ can equivalently be written as

$$\boldsymbol{X} = (\mathbf{F}'\otimes\mathbf{I}_n)\boldsymbol{\beta} + vec(\epsilon). \tag{2.17}$$

Derivation of the conditional moments requires the expressions of some of the unconditional moments. Namely, these are:

1. The means of $\boldsymbol{X}$ and $\boldsymbol{\beta}$, denoted as $\mu_{\mathbf{X}}$ and $\mu_{\boldsymbol{\beta}}$, respectively.

2. The variances of $\boldsymbol{X}$ and $\boldsymbol{\beta}$, denoted as $\Sigma_{\mathbf{X}}$ and $\Sigma_{\boldsymbol{\beta}}$, respectively.

3. The covariance of $\boldsymbol{X}$ and $\boldsymbol{\beta}$, denoted as $\Sigma_{\boldsymbol{\beta},\mathbf{X}}$.

These moments in turn are dependent upon the errors associated with the time series factors and the model error ϵ.

First consider the variance matrix for each factor time series $\boldsymbol{\beta}_k$. Recall from Equation (2.9) that for the purposes of this dissertation it is assumed all the K factors follow an AR(p) process of the form $\beta_{ik} = c_k + \sum_{r=1}^{p}\varphi_{r,k}\beta_{i-r,k} + v_{ik}$. Then following Hamilton (1994), the following result can be shown.

Lemma 2.4.3. *Let $\beta_{1,k}, \ldots, \beta_{n,k}$ follow a covariance-stationary AR(p) process represented by $\beta_{ik} = c_k + \sum_{r=1}^{p} \varphi_{r,k}\beta_{i-r,k} + v_{ik}$; with $v_{ik} \overset{iid}{\sim} N(0, \sigma_k^2)$ for $i = 1, \ldots, n$. Define $\gamma_{k,s} = Cov(\beta_{ik}, \beta_{i+s,k})$.*

Then

$$E[\beta_{ik}] = \frac{c_k}{1 - (\sum_{r=1}^{p} \varphi_{r,k})},$$

and

$$\gamma_{k,s} = \begin{cases} \varphi_{k,1}\gamma_{k,s-1} + \varphi_{k,2}\gamma_{k,s-2} + \ldots \varphi_{k,p}\gamma_{k,s-p} & \text{for } s = 1, 2, \ldots \\ \varphi_{k,1}\gamma_{k,1} + \varphi_{k,1}\gamma_{k,2} + \ldots \varphi_{k,1}\gamma_{k,p} + \sigma_k^2 & \text{for } s = 0 \end{cases}.$$

Define the $n \times n$ variance matrix for $\boldsymbol{\beta}_k$ as Σ_k. Its elements are then $[\Sigma_k]_{h,i} = \gamma_{k,|h-i|}$. These results give rise to the unconditional moments for $\boldsymbol{X}$ and $\boldsymbol{\beta}$ which are collected in the following proposition.

Proposition 2.4.1. *Recall from the Equations* (2.4) *that $\epsilon_{ij} \sim N(0, \sigma^2)$. Then*

$$Var[vec(\epsilon)] = \sigma^2 \mathbf{I}_{nm}.$$

Further, let $\mathbf{c}$ be the $K \times 1$ vector with elements $c_k/[1 - (\sum_{r=1}^{p} \varphi_{r,k})]$. Then

$$\begin{aligned} \mu_{\boldsymbol{\beta}} &= \mathbf{c} \otimes \mathbf{1}_n \\ \mu_{\mathbf{X}} &= (\mathbf{F}' \otimes \mathbf{I}_n)\mu_{\boldsymbol{\beta}} \\ \Sigma_\beta &= diag\{\Sigma_1, \ldots, \Sigma_K\} \\ \Sigma_{\mathbf{X}} &= (\mathbf{F}' \otimes \mathbf{I}_n)\Sigma_\beta(\mathbf{F} \otimes \mathbf{I}_n) + \sigma^2\mathbf{I}_{nm} \\ \Sigma_{\beta,\mathbf{X}} &= \Sigma_\beta(\mathbf{F} \otimes \mathbf{I}_n). \end{aligned}$$

Next, using properties of multivariate normal random vectors, the conditional distribution of $\beta|\boldsymbol{X}$ can be found:

Proposition 2.4.2. *Let*

$$\begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{X} \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_\beta \\ \mu_{\mathbf{X}} \end{pmatrix}, \begin{pmatrix} \Sigma_\beta & \Sigma_{\beta,\mathbf{X}} \\ \Sigma_{\mathbf{X},\beta} & \Sigma_{\mathbf{X}} \end{pmatrix}\right].$$

Then

$$\mu_{\beta|\mathbf{X}} \equiv E[\boldsymbol{\beta}|\mathbf{X}] = \mu_\beta + \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}(\boldsymbol{X} - \mu_{\mathbf{X}}),$$

and

$$\Sigma_{\beta|\mathbf{X}} \equiv Var[\boldsymbol{\beta}|\mathbf{X}] = \Sigma_\beta - \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\Sigma_{\mathbf{X},\beta}.$$

This then implies that

$$E[\boldsymbol{\beta}\boldsymbol{\beta}'|\mathbf{X}] = \Sigma_{\beta|\mathbf{X}} + \mu_{\beta|\mathbf{X}}\mu'_{\beta|\mathbf{X}}.$$

Note that from a computational standpoint there is concern over the appearance of $\Sigma_{\mathbf{X}}^{-1}$ in the expressions for both $\mu_{\boldsymbol{\beta}|\mathbf{X}}$ and $\Sigma_{\boldsymbol{\beta}|\mathbf{X}}$ since this is an inversion of order nm; because the EM is an iterative procedure, this could be especially problematic (recall the call center data with $n = 210$ and $m = 68$). Thankfully, there is a method by which the inversion can be reduced to K sequential $n \times n$ inversions. This and more computational efficiencies are discussed in Section 2.4.6.

With these conditional moments, the E-step of the EM posits that the missing data (the time series factors) are replaced with the known values of the conditional distribution given $\boldsymbol{X}$. Thus in the M-step, in solving for MLEs, expressions involving $\boldsymbol{\beta}_k$ will utilize values from $\mu_{\beta|\mathbf{X}}$, $\Sigma_{\beta|\mathbf{X}}$, and $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}]$.

2.4.4 The M-Step

The M-step uses the conditional moments from the E-step as "stand-in" values so that the penalized likelihood expression (2.12) can be optimized. Specifically, in the M-step, MLEs can be determined using the surrogate conditional moments $\mu_{\beta|\mathbf{X}}$, $\Sigma_{\beta|\mathbf{X}}$, and $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}]$ in place of the missing factor time series, $\boldsymbol{\beta}_k$. In fact, the crux of the E-step/M-step transition is replacing the unknown factor terms in MLE solutions with the corresponding known terms from the conditional expectations.

To do this requires some overhead in terms of notation and some minimal derivation; bear with it, it's worth it. Recall the set Θ from Equation (2.13); and denote the lth EM iteration parameter estimates as $\Theta(l)$. For each iteration l, the M-step optimizes the *conditional* penalized log-likelihood (2.12) given the observed data and the lth parameter estimates:

$$\begin{aligned} E[PL|\mathbf{X}]_{|\Theta(l)} &\propto \frac{1}{\sigma^2}\sum_{i,j} E\left[(x_{ij} - \sum_{k=1}^{K}\beta_{ik}f_{kj})^2|\mathbf{X}\right]_{|\Theta(l)} \\ &+ \sum_{i=p+1}^{n}\sum_{k=1}^{K}\frac{1}{\sigma_k^2}E\left[(\beta_{ik} - c_k - \sum_{r=1}^{p}\varphi_{rk}\beta_{i-r,k})^2|\mathbf{X}\right]_{|\Theta(l)} + \sum_{k=1}^{K}\lambda_k\mathbf{f}_k\Omega\mathbf{f}_k'. \end{aligned} \tag{2.18}$$

As a matter of notation, where necessary, parameters or random variables will be suffixed with a value indicating the relative iteration of the EM; 0 represents the initial values and $l = 0, \ldots, L$ represents the current iteration of the EM.

Based on the above expression, it is clear that in the MLEs, the factor time series appear either singly or in terms of cross products. Further, the cross products occur either within or between factors. These three variants, and the corresponding replacements are made thusly:

Individual factors: The conditional mean of the vector-ized factors, $\mu_{\beta|\mathbf{X}}$, consists of the conditional means of each factor: $[E(\boldsymbol{\beta}_1|\boldsymbol{X})'_{n\times 1}, \ldots, E(\boldsymbol{\beta}_K|\boldsymbol{X})']'$. Thus for each time point i and each factor k, $\beta_{ik;(l)}$ is replaced with the ith element of factor k's conditional expectation, $[E(\boldsymbol{\beta}_k|\boldsymbol{X})]_i$.

Within factor cross products: For a given factor k, and time points $i, h = 1, \ldots, n$, then $[\beta_{ik}\beta_{hk}]_{(l)} = E[\beta_{ik}\beta_{hk}|\boldsymbol{X}]$. An exciting result is that $\Sigma_{\beta|\mathbf{X}}$ is block diagonal with K $n \times n$ blocks. Therefore, $E[\beta_{ik}\beta_{hk}|\boldsymbol{X}]$ is simply the i, jth element of the kth diagonal block from the matrix $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}]$. Section 2.4.6 highlights the computational efficiencies of this result; for a derivation of this see Chapter 3.

Between factor cross products: For a given factors k and k'; and some given time points $i, h = 1, \ldots, n$; the replacement is $[\beta_{ik'}\beta_{hk}]_{(l)} = E[\beta_{ik'}\beta_{hk}|\boldsymbol{X}]$. An even more amazing result is that $E[\beta_{ik'}\beta_{hk}|\boldsymbol{X}]$ is simply the i, jth element of the matrix formed by $[E(\boldsymbol{\beta}_k|\boldsymbol{X})] \cdot [E(\boldsymbol{\beta}'_k|\boldsymbol{X})]'$. Put another way, *the conditional expectation of the product* is just the *product of the conditional expectations.* For a derivation of this see Chapter 3.

The M-step, then, is just a matter of making these substitutions into (2.18), and solving for the MLEs. For ease of notation, going forward it will be implicit in the expressions and derivations that items like β_{ik} or $\beta_{ik}\beta_{hk'}$ are equivalent to $E[\beta_{ik}|\boldsymbol{X}]$ and $E[\beta_{ik}\beta_{hk'}|\boldsymbol{X}]$, respectively; for $i, h = 1, \ldots n$ and $k, k' = 1, \ldots, K$. These will occasionally be suffixed with an (l) to emphasize the iteration of the EM.

The first part of the M-step is solving for the factor loading curves $\mathbf{f}_k$. It is also the most complicated, in that it involves a GCV selection procedure for the smoothing parameters λ_k.

Recall the MLE solution for σ^2 from Equation (2.14). Making the appropriate substitutions discussed in the preceding paragraphs, the solutions for the $\mathbf{f}_k$ are found in the following manner.

Components of the penalized likelihood expression involving the factor loadings $\mathbf{f}_k$ can be rewritten as expression of the vector $\tilde{F}' \equiv vec(\mathbf{F}')$; this is each factor loading curve $\mathbf{f}'_k$ stacked on top of each other. Assume for the moment the $\{\lambda_k\}$ are known. Then differentiating with respect to the vector $\tilde{F}'$ and setting the result equal to zero yields the simultaneous solutions for all the factor loading curves. Recall from Section 2.3 the block diagonal matrix $\mathbf{S}$ with K $m \times m$ blocks $\lambda_k\Omega_k$. Then the following proposition illustrates the solution.

Proposition 2.4.3. *Let* $\tilde{X} \equiv vec(\mathbf{X}')$*,* $\tilde{F} \equiv vec(\mathbf{F}')$*, and* $\mathbf{Z} \equiv \mathbf{B} \otimes \mathbf{I}_m$*. Then*

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - \sum_{k=1}^{K}\beta_{ik}f_{kj})^2 + \sum_{k=1}^{K}\lambda_k\mathbf{f}_k\Omega_k\mathbf{f}'_k,$$

is equivalent to

$$\frac{1}{\sigma^2}\tilde{X}'\tilde{X} - \frac{2}{\sigma^2}\tilde{F}'\mathbf{Z}'\tilde{X} + \tilde{F}'\left(\frac{\mathbf{Z}'\mathbf{Z}}{\sigma^2} + \mathbf{S}\right)\tilde{F}, \tag{2.19}$$

which suggests that

$$\hat{\hat{F}} = \left(\frac{\mathbf{Z}'\mathbf{Z}}{\sigma^2} + \mathbf{S}\right)^{-1}_{mK\times mK} \frac{1}{\sigma^2}\mathbf{Z}'\tilde{X}. \tag{2.20}$$

Let $\| \cdot \|$ and $\langle \cdot, \cdot \rangle$ denote the Euclidean norm and Euclidean inner-product, respectively. Then for each factor loading curve $\mathbf{f}_k$, it can be shown that

$$\hat{\mathbf{f}}'_k \quad = \quad \left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k \Omega_k\right]^{-1}_{m\times m} \frac{1}{\sigma^2} \left[\sum_{i=1}^{n} \beta_{ki} X_i - \sum_{k\neq h} \langle \boldsymbol{\beta}_h, \boldsymbol{\beta}_k \rangle \hat{\mathbf{f}}'_k\right]_{m\times 1},$$

for $k, h = 1, \ldots, K$.

The expression for any one $\mathbf{f}_k$ depends on all of the others, thus unfortunately a simultaneous solution is difficult to derive. Therefore it is proposed to solve for the $\hat{\mathbf{f}}_k$ sequentially. Let $h = 1, \ldots, K$. Then to solve for a particular $\hat{\mathbf{f}}_k$, the set $\{\hat{\mathbf{f}}_h\}$ consisting of the other $K-1$ factor loading curves are assumed given; the values used for them are provided by the previous iteration of the EM. In practice, for iteration l of the EM, the kth factor loading curve is given by:

$$\hat{\mathbf{f}}'_{(l);k} \quad = \quad \left[\frac{\|\boldsymbol{\beta}_{(l);k}\|^2}{\sigma^2_{(l-1)}}\mathbf{I}_m + \lambda_k \Omega_k\right]^{-1} \times \frac{1}{\sigma^2_{(l-1)}} \left[\sum_{i=1}^{n} \beta_{(l);ik} X_i - \sum_{h\neq k} \langle \boldsymbol{\beta}_{(l);h}, \boldsymbol{\beta}_{(l);k} \rangle \hat{\mathbf{f}}'_{(*);h}\right].$$

The values for $\hat{\mathbf{f}}'_{(*);h}$ in turn are provided by the following rule:

$$\hat{\mathbf{f}}_{(*);h} = \begin{cases} \hat{\mathbf{f}}_{(l);h} & \text{if } h < k \\ \hat{\mathbf{f}}_{(l-1);h} & \text{if } h > k \end{cases}.$$

So for example, suppose it is the seventh EM iteration and the third factor loading curve is being solved for. Then curves one and two will already have been updated, so that the most recent estimates may be used. For curves 4 to K, the only estimates available will be those from the previous sixth iteration of the EM.

The missing component in all of this is the treatment of the set of smoothing parameters $\{\lambda_k\}$. They are neither random components of the functional dynamic factor model; nor are they "parameters" in the maximum likelihood estimator sense. Nay, they are a unique model component in and of themselves. Their origin has been discussed in Section 2.3; the values chosen for them are discussed in the following one.

2.4.5 GCV Selection

When sequentially solving for the K factor loading curves, the optimal smoothing parameter λ_k is selected using generalized cross validation, akin to the methods of Green and Silverman (1994). This requires calculating the solution for $\mathbf{f}_k$ over multiple candidate values for λ_k; then selecting the one minimizing the GCV criterion. The justification is that the solution for $\mathbf{f}_k$ can be posed as a ridge regression problem; provided that for $h = 1, \ldots, K$, the other $K - 1$ $\hat{\mathbf{f}}_h$ are fixed. To see this, recall the expression from Proposition 2.4.3:

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - \sum_{k=1}^{K}\beta_{ik}f_{kj})^2 + \sum_{k=1}^{K}\lambda_k \mathbf{f}_k \Omega \mathbf{f}_k'.$$

The solution for a single $\mathbf{f}_k$ requires fixing the remaining $K - 1$ factor loading curves and their corresponding smoothing parameters. This effectively renders them as constant; therefore the above expression can be rewritten as:

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\left(x_{ij} - \sum_{h\neq k}\beta_{ih}f_{hj}\right) - \beta_{ik}f_{kj}\right]^2 + \sum_{h\neq k}\lambda_h \mathbf{f}_h \Omega \mathbf{f}_h' + \lambda_k \mathbf{f}_k \Omega \mathbf{f}_k'.$$

Defining $X_i^{-k} \equiv X_i - \sum_{h\neq k}\beta_{ih}\hat{\mathbf{f}}_h'$ and $\tilde{X}^{-k}$ as the stacked columns of X_i^{-k} for $i = 1, \ldots, n$, the criterion that needs to be minimized to obtain $\hat{\mathbf{f}}_k$ can be rewritten as

$$\left\|\frac{1}{\sigma}\tilde{X}^{-k} - \frac{1}{\sigma}(\boldsymbol{\beta}_k \otimes \mathbf{I}_m)\cdot \mathbf{f}_k'\right\|^2 + \lambda_k \mathbf{f}_k \Omega \mathbf{f}_k'.$$

Based upon this formulation it is evident that this expression exactly matches a ridge regression problem with:

- The $nm \times 1$ vector $\frac{1}{\sigma}\tilde{X}^{-k}$ as the "dependent variable."

- The $nm \times m$ matrix $\frac{1}{\sigma}(\boldsymbol{\beta}_k \otimes \mathbf{I}_m)$ as the "independent variables."

- The $m \times 1$ vector $\mathbf{f}'_k$ as the "parameter" vector for which to be solved.

- Finally, the ridge penalty term as $\lambda_k \mathbf{f}_k \Omega \mathbf{f}'_k$.

Keeping in mind that $(\boldsymbol{\beta}'_k \otimes \mathbf{I}_m)(\boldsymbol{\beta}_k \otimes \mathbf{I}_m) = \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m$, then the GCV criterion can be used to select the optimal λ_k (Green and Silverman, 1994):

$$\mathrm{GCV}(\lambda_k) = \frac{\|(\mathbf{I}_{nm} - H_{\lambda_k})\tilde{X}^{-k}\|^2/nm}{[1 - \mathrm{tr}(H_{\lambda_k})/nm]^2}, \tag{2.21}$$

with

$$H_{\lambda_k} = \left\{ \frac{1}{\sigma^2}(\boldsymbol{\beta}_k \otimes \mathbf{I}_m) \left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k \Omega_k \right]^{-1}_{m\times m} (\boldsymbol{\beta}'_k \otimes \mathbf{I}_m) \right\}_{nm\times nm}. \tag{2.22}$$

$\mathrm{GCV}(\lambda_k)$ is calculated over a grid of possible values during the estimation of each factor loading curve. The smoothing parameter that corresponds to the least value of $\mathrm{GCV}(\cdot)$ is selected as the optimal one. Then the M-step proceeds to estimation of the next factor loading curve, along with the selection of the penalty parameter for that loading curve.

Before moving on to the final steps of EM estimation in the FDFM, it is worthwhile to note that the M-step with GCV is computationally intensive. For example, for a K factor model using, say, W possible values for each of the smoothing parameters requires $K \cdot W$ steps to solve for $\mathbf{F}$. H_{λ_k} is a large matrix and its calculation includes the inversion of a smaller (though not unformidable) matrix. However, it will be shown in Section 2.4.6 that only one calculation need be performed in place of $K \cdot W$ calculations.

Regardless, after all of the $\mathbf{f}_k/\lambda_k$ pairs are determined, normalization/orthogonal-ization is required in order to maintain the assumption of orthogonality; that $\mathbf{F}_{(l)}\mathbf{F}'_{(l)} = \mathbf{I}_K$. Following this adjustment, the factor time series must be appropriately adjusted themselves. This is method is detailed in Section 3.2.4.

With the updated orthogonal factor loadings and the updated factors, the M-step concludes with the calculation of the overall model variance (σ^2), and the auto-regressive factor parameters. The ML solutions are, of course, no different than those presented in *Step 0* (Section 2.4.2). However there are a few points worth noting now that the conditional expectations for the $\{\boldsymbol{\beta}_k\}$

are being used (in place of the left singular vectors from SVD of the data matrix). Specifically,

1. Using the vector-ized model notation (4.12), the error variance estimate is expressed as $\hat{\sigma}^2 = \|\boldsymbol{X} - (\hat{\mathbf{F}}' \otimes \mathbf{I}_n)\boldsymbol{\beta}\|^2/nm$. Because the error variance involves cross-products of the factors, care must be taken in implementation of the EM. Thus, expanding the expression illustrates the proper distinctions to made:

$$nm \cdot \hat{\sigma}^2_{(l)} \quad = \quad \|\boldsymbol{X}\|^2 - 2\langle \boldsymbol{X}, (\mathbf{F}'_{(l)} \otimes \mathbf{I}_n)\boldsymbol{\beta}_{(l)}\rangle + \sum_{k=1}^{K} \|\boldsymbol{\beta}_{k;(l)}\|^2.$$

   Here, $\|\boldsymbol{\beta}_k\|^2$ represents the sum of the diagonal elements of $E[\boldsymbol{\beta}_k\boldsymbol{\beta}'_k|\mathbf{X}]$. Whereas $\boldsymbol{\beta}_{(l)}$ is simply $\mu_{\boldsymbol{\beta}|\mathbf{X};(l)}$.

2. For illustrative purposes, consider the case of independent AR(1) factors with a constant. Then c_k and φ_k minimize the sum of squares $\sum_{i=2}^{n}(\beta_{ik} - c_k - \varphi_k\beta_{i-1,k})^2$ yielding the standard OLS result:

$$\begin{bmatrix} \hat{c}_k \\ \hat{\varphi}_k \end{bmatrix} = \begin{bmatrix} n-1 & \sum_{i=1}^{n-1} E[\beta_{ik}|\mathbf{X}] \\ \sum_{i=1}^{n-1} E[\beta_{ik}|\mathbf{X}] & \sum_{i=1}^{n-1} E[\beta_{ik}^2|\mathbf{X}] \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=2}^{n} E[\beta_{ik}|\mathbf{X}] \\ \sum_{i=2}^{n} E[\beta_{i-1,k}\beta_{ik}|\mathbf{X}] \end{bmatrix}.$$

   The $E[\cdot|\mathbf{X}]$ notation is temporarily reintroduced in order to emphasize the distinction between terms like $E[\beta_{i-1,k}|\mathbf{X}]E[\beta_{ik}|\mathbf{X}]$ and $E[\beta_{i-1,k}\beta_{ik}|\mathbf{X}]$, which are obtained from the $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\mathbf{X}]$ matrix. The point is that the correct EM estimates for c_k and φ_k are not obtained via merely a regression of $E[\boldsymbol{\beta}_k|\mathbf{X}]$ on itself; the distinction presented here will not be accounted for by a built-in regression function in software packages. For a more general discussion, Chapter 3 includes a thorough derivation of the case of factors with multiple intercepts.

3. As with the above cases, similar care must be taken in regards to the factor time series' error variances. Reverting again to the AR(1) model it can be shown that

$$(n-1) \cdot \hat{\sigma}_k^2 = \sum_{i=2}^{n}(\beta_{ik}^2 + c_k^2 + \varphi_k^2\beta_{i-1,k}^2 - 2c_k\beta_{ik} - 2\varphi_k\beta_{ik}\beta_{i-1,k} + 2c_k\varphi_k\beta_{i-1,k}).$$

> The appropriate replacements for the squared and cross product β terms are obtained from the components on the diagonals and off-diagonals of the $E[\boldsymbol{\beta}_k \boldsymbol{\beta}_k' | \mathbf{X}]$ matrix. Some useful results for sums of these elements are discussed in the next section.

Finally, after these parameter estimates are updated, the EM continues on to the next E-step where the factor time series are updated and fed into another M-step. The process continues until the factor loading curves and other parameters from one EM iteration to the next are sufficiently close. For example, when $\max |\Theta_{(l)} - \Theta_{(l-1)}| < \delta$, for some small number δ.

The EM is an elegant procedure, though not without some limitations. It is iterative, and both the E-step and M-step do require some rather large matrix manipulations. Fortunately, presented in the next section are several convenient results that assuage some of the computational intensity encountered in this iterative process.

2.4.6 Computational Efficiency

This section presents results intended to ease some of the computational aspects of the estimation for the functional dynamic factor model. This is done by exploiting the properties of some of the matrices involved in key calculations. The impetus for this is not just to provide the elegant solution. Rather, from a very practical standpoint, even with modern statistical software and high performance computing platforms, run-time can be an issue. The reason being that the EM algorithm is an iterative procedure. Both the E-step and M-step contain some large matrix inversions and manipulations. The M-step is further complicated by the sequential solution of the factor loading curves, and the solution for each is computed over a range of values for the smoothing parameter. All of this amounts to what could be a computationally intensive estimation.

The crux of computational efficiency is in the details for the conditional moments of the time series factors; these are revisited below. It is first shown that a considerably faster inversion exists for the variance matrix of the observed data $\mathbf{X}$. Next, a result regarding the structure of the conditional variance of the factors, $\boldsymbol{\beta}$, facilitates a useful finding in determining factor cross products; these are terms that are all but omnipresent in the M-step. Finally, an eigendecomposition of a particular matrix in the GCV procedure obviates a matrix inversion for each candidate value of the smoothing parameter.

Matrix Inversion: Recall the results of Proposition 2.4.2 regarding the conditional mean and variance of the factor time series:

$$\begin{aligned}
\mu_{\beta|\mathbf{X}} &= \mu_\beta + \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}(\mathbf{X} - \mu_{\mathbf{X}}), \\
\Sigma_{\beta|\mathbf{X}} &= \Sigma_\beta - \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\Sigma_{\mathbf{X},\beta}.
\end{aligned}$$

and

$$E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}] = \Sigma_{\beta|\mathbf{X}} + \mu_{\beta|\mathbf{X}}\mu'_{\beta|\mathbf{X}}.$$

The inverse of the variance matrix for $\mathbf{X}$, $(\Sigma_{\mathbf{X}}^{-1})$, appears in each of these. The inversion is of order nm. It must be performed for each of the L iterations of the EM. However, the order inversion can be reduced from order nm to a sequence of K (the number of factors) order n inversions. The following lemma facilitates this result:

Lemma 2.4.4 (Sherman–Morrison–Woodbury Formula (Press et al., 1992).)**.** *Let A be a $T \times T$ matrix, B be a $d \times d$ matrix and C be a $d \times T$ matrix. If A^{-1} and B^{-1} exist, then*

$$(A^{-1} + C'B^{-1}C)^{-1} = A - AC'(CAC' + B)^{-1}CA.$$

Using this formula, the result of a simplified inversion can be shown. Let $A = \sigma^{-2}\mathbf{I}_{nm}$, $B = \Sigma_\beta^{-1}$, and $C = (\mathbf{F} \otimes \mathbf{I}_n)$. Then it can be shown that

$$\Sigma_{\mathbf{X}}^{-1} = \sigma^{-2}\mathbf{I}_{nm} - \sigma^{-4}(\mathbf{F}' \otimes \mathbf{I}_n)\left[\sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]^{-1}(\mathbf{F} \otimes \mathbf{I}_n). \tag{2.23}$$

The form itself is not so important as what it means. Instead of inverting $\Sigma_{\mathbf{X}}$ directly, which is an $nm \times nm$ matrix, only the middle matrix $\left[\sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]$ need be inverted. This matrix is of smaller size $nK \times nK$.

It gets better. Recall from Proposition 2.4.1 that Σ_β is block-diagonal, with K, $n \times n$

blocks. That Σ_β is a block-diagonal matrix implies that Σ_β^{-1} is also block-diagonal. Obviously then, $\sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}$ is likewise block-diagonal. So in fact, the inversion of $\left[\sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]$ amounts to a sequence of K, $n \times n$ inversions. That is, *using this factorization, the inversion of* $\Sigma_{\mathbf{X}}$ *is reduced from an* $nm \times nm$ *inversion to* K*,* $n \times n$ *inversions.* For example, consider a five factor model using the call center data where $n = 210$ and $m = 68$. In place of inverting an order $nm = 14,280$ matrix for as many iterations as the EM requires, five order $n = 210$ matrices are inverted.

Block Diagonality: The ML solutions used in the M-step are rife with expressions involving the factor time series $\{\boldsymbol{\beta}_k\}$ in various forms. When some $\boldsymbol{\beta}_k$ appears singly, those values are imputed with values from $\mu_{\beta|\mathbf{X}}$. However, when products of factors appear, such as $\langle \boldsymbol{\beta}_k, \boldsymbol{\beta}_h \rangle$, then the imputation comes from the $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}]$ matrix. This is a seemingly ominous matrix from which to draw values:

$$E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}] = \Sigma_\beta - \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\Sigma_{\mathbf{X},\beta} + \mu_{\beta|\mathbf{X}}\mu'_{\beta|\mathbf{X}}.$$

Fortunately, it is actually not ominous at all. It turns out that $\Sigma_{\boldsymbol{\beta}|\mathbf{X}}$ is block diagonal, and that this property facilitates a rather convenient result regarding between-factor cross products:

Proposition 2.4.4. *$\Sigma_{\boldsymbol{\beta}|\boldsymbol{X}}$ is block diagonal with K $n \times n$ blocks.*

Corollary 2.4.1. *For $h \neq k$, $E[\langle \boldsymbol{\beta}_k, \boldsymbol{\beta}_h \rangle|\mathbf{X}] = \langle \mu_{\boldsymbol{\beta}_k|\mathbf{X}}, \mu_{\boldsymbol{\beta}_h|\mathbf{X}} \rangle$.*

For a proof the reader is referred to Chapter 3. Essentially what this means is that the conditional expectation of a product of two (distinct) factors is simply the product of their individual expectations. This greatly simplifies the M-step for the factor loading curves in particular; the GCV step can be time consuming so computational efficiency here is quite beneficial.

Eigen-decomposition: The final result of this section is in regard to the GCV selection of the smoothing parameters $\{\lambda_k\}$. Solving for the loading curves sequentially and for multiple values of the smoothing parameter can increase computing time. Recall from

Equation (2.21) that GCV(λ_k) is dependent on the matrix H_{λ_k}. This matrix in turn depends on the matrix $\left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k\Omega_k\right]_{m\times m}$ (see Equation (2.22)) which, ordinarily, must be inverted for each of, say, W candidate values for λ_k. This occurs for each of how many ever iterations it takes for the EM to converge. However, two helpful results greatly increase the efficiency of the GCV selection within the M-Step. Using the eigen-decomposition of Ω_k, a method exists for which the only inversion required is the inversion of a diagonal matrix. Consider the following proposition:

Proposition 2.4.5. *Let* $\mathbf{S}(\lambda_k) = \left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k\Omega_k\right]^{-1}$ *so that* $\hat{\mathbf{f}}'_k = [\mathbf{S}(\lambda_k)]\left[\frac{1}{\sigma^2}(\boldsymbol{\beta}'_k \otimes \mathbf{I}_m)\tilde{X}^{-k}\right]$. *Given the eigen-decomposition of the* $m \times m$ *penalty matrix* $\Omega_k = \Gamma\Delta\Gamma'$ *with* $\Delta_{m\times m} = diag\{\delta_j\}_{j=1}^m$, *then*

$$\mathbf{S}(\lambda_k) \quad = \quad \Gamma \cdot diag\left\{\left(\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2} + \lambda_k\delta_j\right)^{-1}\right\}\Gamma',$$

and

$$tr\{H_{\lambda_k}\} \quad = \quad \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\sum_{j=1}^{m}\frac{1}{\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2} + \lambda_k\delta_j}.$$

With this result, looping through W possible values for the smoothing parameter is accomplished through a single eigen-decomposition, followed by a diagonal matrix inversion for each of the factors; clearly this is more desirable than performing an $m \times m$ inversion for each of the K factors and W candidate values for λ_k.

With these computational aspects covered, this concludes the exciting discussion of EM estimation for the functional dynamic factor model. This Section 2.4 opened with a brief description of the EM as applied to the functional dynamic factor model presented here. Detailed descriptions of the initialization process (Step-0) followed; the results of which inaugurate the E-step. The E-step specified the imputation for the missing data, otherwise known as the factor time series. With proxies for the factors, the M-step updates values for all of the parameters and factor loading curves. Then these updated values in turn are used for the next iteration of the E-step, and so on. Finally, some results regarding efficient computation have been presented to aid in the estimation. The chapter concludes with some alternative specifications of the aforementioned model to provide greater generality for the sake of some applications used in later chapters.

2.5 Alternative Models

Presented in this section are two alternative specifications of the functional dynamic factor model. The first specification is a method by which to estimate the K factor model as if it were a sequence of single factor models. The second alternative model is merely a more general specification of the original model where the time series factors contain observed explanatory variables.

2.5.1 Sequential K Factor Model

Though shown to be an elegant procedure, EM estimation of the FDFM does have some drawbacks in terms of computational intensity. As has been seen in the M-step earlier in Sections 2.4.4 and 2.4.5, solutions for the factor loading curves required a sequential approach and further included an additional loop in order to determine the optimal smoothing parameter for each k. In the Section 2.4.6 it was shown that some of the larger matrix inversions required in both the E-step and M-step can be simplified. However, even with these results, the iterative EM still requires many large matrix manipulations over many iterations.

A simpler approach is the use of a single factor model, which typically reduces the dimension of some of the larger matrices by a factor of K. For example, consider the variance matrix of the factor time series $\Sigma_{\boldsymbol{\beta}}$. In the K factor case this is of size $nK \times nK$; in the single factor case, it is only of size $n \times n$. The cost of the simpler specification is loss in goodness-of-fit, however. An intermediate approach exists which estimates multiple factors while still using the single factor method. The idea is to estimate the K factor model as a sequence of K single factor models. To illustrate this, it is assumed the factor follow an AR(1) process.

The one factor model is expressed as

$$\mathbf{X}_{n\times m} = \boldsymbol{\beta}_{n\times 1}\mathbf{f}_{1\times m} + \epsilon.$$

Equivalently, in scalar notation:

$$\begin{cases} x_{ij} = \beta_i f_j + \epsilon_{ij}, \\ \beta_i = c + \varphi \beta_{i-1} + v_i, \end{cases}$$

where $v_i \overset{i.i.d.}{\sim} N(0, \sigma_1^2)$, $\epsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma^2)$, and $E v_t \epsilon_{sj} = 0$ for all $t, s = 1, \ldots, n; j = 1, \ldots, m$.

As in the multi-factor case, the first step is to decompose the data matrix using SVD; for a K factor specification, K left and right singular vectors are retained. Initial estimates for the factor and factor loadings result from the singular values and vectors, and initial parameter estimates are calculated based on the likelihood expressions from the one factor case outlined in Chapter 3.

The EM algorithm is then implemented K times for each of the K factors, *but is based on a modified data matrix* $\mathbf{X}^{-k}$. For h, k in $1, \ldots, K$:

$$\mathbf{X}^{-k} = \mathbf{X} - \sum_{h \neq k} \boldsymbol{\beta}_{h;(*)} \hat{\mathbf{f}}_{h;(*)},$$

where

$$\hat{\mathbf{f}}_{(*);h}, \boldsymbol{\beta}_{h;(*)} = \begin{cases} \hat{\mathbf{f}}_{(0);h}, \boldsymbol{\beta}_{h;(0)} & \text{if } h > k \\ \hat{\mathbf{f}}_{(L);h}, \boldsymbol{\beta}_{h;(L)} & \text{if } h < k \end{cases}. \tag{2.24}$$

That is, the modified data matrix for the factor of interest k is the difference between the actual data matrix and the sum of the estimated loading and factor products for the other factors. In effect, $\mathbf{X}^{-k}$ is a residual matrix of the actual data less components predicted by either the EM or the Step 0 SVD estimates. So for example, for a three factor model, EM estimation for factor 2 parameters are based on the data matrix X less the EM estimates for $\{\boldsymbol{\beta}_1, \mathbf{f}_1\}$ for factor one, and less the Step 0 SVD estimates for $\{\boldsymbol{\beta}_3, \mathbf{f}_3\}$ for factor three.

Again, the reason for this sequential factor approach is in the interest of computational expediency, so that estimation may be hastened.

2.5.2 Factor Time Series Parameters With Regressors

Situations may arise where additional explanatory power may be desired by way of explanatory variables in the time series factors of the functional dynamic factor model; these are represented by the vectors A_{ik} in the model Equations (2.4). As the factors are latent, situations where known explanatory variables are hypothesized to influence unknown, unobserved variables are likely rare. However, such cases do exist. In the context of the call center data, for example, a day of the week variable can serve as a known explanatory variable for the latent factors.

Solving for the maximum likelihood estimates when the time series have explanatory variables is slightly more complicated than the case of a single intercept presented in the EM Section 2.4 above. Again calling upon the call center data example, a particularly neat result exists when the data consist of full weeks; this result is presented in Section 3.3.2. The more general result for any type of non-stochastic regressor is presented here.

Consider regressors $A_{ik} = [a_{ik}^{(1)}, \ldots, a_{ik}^{(d)}]$ with corresponding coefficients $\mu_k = [\mu_k^{(1)}, \ldots, \mu_k^{(d)}]$. Without loss of generality, it can be assumed d is the same for all $k = 1, \ldots, K$. Recall the original model specification (2.4):

$$\beta_{ik} - A_{ik}\mu_k = \textstyle\sum_{r=1}^{p} \varphi_{rk}(\beta_{i-r,k} - A_{i-r,k}\mu_k) + v_{ik}, \quad k = 1, \ldots, K$$

with $v_{ik} \overset{i.i.d.}{\sim} N(0, \sigma_k^2)$; again maintaining the assumption that the K time series are independent.

The portion of the penalized log-likelihood expression (2.12) involving the factor time series is then a sum of squared errors similar to that presented in Lemma 2.4.1:

$$SSE \;=\; \sum_{i=p+1}^{n} \sum_{k=1}^{K} \left[(\beta_{ik} - A_{ik}\mu_k) - \sum_{r=1}^{p} \varphi_{rk}(\beta_{i-r,k} - A_{i-r,k}\mu_k) \right]^2 \qquad (2.25)$$

At first glance of the above expression, it appears all that is required is the usual OLS solution. But note that expanding the expression results in cross product terms involving both μ_k and the $\{\varphi_{rk}\}_{r=1}^{P}$. If the $\{\varphi_{rk}\}$ were known, then indeed OLS could be used to solve for μ_k. Conversely, were μ_k known, OLS could be used to solve for the $\{\varphi_{rk}\}$. To resolve this dependency, two different iterative methods are proposed. The μ_k parameters are estimated first, given initial values for the $\{\varphi_{rk}\}$. Then the μ_k estimates are used to solve for the $\{\varphi_{rk}\}$. The process continues until estimates from one iteration are sufficiently close to those from the next.

The Iterated Cochrane-Orcutt Method: One easily implemented method, especially in the case of AR(1) processes, is the Iterated Cochrane-Orcutt Method (Hamilton, 1994) which can iteratively solve for the optimal values of μ_k and the φ_{rk}.

***Step 1*:** An initial value is selected for the φ_{rk}. Unless a more intuitive choice exists, typically the initial values are set to 0. Solve for $\hat{\mu}_k$ as the OLS solution to $\beta_{ik} - \sum_{r=1}^{p} \varphi_{rk}\beta_{i-r,k} = (A_{ik} - \sum_{r=1}^{p} \varphi_{rk}A_{i-r,k})\mu_k + \eta_{ik}$.

***Step 2*:** Using the μ_k estimated in the previous step, compute the OLS solution for the φ_{rk} in $(\beta_{ik} - A_{ik}\mu_k) = \sum_{r=1}^{p} \varphi_{rk}(\beta_{i-r,k} - A_{i-r,k}\mu_k) + \eta_{ik}$.

Then the φ_{rk} estimates from step 2 are used to find a new estimate for μ_k in step 1. The steps are then repeated until estimates for μ_k and the φ_{rk} converge to a local maximum.

The Yule Walker/GLS Method: A more general iterative method is the the Yule Walker/GLS Method (Judge, 1985) which is based on using the sample autocorrelation estimates to construct an estimated covariance matrix with which GLS can be performed. The benefit of this approach is that more of the data can be used for the estimates whereas the Iterated Cochrane-Orcutt method is restricted to using only $n-p+1$ of the observations for each estimate. However this comes at the expense of added complexity.

The first step is define the regressor matrix. Consider the following linear model with AR(p) errors:

$$\begin{aligned}\beta_{ik} &= A_{ik}\mu_k + \eta_{ik}\\ \eta_{ik} &= \sum_{r=1}^{p} \varphi_{rk}\eta_{i-r,k} + \upsilon_{ik},\end{aligned}$$

where A_{ik} is a $1 \times d$ vector containing the regressor variables for observation i, and μ_k is a $d \times 1$ vector of coefficients. Collecting the regressors into the matrix $\mathbf{A}_k$:

$$\mathbf{A}_k \equiv \begin{bmatrix} A_{1,k} \\ A_{2,k} \\ \vdots \\ A_{n,k} \end{bmatrix}. \tag{2.26}$$

The auto-covariances are denoted as $\gamma_{k,s} \equiv E\eta_{ik}\eta_{i-s,k}$ for $i, s = 1, \ldots, n$. Recall the auto-covariances of an AR(p) process from Lemma 2.4.3:

$$\gamma_{k,s} = \begin{cases} \varphi_{k,1}\gamma_{k,s-1} + \varphi_{k,2}\gamma_{k,s-2} + \ldots \varphi_{k,p}\gamma_{k,s-p} & \text{for } s = 1, 2, \ldots \\ \varphi_{k,1}\gamma_{k,1} + \varphi_{k,1}\gamma_{k,2} + \ldots \varphi_{k,1}\gamma_{k,p} + \sigma_k^2 & \text{for } s = 0. \end{cases}$$

Defining $\gamma_{k,0} \equiv \sigma_k^2$ and the auto-correlation as $\rho_{k,s} \equiv \gamma_{k,s}/\gamma_{k,0}$, then dividing the above expression by $\gamma_{k,0}$ gives what are known as the Yule Walker Equations:

$$\rho_{k,s} = \quad \varphi_{k,1}\rho_{k,s-1} + \varphi_{k,2}\rho_{k,s-2} + \ldots \varphi_{k,p}\rho_{k,s-p} \quad \text{for } s = 1, 2, \ldots$$

Let η_{ik} denote the vector $[\eta_{1,k}, \ldots, \eta_{nk}]'$. Then OLS estimation of the model $\boldsymbol{\beta}_k = \mathbf{A}_k\mu_k + \eta_k$ produces the OLS estimates $\hat{\mu}_k$, and $\hat{\sigma}_k^2$ as

$$\hat{\mu_k} = [\mathbf{A}_k'\mathbf{A}_k]^{-1}\mathbf{A}_k'\boldsymbol{\beta}_k,$$

and

$$\hat{\sigma_k}^2 = \frac{1}{n-d}\|\boldsymbol{\beta}_k - \mathbf{A}_k\hat{\mu_k}\|^2.$$

Then the residuals are calculated as $\hat{\eta}_{ik} = \beta_{ik} - A_{ik}\hat{\mu_k}$. With these, the sample auto-correlations can be calculated:

$$\hat{\rho}_{k,s} = \frac{\frac{1}{n}\sum_{i=s+1}^{n}\hat{\eta}_{ik}\hat{\eta}_{i-s,k}}{\frac{1}{n}\sum_{i=1}^{n}\hat{\eta}_{ik}^2} = \frac{\sum_{i=s+1}^{n}\hat{\eta}_{ik}\hat{\eta}_{i-s,k}}{\sum_{i=1}^{n}\hat{\eta}_{ik}^2},$$

for $s = 0, 1, 2, \ldots, n-1$. These estimated $\{\hat{\rho}_{k,s}\}$ are used in the Yule Walker Equations which in turn yields estimates for the AR parameters $\{\varphi_{k,s}\}$. Next these are used with the OLS estimate $\hat{\sigma}_k^2$ to obtain an estimate of the covariance matrix, $\hat{\Sigma}_k$ (recall the expression from Lemma 2.4.3. Finally, with an estimate of Σ_k, GLS can be used on the regression $\hat{\Sigma}_k^{-1/2}\boldsymbol{\beta}_k = [\hat{\Sigma}_k^{-1/2}\mathbf{A}_k]\mu_k + \hat{\Sigma}_k^{-1/2}\eta_k$ to get estimates for μ_k that account for the auto-correlation in the error:

$$\hat{\mu_k} = [\mathbf{A}_k'\hat{\Sigma}_k^{-1}\mathbf{A}_k]^{-1}\mathbf{A}_k'\hat{\Sigma}_k^{-1}\boldsymbol{\beta}_k.$$

Following either estimation method for the AR parameters, the EM estimation for the entire FDFM proceeds as before.

This concludes Chapter 2. The chapter began with the formulation of an exciting new class of models named functional dynamic factor models. Based on the model assumptions, the joint distribution of the observed data was derived, which facilitated the expression for the likelihood function. Using a smoothness penalty approach a penalized log-likelihood was developed from which maximum likelihood could provide parameter estimates. Finally an EM estimation for the model was proposed. This included detailed descriptions of starting values, the E-step and the M-step which included smoothing parameter selection. The chapter then closed with some useful computational results and some alternative model specifications.

The next chapter focuses on the derivations of many of the mathematical results, lemmas and propositions used in this chapter, and is organized in a similar manner.

Chapter 3

Implementation and Derivations

The present chapter derives aforementioned results from Chapter 2 in greater detail. The reader is of course invited to proceed, but the content herein need not be perused in order to continue on to the following chapters covering simulation results, applied data results and future work. Chapter 2 discussed the framework of the functional dynamic factor model from the specification to the likelihood to the implementation. In this chapter, several results of that chapter are revisited with more technical detail, discussed further and expanded upon. Various M-step results are then revisited in greater detail, followed by an equally detailed discussion of the GCV component of the EM estimation. The latter of these also include description of another of the computational efficiencies presented in Section 2.4.6, complete with a cautionary tale regarding computing packages. Finally, a special case of the FDFM is presented for use in the sequential specification outlined in Section 2.5.1.

3.1 The E-Step and Matrix Results

In this section are several derivations of significant results presented in Section 2.4; roughly in the same order as displayed in that chapter. First derived are the unconditional moments of the observed data and factor time series, with the help of some vector notation. Next some of the useful matrix structures alleged in the computational efficiency section are proven.

3.1.1 Vector-ized Model Expression

All of the matrix results presented in this section depend upon the vector-ized specification of the model. Recall Lemma 2.4.2 from Magnus and Neudecker (1999) where a product of matrices has an equivalent vector representation using the *vec* operator and the kronecker product. Then given the original matrix representation of the model $\mathbf{X} = \mathbf{BF} + \epsilon$,

$$\begin{aligned} vec(\mathbf{X}) &= vec(\mathbf{BF}) + vec(\epsilon) \\ &= (\mathbf{F}' \otimes \mathbf{I}_n) vec(\mathbf{B}) + vec(\epsilon). \end{aligned} \tag{3.1}$$

Then recalling the notational convention that $\boldsymbol{X} \equiv vec(\mathbf{X})$ and $\boldsymbol{\beta} \equiv vec(\mathbf{B})$, the vector-ized model expression from section 2.4.3 immediately follows:

$$\boldsymbol{X} = (\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta} + vec(\epsilon).$$

3.1.2 Proof of Proposition 2.4.1

This section derives the five results presented in Proposition 2.4.1 regarding the unconditional means and variances of $\boldsymbol{X}$ and $\boldsymbol{\beta}$.

The mean and variance of the factor time series are as follow:

Factor time series' mean, $\mu_{\boldsymbol{\beta}}$:

Recall that $\mu_{\boldsymbol{\beta}} = E[\boldsymbol{\beta}] = E[vec(\mathbf{B})]$. From Lemma 2.4.3, $\mu_k \equiv E[\beta_{ik}] = c_k / \left[1 - \left(\sum_{r=1}^{P} \varphi_{r,k}\right)\right]$.

$$\mu_{\boldsymbol{\beta}} = \begin{bmatrix} E[\boldsymbol{\beta}_1]_{n\times 1} \\ \vdots \\ E[\boldsymbol{\beta}_K]_{n\times 1} \end{bmatrix}_{nK\times 1} = \begin{bmatrix} \mu_1 \cdot \mathbf{1}_n \\ \vdots \\ \mu_K \cdot \mathbf{1}_n \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_K \end{bmatrix}_{K\times 1} \otimes \mathbf{1}_n.$$

Let $\mathbf{c}$ be the $K \times 1$ vector with elements $[\mu_1, \ldots, \mu_K]'$; then the result $\mu_{\boldsymbol{\beta}} = \mathbf{c} \otimes \mathbf{1}_n$ immediately follows.

Factor time series' covariance, $\Sigma_{\boldsymbol{\beta}}$:

Recall $Var(\boldsymbol{\beta}_k) \equiv \Sigma_k$ for each $k = 1, \ldots, k$. With the assumption of independent factor time series,

$$cov(\boldsymbol{\beta}_k, \boldsymbol{\beta}_{k'}) = \begin{cases} \Sigma_k & \text{for } k = k' \\ \mathbf{0}_{n\times n} & \text{otherwise} \end{cases} .$$

It follows then that the $nK \times nK$ covariance matrix for all factors, $\Sigma_{\boldsymbol{\beta}}$ is a block diagonal matrix with K order n matrices $\Sigma_1, \ldots \Sigma_K$.

Next, the mean and variance of the observed data are as follow. These results depend on the vector version of the model:

$$\boldsymbol{X} = (\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta} + vec(\epsilon).$$

Observed data mean, $\mu_{\mathbf{X}}$:

$$\begin{aligned} \mu_{\mathbf{X}} = E[\boldsymbol{X}] &= E[(\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta} + vec(\epsilon)] \\ &= (\mathbf{F}' \otimes \mathbf{I}_n)E[\boldsymbol{\beta}] + \mathbf{0}_{nm\times 1} \\ &= (\mathbf{F}' \otimes \mathbf{I}_n)\mu_{\boldsymbol{\beta}}. \end{aligned}$$

Observed data covariance, $\Sigma_{\mathbf{X}}$:

Because the factor time series $\boldsymbol{\beta}$ is assumed to be independent from the error term ϵ, the variance matrix for $\boldsymbol{X}$ is as follows:

Fact. $\Sigma_{\mathbf{X}} = (\mathbf{F}' \otimes \mathbf{I}_n)\Sigma_\beta(\mathbf{F} \otimes \mathbf{I}_n) + \sigma^2\mathbf{I}_{nm}$.

Proof.

$$\begin{aligned} \Sigma_{\mathbf{X}} = Var(\boldsymbol{X}) &= Var[(\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta} + vec(\epsilon)] \\ &= Var[(\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta}] + Var[vec(\epsilon)] \\ &= (\mathbf{F}' \otimes \mathbf{I}_n)Var[\boldsymbol{\beta}](\mathbf{F}' \otimes \mathbf{I}_n)' + Var[vec(\epsilon)] \\ &= (\mathbf{F}' \otimes \mathbf{I}_n)\Sigma_\beta(\mathbf{F} \otimes \mathbf{I}_n) + \sigma^2\mathbf{I}_{nm} \qquad \square \end{aligned}$$

Factor and observed data covariance, $\Sigma_{\boldsymbol{\beta},\mathbf{X}}$:

Fact. $\Sigma_{\mathbf{X},\beta} = (\mathbf{F}' \otimes \mathbf{I}_n)\Sigma_\beta$.

Proof.

$$\begin{aligned}
\Sigma_{\beta,\mathbf{X}} &= Cov[\boldsymbol{\beta}, \boldsymbol{X}] = Cov[\boldsymbol{\beta}, (\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta} + vec(\epsilon)] \\
&= Cov[\boldsymbol{\beta}, (\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta}] \\
&= Cov[\boldsymbol{\beta}, \boldsymbol{\beta}](\mathbf{F}' \otimes \mathbf{I}_n)' \\
&= \Sigma_\beta(\mathbf{F} \otimes \mathbf{I}_n).
\end{aligned}$$

This, of course, also implies that $\Sigma_{\mathbf{X},\beta} = (\mathbf{F}' \otimes \mathbf{I}_n)\Sigma_\beta$. □

3.1.3 Inversion of $\Sigma_{\mathbf{X}}$

In Section 2.4.6 it was proclaimed that the inversion of the rather large $\Sigma_{\mathbf{X}}$ could be expedited with the help of a particular factorization and some convenient matrix structure. That result is shown here; it is accomplished with the help of the Sherman–Morrison–Woodbury formula presented in Lemma 2.4.4:

$$(A^{-1} + C'B^{-1}C)^{-1} = A - AC'(CAC' + B)^{-1}CA.$$

Let

$$\begin{aligned}
A &= \sigma^{-2}\mathbf{I}_{nm}, \\
B &= \Sigma_\beta^{-1}, \\
\text{and} \quad C &= (\mathbf{F} \otimes \mathbf{I}_n).
\end{aligned}$$

Then

$$\Sigma_{\mathbf{X}}^{-1} = (A^{-1} + C'B^{-1}C)^{-1} = A - AC'(CAC' + B)^{-1}CA.$$

Recall the orthogonal-ization of the factor loading curves so that $\mathbf{FF}' = \mathbf{I}_K$. This yields the following convenient result:

$$(\mathbf{F} \otimes \mathbf{I}_n)(\mathbf{F}' \otimes \mathbf{I}_n) \quad = (\mathbf{F}\mathbf{F}' \otimes \mathbf{I}_n) = (\mathbf{I}_K \otimes \mathbf{I}_n) = \mathbf{I}_{nK}.$$

Therefore,

$$\begin{aligned} CAC' + B &= (\mathbf{F} \otimes \mathbf{I}_n)[\sigma^{-2}\mathbf{I}_{nm}](\mathbf{F}' \otimes \mathbf{I}_n) + \Sigma_\beta^{-1} \\ &= \sigma^{-2}(\mathbf{F} \otimes \mathbf{I}_n)(\mathbf{F}' \otimes \mathbf{I}_n) + \Sigma_\beta^{-1} \\ &= \sigma^{-2}(\mathbf{F}\mathbf{F}' \otimes \mathbf{I}_n) + \Sigma_\beta^{-1} \\ &= \sigma^{-2}(\mathbf{I}_K \otimes \mathbf{I}_n) + \Sigma_\beta^{-1} \\ &= \sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}. \end{aligned}$$

Because Σ_β is a block-diagonal matrix, Σ_β^{-1} is also block-diagonal, obtained by inverting the K individual $n \times n$ blocks on the diagonal. Adding $\sigma^{-2}\mathbf{I}_{nK}$ to Σ_β^{-1} simply adds a scalar to each of the blocks on the diagonal; this maintains block-diagonality in the composite $\sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}$. Therefore, its inverse, $\left[\sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]^{-1}$, is also block-diagonal. The inverse is found in the same manner, which is inversion of the K individual $n \times n$ blocks.

This completes the result that the inversion of $\Sigma_\mathbf{X}$ is achieved by K sequential order n inversions. For completeness, the entire expression for $\Sigma_\mathbf{X}^{-1}$ using the Sherman–Morrison–Woodbury formula is continues below.

Next

$$AC' = (\sigma^{-2}\mathbf{I}_{nm})(\mathbf{F}' \otimes \mathbf{I}_n) = \sigma^{-2}(\mathbf{F}' \otimes \mathbf{I}_n),$$

and

$$CA = (\mathbf{F} \otimes \mathbf{I}_n)(\sigma^{-2}\mathbf{I}_{nm}) = \sigma^{-2}(\mathbf{F} \otimes \mathbf{I}_n).$$

Therefore

$$AC'(CAC' + B)^{-1}CA \quad = \quad \sigma^{-2}(\mathbf{F}' \otimes \mathbf{I}_n)\left[\sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]^{-1}\sigma^{-2}(\mathbf{F} \otimes \mathbf{I}_n).$$

Finally, combining all of these components yields the expression for $\Sigma_{\mathbf{X}}^{-1}$:

$$\begin{aligned}\Sigma_{\mathbf{X}}^{-1} &= (A^{-1} + C'B^{-1}C)^{-1} = A - AC'(CAC' + B)^{-1}CA \\ &= \sigma^{-2}\mathbf{I}_{nm} - \sigma^{-4}(\mathbf{F}' \otimes \mathbf{I}_n)\left[\sigma^{-2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]^{-1}(\mathbf{F} \otimes \mathbf{I}_n).\end{aligned}$$

3.1.4 $\Sigma_{\boldsymbol{\beta}|\mathbf{X}}$ is Block Diagonal

In Section 2.4.6 it was alleged that $\Sigma_{\boldsymbol{\beta}|\mathbf{X}}$ is block-diagonal. The derivation of that result is shown here, followed by the result that the conditional expectation of the product of two different factors is the product of those conditional expectations.

For $\Sigma_{\beta|\mathbf{X}} = \Sigma_\beta - \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\Sigma_{\mathbf{X},\beta}$, consider the second term in this expression. Using the results about $\Sigma_{\mathbf{X}}^{-1}$ from the previous section and $\Sigma_{\boldsymbol{\beta},\mathbf{X}}$ from earlier in the chapter,

$$\Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\Sigma_{\mathbf{X},\beta} = \Sigma_\beta(\mathbf{F} \otimes \mathbf{I}_n)\left\{\frac{1}{\sigma^2}\mathbf{I}_{nm} - \frac{1}{\sigma^4}(\mathbf{F}' \otimes \mathbf{I}_n)\left[\frac{1}{\sigma^2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]^{-1}(\mathbf{F} \otimes \mathbf{I}_n)\right\}(\mathbf{F}' \otimes \mathbf{I}_n)\Sigma_\beta.$$

Distributing the terms simplifies the expression:

$$\begin{aligned}\Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\Sigma_{\mathbf{X},\beta} &= \Sigma_\beta(\mathbf{F} \otimes \mathbf{I}_n)\left[\frac{1}{\sigma^2}\mathbf{I}_{nm}\right](\mathbf{F}' \otimes \mathbf{I}_n)\Sigma_\beta \\ &- \frac{1}{\sigma^4}\Sigma_\beta(\mathbf{F} \otimes \mathbf{I}_n)(\mathbf{F}' \otimes \mathbf{I}_n) \\ &\times \left[\frac{1}{\sigma^2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]^{-1} \\ &\times (\mathbf{F} \otimes \mathbf{I}_n)(\mathbf{F}' \otimes \mathbf{I}_n)\Sigma_\beta \\ &= \frac{1}{\sigma^2}\Sigma_\beta\Sigma_\beta - \frac{1}{\sigma^4}\Sigma_\beta\left[\frac{1}{\sigma^2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]^{-1}\Sigma_\beta.\end{aligned}$$

Subtracting this expression from $\Sigma_{\boldsymbol{\beta}}$ yields the alternative from for $\Sigma_{\beta|\mathbf{X}}$:

$$\Sigma_{\beta|\mathbf{X}} = \Sigma_\beta - \frac{1}{\sigma^2}\Sigma_\beta\Sigma_\beta + \frac{1}{\sigma^4}\Sigma_\beta\left[\frac{1}{\sigma^2}\mathbf{I}_{nK} + \Sigma_\beta^{-1}\right]^{-1}\Sigma_\beta.$$

Each of these three terms are block diagonal with K blocks of size $n \times n$. Therefore $\Sigma_{\beta|\mathbf{X}}$ is block diagonal.

As an aside, it would be even more useful if $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}] = \Sigma_{\beta|\mathbf{X}} + \mu_{\beta|\mathbf{X}}\mu'_{\beta|\mathbf{X}}$ were block diagonal. Unfortunately this is not the case.

To show this in a swift manner, assume for the moment that all of the K factor time series have mean zero (rather than an intercept or any regressors). Then

$$\mu_{\mathbf{X}} = (\mathbf{F}' \otimes \mathbf{I}_n)E[\boldsymbol{\beta}] + E[\epsilon] = (\mathbf{F}' \otimes \mathbf{I}_n)\mu_{\beta} = \mathbf{0}_{nm\times 1}.$$

Therefore

$$\mu_{\beta|\mathbf{X}} = \mu_{\beta} + \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}(\boldsymbol{X} - \mu_{\mathbf{X}}) = \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\boldsymbol{X}.$$

This implies

$$\mu_{\beta|\mathbf{X}}\mu'_{\beta|\mathbf{X}} \quad = \quad \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\boldsymbol{X}\boldsymbol{X}'\Sigma_{\mathbf{X}}^{-1}\Sigma_{\mathbf{X},\beta}$$

Which, unfortunately, due to the $\boldsymbol{X}\boldsymbol{X}'$ in the center will not generally have a block diagonal form. Thus, $\Sigma_{\beta|\mathbf{X}}$ is not block diagonal.

Moving on, in the maximum likelihood solutions for several of the model parameters and factor loading curves, expressions involving between-factor cross products often arise. It was asserted in Section 2.4.6 that for two distinct factors h and k that

$$E[(\boldsymbol{\beta}_k\boldsymbol{\beta}'_h)|\boldsymbol{X}] = E[\boldsymbol{\beta}_k|\boldsymbol{X}]E[\boldsymbol{\beta}'_h|\boldsymbol{X}].$$

This follows from the block diagonal structure of $\Sigma_{\boldsymbol{\beta}|\mathbf{X}}$. Specifically, from

$$E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}] = \Sigma_{\boldsymbol{\beta}|\mathbf{X}} + \mu_{\boldsymbol{\beta}|\mathbf{X}}\mu'_{\boldsymbol{\beta}|\mathbf{X}},$$

$E[(\boldsymbol{\beta}_k\boldsymbol{\beta}'_h)|\boldsymbol{X}]$ is an off-diagonal block of $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}]$. Off diagonal blocks of $\Sigma_{\boldsymbol{\beta}|\mathbf{X}}$ are $\mathbf{0}_{n\times n}$. $\mu_{\boldsymbol{\beta}|\mathbf{X}}$

consists of the following components:

$$\mu_{\boldsymbol{\beta}|\mathbf{X}} = \begin{bmatrix} E[\boldsymbol{\beta}_1|\boldsymbol{X}]_{n\times 1} \\ \vdots \\ E[\boldsymbol{\beta}_K|\boldsymbol{X}]_{n\times 1} \end{bmatrix}_{nK\times 1}.$$

Thus

$$E[(\boldsymbol{\beta}_k\boldsymbol{\beta}_h')|\boldsymbol{X}] = \mathbf{0}_{n\times n} + E[\boldsymbol{\beta}_k|\boldsymbol{X}]E[\boldsymbol{\beta}_h'|\boldsymbol{X}].$$

3.2 M-step I: Factor Loading Curves

This section contains more detail about the maximum likelihood solutions for the factor loading curves covered in Section 2.4. The discussion begins with a thorough derivation of the factor loading curve solutions. Then, the GCV procedure used to select the smoothing parameters is derived. The section concludes with justification for the eigen-decomposition method proposed in Section 2.4.6, including a brief statement concerning the practical application using various software packages. Other M-step results are deferred until the next section.

3.2.1 Proof of Proposition 2.4.3

Recall the section of the penalized log-likelihood (2.12) involving the factor loading curves $\{\mathbf{f}_k\}$:

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - \sum_{k=1}^{K}\beta_{ik}f_{kj})^2 + \sum_{k=1}^{K}\lambda_k\mathbf{f}_k\Omega_k\mathbf{f}_k'.$$

Recall the block diagonal $mK \times mK$ matrix $\mathbf{S}$ with K blocks $\lambda_k\Omega_k$, $k = 1, \ldots, K$ (that is, $\mathbf{S} = diag\{\lambda_k\Omega_k\}$. Then

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - \sum_{k=1}^{K}\beta_{ik}f_{kj})^2 + \lambda\sum_{k=1}^{K}\mathbf{f}_k\Omega\mathbf{f}_k' \quad = \quad \frac{1}{\sigma^2}\|vec(\mathbf{X} - \mathbf{BF})\|^2 + [vec(\mathbf{F}')']\mathbf{S}vec(\mathbf{F}').$$

Using the result of Lemma 2.4.2 (Magnus and Neudecker, 1999) yields

$$\frac{1}{\sigma^2}\|vec(\mathbf{X}-\mathbf{BF})\|^2+[vec(\mathbf{F}')']\mathbf{S}vec(\mathbf{F}')=\frac{1}{\sigma^2}\|vec(\mathbf{X}')-(\mathbf{B}\otimes\mathbf{I}_m)vec(\mathbf{F}')\|^2+[vec(\mathbf{F}')']\mathbf{S}vec(\mathbf{F}').$$

Let $\tilde{X}\equiv vec(\mathbf{X}')$, $\tilde{F}\equiv vec(\mathbf{F}')$, and $\mathbf{Z}\equiv\mathbf{B}\otimes\mathbf{I}_m$. Then

$$\begin{aligned}\frac{1}{\sigma^2}\|vec(\mathbf{X}')-(\mathbf{B}\otimes\mathbf{I}_m)vec(\mathbf{F}')\|^2+[vec(\mathbf{F}')']\mathbf{S}vec(\mathbf{F}') &= \frac{1}{\sigma^2}(\tilde{X}-\mathbf{Z}\tilde{F})'(\tilde{X}-\mathbf{Z}\tilde{F})+\tilde{F}'\mathbf{S}\tilde{F}\\ &= \frac{1}{\sigma^2}\tilde{X}'\tilde{X}-\frac{2}{\sigma^2}\tilde{F}'\mathbf{Z}'\tilde{X}+\tilde{F}'\left(\frac{\mathbf{Z}'\mathbf{Z}}{\sigma^2}+\mathbf{S}\right)\tilde{F}.\end{aligned}$$

Differentiating with respect to the vector $\tilde{F}'$ and setting the result equal to zero yields the simultaneous solutions for all the factor loading curves.

$$\begin{aligned}\frac{\mathbf{d}}{\mathbf{d}\tilde{F}'}\left[\frac{1}{\sigma^2}\tilde{X}'\tilde{X}-\frac{2}{\sigma^2}\tilde{F}'\mathbf{Z}'\tilde{X}+\tilde{F}'\left(\frac{\mathbf{Z}'\mathbf{Z}}{\sigma^2}+\mathbf{S}\right)\tilde{F}\right]&=0\\ \hat{\tilde{F}}=\left(\frac{\mathbf{Z}'\mathbf{Z}}{\sigma^2}+\mathbf{S}\right)^{-1}_{mK\times mK}\frac{1}{\sigma^2}\mathbf{Z}'\tilde{X}.\end{aligned}$$

In the special case where $\lambda_k=\lambda\ \forall k$, $\mathbf{S}$ becomes $\lambda(\mathbf{I}_K\otimes\Omega)$, but otherwise the result is the same.

3.2.2 Individual Loading Curves

In Section 2.4.4 it was suggested to solve for the factor loading curves in a sequential manner. To do this requires an expression for a distinct $\mathbf{f}_k$; this is given below.

As shown in the previous section, $vec(\mathbf{F}'\mathbf{B}')=\mathbf{B}\otimes\mathbf{I}_m\cdot vec(\mathbf{F}')$ and the vector-ized model is expressed as $vec(\mathbf{X}')=\mathbf{B}\otimes\mathbf{I}_m\cdot vec(\mathbf{F}')+vec(\epsilon)$. The solution for all of the factor loading curves is given by

$$\hat{\tilde{F}}=\left(\frac{\mathbf{Z}'\mathbf{Z}}{\sigma^2}+\mathbf{S}\right)^{-1}\frac{1}{\sigma^2}\mathbf{Z}'\tilde{X}=\left[\frac{1}{\sigma^2}(\mathbf{B}'\mathbf{B}\otimes\mathbf{I}_m)+diag\left\{\lambda_k\Omega_k\right\}\right]^{-1}\frac{1}{\sigma^2}(\mathbf{B}'\otimes\mathbf{I}_m)\tilde{X}.$$

Equivalently,

$$\left[\frac{1}{\sigma^2}(\mathbf{B}'\mathbf{B}\otimes\mathbf{I}_m)+diag\left\{\lambda_k\Omega_k\right\}\right]\hat{\tilde{F}} \;=\; \frac{1}{\sigma^2}(\mathbf{B}'\otimes\mathbf{I}_m)\tilde{X}.$$

Expanding both sides of the equation,

$$\left\{\frac{1}{\sigma^2}\begin{bmatrix} \|\boldsymbol{\beta}_1\|^2\mathbf{I}_m & & \langle\boldsymbol{\beta}_k,\boldsymbol{\beta}_{k'}\rangle\mathbf{I}_m \\ & \ddots & \\ \langle\boldsymbol{\beta}_{k'},\boldsymbol{\beta}_k\rangle\mathbf{I}_m & & \|\boldsymbol{\beta}_K\|^2\mathbf{I}_m \end{bmatrix} + \begin{bmatrix} \lambda_1\Omega_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_K\Omega_K \end{bmatrix}\right\}\begin{bmatrix} \hat{\mathbf{f}}'_1 \\ \vdots \\ \hat{\mathbf{f}}'_K \end{bmatrix} = \frac{1}{\sigma^2}\begin{bmatrix} \sum_{i=1}^n \beta_{1i}X_i \\ \vdots \\ \sum_{i=1}^n \beta_{Ki}X_i \end{bmatrix}.$$

Finally, by simplifying the left side of the equation,

$$\begin{bmatrix} \frac{\|\boldsymbol{\beta}_1\|^2}{\sigma^2}\mathbf{I}_m + \lambda_1\Omega_1 & & \frac{1}{\sigma^2}\langle\boldsymbol{\beta}_k,\boldsymbol{\beta}_{k'}\rangle\mathbf{I}_m \\ & \ddots & \\ \frac{1}{\sigma^2}\langle\boldsymbol{\beta}_{k'},\boldsymbol{\beta}_k\rangle\mathbf{I}_m & & \frac{\|\boldsymbol{\beta}_K\|^2}{\sigma^2}\mathbf{I}_m + \lambda_K\Omega_K \end{bmatrix}\begin{bmatrix} \hat{\mathbf{f}}'_1 \\ \vdots \\ \hat{\mathbf{f}}'_K \end{bmatrix} = \frac{1}{\sigma^2}\begin{bmatrix} \sum_{i=1}^n \beta_{1i}X_i \\ \vdots \\ \sum_{i=1}^n \beta_{Ki}X_i \end{bmatrix}.$$

Let k index a specific factor loading curve and $h \neq k$ index all other factor loading curves, $h, k = 1, \ldots, K$. Then using the expression above,

$$\left(\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k\Omega_k\right)\hat{\mathbf{f}}'_k + \frac{1}{\sigma^2}\sum_{h\neq k}\langle\boldsymbol{\beta}_k,\boldsymbol{\beta}_h\rangle\hat{\mathbf{f}}'_h \;=\; \frac{1}{\sigma^2}\sum_{i=1}^n \beta_{ki}X_i.$$

Solving for $\mathbf{f}_k$:

$$\hat{\mathbf{f}}'_k \;=\; \frac{1}{\sigma^2}\left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k\Omega_k\right]^{-1}\left[\sum_{i=1}^n \beta_{ki}X_i - \sum_{h\neq k}\langle\boldsymbol{\beta}_k,\boldsymbol{\beta}_h\rangle\hat{\mathbf{f}}'_h\right]. \tag{3.2}$$

3.2.3 GCV Selection

In Section 2.4.5 it was shown that solving for an optimal $\lambda_k/\mathbf{f}_k$ pair is equivalent to a ridge regression formulation, for which generalized cross validation is a natural choice. Here, the connection is made explicit.

To see this, let $h, k = 1, \ldots, K$ and define

$$X_i^{-k} \equiv X_i - \sum_{h\neq k}\beta_{ih}\hat{\mathbf{f}}'_h, \tag{3.3}$$

where X_i is an $m \times 1$ column vector corresponding to the ith *row* of $\mathbf{X}$. Let $\tilde{X}^{-k}$ denote the $nm \times 1$ vector consisting of the stacked vectors X_i^{-k} for $i = 1, \ldots, n$. Then multiplying both sides of (3.3) by β_{ik} yields

$$\beta_{ik} X_i^{-k} \quad = \quad \beta_{ik} X_i - \beta_{ik} \sum_{h \neq k} \beta_{ih} \hat{\mathbf{f}}_h'.$$

Next, sum over $i = 1, \dots n$:

$$\begin{aligned} \sum_{i=1}^{n} \beta_{ik} X_i^{-k} &= \sum_{i=1}^{n} \beta_{ik} X_i - \sum_{i=1}^{n} \beta_{ik} \sum_{h \neq k} \beta_{ih} \hat{\mathbf{f}}_h' \\ &= \sum_{i=1}^{n} \beta_{ik} X_i - \sum_{h \neq k} \langle \boldsymbol{\beta}_k, \boldsymbol{\beta}_h \rangle \hat{\mathbf{f}}_h'. \end{aligned}$$

Because $(\boldsymbol{\beta}_k' \otimes \mathbf{I}_m) \tilde{X}^{-k} = \sum_{i=1}^{n} \beta_{ki} X_i^{-k}$, this implies

$$(\boldsymbol{\beta}_k' \otimes \mathbf{I}_m) \tilde{X}^{-k} \quad = \quad \sum_{i=1}^{n} \beta_{ki} X_i - \sum_{h \neq k} \langle \boldsymbol{\beta}_k, \boldsymbol{\beta}_h \rangle \hat{\mathbf{f}}_h'. \tag{3.4}$$

Finally, using (3.4), the original expression (3.2):

$$\hat{\mathbf{f}}_k' \quad = \quad \frac{1}{\sigma^2} \left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2} \mathbf{I}_m + \lambda_k \Omega_k \right]^{-1} \left[\sum_{i=1}^{n} \beta_{ki} X_i - \sum_{h \neq k} \langle \boldsymbol{\beta}_k, \boldsymbol{\beta}_h \rangle \hat{\mathbf{f}}_h' \right],$$

simplifies to the alternative expression for $\mathbf{f}_k$:

$$\hat{\mathbf{f}}_k' = \left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2} \mathbf{I}_m + \lambda_k \Omega_k \right]^{-1} \frac{1}{\sigma^2} (\boldsymbol{\beta}_k' \otimes \mathbf{I}_m) \tilde{X}^{-k}. \tag{3.5}$$

Since $\|\boldsymbol{\beta}_k\|^2 \mathbf{I}_m = (\boldsymbol{\beta}_k' \boldsymbol{\beta}_k \otimes \mathbf{I}_m) = (\boldsymbol{\beta}_k \otimes \mathbf{I}_m)'(\boldsymbol{\beta}_k \otimes \mathbf{I}_m)$, equation (3.5) is equivalently expressed as

$$\hat{\mathbf{f}}_k' = \left[(\frac{1}{\sigma} \boldsymbol{\beta}_k \otimes \mathbf{I}_m)'(\frac{1}{\sigma} \boldsymbol{\beta}_k \otimes \mathbf{I}_m) + \lambda_k \Omega_k \right]^{-1} (\frac{1}{\sigma} \boldsymbol{\beta}_k \otimes \mathbf{I}_m)'(\frac{1}{\sigma} \tilde{X}^{-k}).$$

This is exactly the solution to the ridge regression problem presented in Section 2.4.5:

$$\left\| \frac{1}{\sigma} \tilde{X}^{-k} - \frac{1}{\sigma} (\boldsymbol{\beta}_k \otimes \mathbf{I}_m) \cdot \mathbf{f}_k' \right\|^2 + \lambda_k \mathbf{f}_k \Omega \mathbf{f}_k'.$$

To see this, consider a generic formulation of a ridge regression for some dependent variable vector $\mathbf{z}_{T\times 1}$, regressors $\mathbf{W}_{T\times q}$ and coefficients $\gamma_{q\times 1}$ with error vector $\mathbf{v}_{T\times 1}$: $\mathbf{z} = \mathbf{W}\gamma + \mathbf{v}$. Suppose the ridge penalty is $\alpha\gamma'\Psi\gamma$. Then the the optimal coefficient vector solves the problem:

$$\hat{\gamma} \quad = \quad \arg\min \|\mathbf{z} - \mathbf{W}\gamma\|^2 + \alpha\gamma'\Psi\gamma = [\mathbf{W}'\mathbf{W} + \alpha\Psi]^{-1}\mathbf{W}'\mathbf{z},$$

and $\hat{\mathbf{z}} = \mathbf{W}\hat{\gamma}$. According to Green and Silverman (1994), selection of the ridge penalty α is achieved via generalized cross validation for which the optimal α minimizes the criterion

$$\mathrm{GCV}(\alpha) = \frac{\|\mathbf{z} - \hat{\mathbf{z}}\|^2/T}{[1 - \mathrm{tr}(H_\alpha)/T]^2} = \frac{\|(\mathbf{I}_T - H_\alpha)\mathbf{z}\|^2/T}{[1 - \mathrm{tr}(H_\alpha)/T]^2},$$

where H_α is the modified "hat" matrix

$$H_\alpha \quad = \quad \mathbf{W}[\mathbf{W}'\mathbf{W} + \alpha\Psi]^{-1}\mathbf{W}'.$$

Expanding upon the assertions made in Section 2.4.5, the similarities of this generic problem with the specific one are evident:

Dimensions: $T = nm$; $q = m$.

Dependent variable: $\mathbf{z} = \frac{1}{\sigma}\tilde{X}^{-k}$.

Independent variables: $\mathbf{W} = \frac{1}{\sigma}(\boldsymbol{\beta}_k \otimes \mathbf{I}_m)$.

Parameters: $\gamma = \mathbf{f}_k'$.

Ridge penalty term: $\alpha\gamma'\Psi\gamma = \lambda_k \mathbf{f}_k \Omega \mathbf{f}_k'$.

Therefore, in the current setting the modified hat matrix is

$$\begin{aligned} H_{\lambda_k} \quad &= \quad \left\{ \frac{1}{\sigma^2}(\boldsymbol{\beta}_k \otimes \mathbf{I}_m) \left[\frac{1}{\sigma^2}(\boldsymbol{\beta}_k'\boldsymbol{\beta}_k \otimes \mathbf{I}_m) + \lambda_k\Omega_k \right]^{-1}_{m\times m} (\boldsymbol{\beta}_k' \otimes \mathbf{I}_m) \right\}_{nm\times nm}, \\ &= \quad \left\{ \frac{1}{\sigma^2}(\boldsymbol{\beta}_k \otimes \mathbf{I}_m) \left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k\Omega_k \right]^{-1} (\boldsymbol{\beta}_k' \otimes \mathbf{I}_m) \right\}. \end{aligned}$$

Then the GCV criterion becomes

$$\mathrm{GCV}(\lambda_k) \quad = \quad \frac{\|(\mathbf{I}_{nm} - H_{\lambda_k})\tilde{X}^{-k}\|^2/nm}{[1 - \mathrm{tr}(H_{\lambda_k})/nm]^2}.$$

As mentioned previously, the calculation of GCV(λ_k), including H_{λ_k}, can be computationally burdensome for multiple candidate values of λ_k. The details of the efficient computation of these introduced in Section 2.4.6 are expanded upon below.

3.2.4 Orthogonalization

After all of the $\mathbf{f}_k/\lambda_k$ pairs are determined, normalization is required in order to maintain the assumption of orthogonality; that $\mathbf{F}'\mathbf{F} = \mathbf{I}_K$. For each iteration l of the EM, the E-step (Section 2.4.3) provides values for the factor time series. Similarly, an application of the M-step provides the lth iteration's factor loading curves. Before moving onto the next EM iteration, though, the factor loading curves must be orthogonalized, and a commensurate adjustment must be made to the factor time series: the pre-orthogonalized factor loading curves and factors are denoted as $\tilde{\mathbf{F}}_{(l)}$ and $\tilde{\boldsymbol{\beta}}_{(l)}$, respectively, for

$$\begin{aligned} \tilde{\boldsymbol{\beta}}_{(l)} &= vec([\mu_{\boldsymbol{\beta}_1|\mathbf{X}}, \ldots, \mu_{\boldsymbol{\beta}_K|\mathbf{X}}]_{n\times n}) \\ \widetilde{\boldsymbol{\beta}\boldsymbol{\beta}'}_{(l)} &= E[\widetilde{\boldsymbol{\beta}\boldsymbol{\beta}'}|\mathbf{X}]. \end{aligned} \tag{3.6}$$

Various methods exist to do this, but the method used here is to orthogonalize the initial $\tilde{\mathbf{F}}$ by a QR decomposition via a Gram–Schmidt orthogonalization (see Lay, 2003, e.g.). For ease of notation, the $_{(l)}$ subscript is dropped.

1. First, decompose the initial factor loading estimate matrix,

$$\tilde{\mathbf{F}} \overset{QR}{=} \mathbf{Q}_{m\times K}\mathbf{R}_{K\times K},$$

where $\mathbf{Q}$ forms an orthonormal basis and $\mathbf{R}$ is upper triangular with positive diagonal entries.

2. Next, set the new $\mathbf{F} = \mathbf{Q}$.

The factor time series must be appropriately adjusted so that updated factors $\mathbf{B}$ and factor loading curves $\mathbf{F}$ maintain the relationship

$$\mathbf{B} \cdot \mathbf{F}' = \tilde{\mathbf{B}}\tilde{\mathbf{F}}'. \tag{3.7}$$

Substituting in $\tilde{\mathbf{F}}' = \mathbf{R}'\mathbf{Q}'$ and $\mathbf{F} = \mathbf{Q}$ into (3.7) yields

$$\mathbf{B}\mathbf{F}' = \tilde{\mathbf{B}}\mathbf{R}'\mathbf{Q}' = \tilde{\mathbf{B}}\mathbf{R}'\mathbf{F}'. \tag{3.8}$$

Therefore, the commensurate adjustment to $\tilde{\mathbf{B}}$ is made by setting $\mathbf{B} = \tilde{\mathbf{B}}\mathbf{R}'$.

Unfortunately, adjusting $\widetilde{\boldsymbol{\beta}\boldsymbol{\beta}'} = E[\widetilde{\boldsymbol{\beta}\boldsymbol{\beta}'}|\mathbf{X}]$ is not as straightforward as simply as setting $\boldsymbol{\beta}\boldsymbol{\beta}' = vec(\tilde{\mathbf{B}})[vec(\tilde{\mathbf{B}})]'$. See Section 2.4.4 for the discussion on within and between factor cross-products. Instead, to make the proper adjustment, recall Lemma 2.4.2 regarding the $vec(\cdot)$ operator. Then the relation

$$\mathbf{B} = \tilde{\mathbf{B}}\mathbf{R}',$$

can be written in vector-ized format

$$\boldsymbol{\beta} = (\mathbf{R} \otimes \mathbf{I}_n)\tilde{\boldsymbol{\beta}}.$$

This then implies that the updated cross product matrix is given by

$$\boldsymbol{\beta}\boldsymbol{\beta}' \quad = \quad (\mathbf{R} \otimes \mathbf{I}_n)\widetilde{\boldsymbol{\beta}\boldsymbol{\beta}}'(\mathbf{R}' \otimes \mathbf{I}_n).$$

Finally, with the updated factor loadings and factors $\mathbf{F}$, $\boldsymbol{\beta}$, and $\boldsymbol{\beta}\boldsymbol{\beta}'$, the final calculations in the M-step can be computed, leading into the next E-step.

3.2.5 Proof of Proposition 4.1.2

Recall the definition of the matrix $\mathbf{S}(\lambda_k)$ which is an inner component of the expression for the hat matrix H_{λ_k} defined in (2.22):

$$\mathbf{S}(\lambda_k) = \left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k\Omega_k\right]^{-1}.$$

It was proposed in Section 2.4.6 that

$$\mathbf{S}(\lambda_k) \quad = \quad \Gamma \cdot diag\left\{\left(\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2} + \lambda_k\delta_j\right)^{-1}\right\}\Gamma',$$

and

$$tr\{H_{\lambda_k}\} \quad = \quad \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\sum_{j=1}^{m}\frac{1}{\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2} + \lambda_k\delta_j}.$$

For the moment, drop the k subscript on the penalty matrix Ω_k; it is possible that for each of the K factors, a different penalty matrix could be employed, but in any case it will always be true that Ω_k is a symmetric, positive semi-definite matrix in the context of the FDFM. The following derivations follows the results of Huang et al. (2008): eigen-decomposition of such a symmetric, positive semi-definite matrix yields

$$\Omega_{m\times m} \quad \overset{eigen}{=} \quad \Gamma_{m\times m}\Delta_{m\times m}\Gamma'_{m\times m},$$

where Γ contains the eigen-vectors, with $\Gamma\Gamma' = \mathbf{I}_m$; and Δ is a diagonal matrix with entries $\delta_j; j = 1, \ldots, m$ along the diagonal as the eigen-values of Ω:

$$\Delta = \begin{bmatrix} \delta_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \delta_m \end{bmatrix}.$$

Using expression (3.5) for the solution of some factor loading curve $\mathbf{f}_k$:

$$\hat{\mathbf{f}}_k' = [\mathbf{S}(\lambda_k)] \left[\frac{1}{\sigma^2} (\boldsymbol{\beta}_k' \otimes \mathbf{I}_m) \tilde{X}^{-k} \right],$$

let $\alpha_k = \|\boldsymbol{\beta}_k\|^2 / \sigma^2$. Then

$$\begin{aligned} \mathbf{S}(\lambda_k) &= [\alpha_k \mathbf{I}_m + \lambda_k \Omega]^{-1} \\ &= \left[\alpha_k \Gamma \mathbf{I} \Gamma' + \lambda_k \Gamma \Delta \Gamma'\right]^{-1} = \left[\Gamma \left(\alpha_k \mathbf{I} + \lambda_k \Delta\right) \Gamma'\right]^{-1} = \Gamma \left[\alpha_k \mathbf{I} + \lambda_k \Delta\right]^{-1} \Gamma' \\ &= \Gamma \begin{bmatrix} (\alpha_k + \lambda_k \delta_1)^{-1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & (\alpha_k + \lambda_k \delta_m)^{-1} \end{bmatrix} \Gamma'. \end{aligned}$$

The benefit is that despite the drawback of calculating $\hat{\mathbf{f}}_k$ for:

- Some W possible values of λ_k,
- Each of $k = 1, \ldots, K$ factors,
- Each of some L possible EM iterations,

a faster computation exists. Instead of performing an $m \times m$ general matrix inversion $W \times K \times L$ times, eigen-decomposition on Ω need only be performed once in order to facilitate a much faster inversion of the diagonal matrix $diag\{(\alpha_k + \lambda_k \delta_m)\}$ for each possible value of λ_k.

An additional computational efficiency is gained by this result to simplify $tr(H_{\lambda_k})$ in the denominator of GCV(λ_k):

$$\begin{aligned}
tr\{H_{\lambda_k}\} &= tr\left\{\frac{1}{\sigma^2}(\boldsymbol{\beta}_k \otimes \mathbf{I}_m)\left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k\Omega_k\right]^{-1}(\boldsymbol{\beta}_k' \otimes \mathbf{I}_m)\right\} \\
&= tr\left\{\frac{1}{\sigma^2}\left[\alpha_k\mathbf{I}_m + \lambda_k\Omega_k\right]^{-1}(\boldsymbol{\beta}_k' \otimes \mathbf{I}_m)(\boldsymbol{\beta}_k \otimes \mathbf{I}_m)\right\} \\
&= tr\left\{\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\left[\alpha_k\mathbf{I}_m + \lambda_k\Omega_k\right]^{-1}\right\} = \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}tr\left\{\left[\alpha_k\mathbf{I}_m + \lambda_k\Omega_k\right]^{-1}\right\} \\
&= \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}tr\left\{\Gamma\left[\alpha_k\mathbf{I} + \lambda_k\Delta\right]^{-1}\Gamma'\right\} = \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}tr\left\{\left[\alpha_k\mathbf{I} + \lambda_k\Delta\right]^{-1}\Gamma\Gamma'\right\} \\
&= \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}tr\left\{\left[\alpha_k\mathbf{I} + \lambda_k\Delta\right]^{-1}\right\} \\
tr\{H_{\lambda_k}\} &= \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\sum_{j=1}^{m}\frac{1}{\alpha_k + \lambda_k\delta_j}.
\end{aligned}$$

This completes the proposed results from Section 2.4.6.

3.2.6 A Final Note on Numerical Precision

Consider the specific form of the penalty matrix Ω_k alluded to in Section 2.3, where Ω_k is based on the discrete second differences:

$$\lambda_k \int \left[f_k''(t)\right]^2 dt \quad \approx \quad \lambda_k \sum_{j=2}^{m-1}\left[f_{k,j-1} - 2f_{k,j} + f_{k,j+1}\right]^2.$$

Coefficients from this sum can be collected in the banded matrix

$$\omega_k' \quad \equiv \quad \begin{bmatrix} 0 & & & \dots & & & 0 \\ 1 & -2 & 1 & 0 & \dots & & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & & \dots & 0 & 1 & -2 & 1 \\ 0 & & & \dots & & & 0 \end{bmatrix}.$$

Let $\mathbf{\Omega}_k \equiv \omega_k \omega_k'$; then

$$\lambda_k \sum_{j=2}^{m-1} [f_{k,j-1} - 2f_{k,j} + f_{k,j+1}]^2 = \lambda_k \mathbf{f}_k \Omega \mathbf{f}_k'.$$

In this case, Ω_k is clearly rank deficient: the first and last columns of ω_k are equal to zero resulting in $rank(\Omega_k) = m - 2$. As such, eigen-decomposition results in two zero-valued eigenvalues. Again for the moment drop the k subscript, and let $\{\gamma_j\}; j = 1, \ldots, m$ denote the $m \times 1$ eigen-vectors of Ω. Then based on the eigen-decomposition from Section 3.2.5:

$$\begin{aligned}
\Omega &= \Gamma \Delta \Gamma' \\
&= [\gamma_1 \ldots \gamma_m] \begin{bmatrix} \delta_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \delta_m \end{bmatrix} \begin{bmatrix} \gamma_1' \\ \vdots \\ \gamma_m' \end{bmatrix} \\
&= [\gamma_1 \ldots \gamma_m] \begin{bmatrix} \Delta^*_{m-2 \times m-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{2\times 2} \end{bmatrix} \begin{bmatrix} \gamma_1' \\ \vdots \\ \gamma_m' \end{bmatrix},
\end{aligned}$$

where Δ^* is the upper $m - 2 \times m - 2$ block of Δ.

The j, lth element of Ω is represented by:

$$[\Omega]_{j,l} = [\Gamma \Delta \Gamma']_{j,l} = \sum_{q=1}^{m} \delta_q \gamma_{qj} \gamma_{lq} = \sum_{q=1}^{m-2} \delta_q \gamma_{qj} \gamma_{lq}.$$

Theoretically, these zero-valued eigenvalues are, of course, exactly zero. However in a computational sense the eigenvalues returned by a software package will be represented by a very small number, and can differ significantly from package to package depending on the underlying algorithm and precision utilized (though will still be essentially 0).

This affects the corresponding eigenvectors, γ_m and γ_{m-1}, that are returned. Although the eigenvalues are trivially small, the elements of the corresponding vectors *need not be*; these vectors can differ *significantly.* This is of little consequence in terms of the reconstruction of Ω based on the decomposition: regardless of the elements of the vectors, multiplication by the close-to-zero eigenvalues nullifies whatever the elements are in the corresponding eigenvectors.

However, consider the context of $\mathbf{S}(\lambda_k)$:

$$\mathbf{S}(\lambda_k) \quad = \quad \Gamma \begin{bmatrix} (\alpha_k + \lambda_k \delta_1)^{-1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & (\alpha_k + \lambda_k \delta_m)^{-1} \end{bmatrix} \Gamma'.$$

For $j, l, q = 1, \ldots, m$

$$\begin{aligned} [\mathbf{S}(\lambda_k)]_{j,l} &= \sum_{q=1}^{m} (\alpha_k + \lambda_k \delta_q)^{-1} \gamma_{qj} \gamma_{lq} \\ &= \sum_{q=1}^{m-2} (\alpha_k + \lambda_k \delta_q)^{-1} \gamma_{qj} \gamma_{lq} + \alpha_k^{-1} (\gamma_{m-1,j} \gamma_{l,m-1} + \gamma_{m,j} \gamma_{l,m}). \end{aligned}$$

In this case, even though a computing package will return very small values for δ_m, and δ_{m-1}, they will not nullify the values $\{\gamma_{m-1,j}\gamma_{l,m-1}, \gamma_{m,j}\gamma_{l,m}\}$ because of the additive term α_k. Further, the values returned for $\{\gamma_{m-1,j}\gamma_{l,m-1}, \gamma_{m,j}\gamma_{l,m}\}$ in one calling routine will differ from those returned from another. Carrying this difference through to the calculation of $\mathbf{f}_k$, and all subsequent calculations, it is possible to get two different sets of results from different software packages, although both would be technically "correct."

To a lesser extent a related problem exists with the trace of $\mathbf{S}(\lambda_k)$:

$$\begin{aligned} tr[\mathbf{S}(\lambda_k)] &= \sum_{j=1}^{m} \frac{1}{\alpha_k + \lambda_k \delta_j} \\ &= \sum_{j=1}^{m-2} \frac{1}{\alpha_k + \lambda_k \delta_j} + \left(\frac{1}{\alpha_k + \lambda_k \delta_{m-1}} + \frac{1}{\alpha_k + \lambda_k \delta_m} \right). \end{aligned}$$

In the theoretical sense, the last term is exactly $2\alpha_k^{-1}$, but again, in the computational sense, δ_m and δ_{m-1} will not be exactly zero. Different routines will return different very small numbers that represent these. Therefore the trace of $\mathbf{S}(\lambda_k)$ can differ as well, and can be carried through to subsequent calculations resulting in differing results from two seemingly identical programs.

Thus, two software packages that return different representations of the zero eigenvalues, will return correspondingly different and nontrivial eigenvectors which can result in differing solutions to $\mathbf{S}(\lambda_k)$ that are both technically correct. Thankfully, as m increases, this effect and the differences between packages diminish.

3.3 M-step II: Some Specific ML Solutions

Next, solutions for the other model parameters are discussed, including model variance and the auto-regressive parameters for the factor time series. A special case of the auto-regressive factors is also considered, which will be useful for forecasting the call center data in Chapter 8.

3.3.1 Error Variances

Because the error variance involves cross-products of the factors, care must be taken in implementation of the EM. Differentiating the penalized log-likelihood (2.12) with respect to σ^2 yields $nm \cdot \hat{\sigma}^2 = \sum_{i,j}(x_{ij} - \sum_{k=1}^{K} \beta_{ik} f_{kj})^2$. Re-expressing this in vector notation, and expanding the terms results in

$$nm \cdot \hat{\sigma}^2 = \|\boldsymbol{X} - (\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta}\|^2 = \|\boldsymbol{X}\|^2 - 2\langle \boldsymbol{X}, (\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta}\rangle + \|(\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta}\|^2.$$

In the second term, $\boldsymbol{\beta}$ is replaced with $\mu_{\boldsymbol{\beta}|\mathbf{X}}$; for the third term:

$$\begin{aligned} \|(\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta}\|^2 &= \boldsymbol{\beta}'(\mathbf{F} \otimes \mathbf{I}_n)(\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta} \\ &= \boldsymbol{\beta}'\boldsymbol{\beta} \\ &= \sum_{k=1}^{K} \|\boldsymbol{\beta}_k\|^2, \end{aligned}$$

for which $\|\boldsymbol{\beta}_k\|^2$ is replaced with the sum of the diagonal elements of $E[\boldsymbol{\beta}_k \boldsymbol{\beta}_k' | \mathbf{X}]$.

A similar method is needed for the solution of the time series factor variances. Although the mathematical derivation of the coefficients for a squared order P polynomial is a straightforward one, in the present setting it is notationally cumbersome and does not add to the understanding of the issue at hand. Therefore for illustrative purposes it is sufficient to show the case for an AR(1) factor. Again, differentiating the penalized log-likelihood (2.12) with respect to σ_k^2 results in

$$(n-p) \cdot \hat{\sigma}_k^2 \quad = \quad \sum_{i=p+1}^{n} (\beta_{ik} - c_k - \sum_{r=1}^{p} \varphi_{rk}\beta_{i-r,k})^2.$$

For the specific AR(1) case this simplifies to

$$(n-1) \cdot \hat{\sigma}_k^2 \quad = \quad \sum_{i=2}^{n} (\beta_{ik} - c_k - \varphi_k \beta_{i-1,k})^2.$$

Expanding the quadratic expression within the sum:

$$(\beta_{ik} - c_k - \varphi_k\beta_{i-1,k})^2 = \beta_{ik}^2 + c_k^2 + \varphi_k^2\beta_{i-1,k}^2 - 2c\beta_{ik} - 2\varphi_k\beta_{ik}\beta_{i-1,k} + 2c\varphi_k\beta_{i-1,k}.$$

Therefore σ_k^2 is equivalently expressed as

$$(n-1) \cdot \hat{\sigma}_k^2 = \sum_{i=2}^{n} (\beta_{ik}^2 + c_k^2 + \varphi_k^2\beta_{i-1,k}^2 - 2c\beta_{ik} - 2\varphi_k\beta_{ik}\beta_{i-1,k} + 2c\varphi_k\beta_{i-1,k}).$$

To this, $E[\cdot|\mathbf{X}]$ is applied. Again, the appropriate replacements for the squared and cross product β terms are obtained from sums of diagonals and off-diagonals of a partitioned $E[\boldsymbol{\beta}_k\boldsymbol{\beta}_k'|\mathbf{X}]$ matrix. For example, $\sum_{i=2}^{n} \beta_{ik}\beta_{i-1,k}$ is obtained by summing the diagonal elements of the matrix created by rows $2:n$ and columns $1:(n-1)$ of $E[\boldsymbol{\beta}_k\boldsymbol{\beta}_k'|\mathbf{X}]$.

3.3.2 Multiple Intercepts for the Auto-regressive Processes

In some cases, it may be helpful to have additional flexibility in the FDFM specification. One method by which this is achieved is to add explanatory variables to the factor time series. As seen in Section 2.5.2, the use of general regressors can add some complexity to the estimation process. This section presents an intermediate approach that is less complicated than using the GLS estimation from Section 2.5.2 but still provides greater flexibility than no additional regressors in the factor time series. This is to use multiple intercepts as regressors. The method is illustrated through example, and will be revisited in Chapters 5 and 8. Due to the independence assumption it is adequate to show the derivation for a single factor $\boldsymbol{\beta}_k$. As such, the k subscript will temporarily be dropped for the purposes of illustration.

In the case of the call center data, the factors are assumed to follow an AR(1) process with intercept; however the constant assumes a different value for each of the five days in the business week. That is $A_{ik} \equiv A_i = [a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}]$ with $a_{id} = 1$ when date i corresponds to day d (d=1,...,5) and 0 otherwise. Therefore, the auto-regressive factor from the original model specification in Chapter refch:model simplifies to

$$\begin{aligned} \beta_i - A_i\mu &= \varphi(\beta_{i-1} - A_{i-1}\mu) + v_i \\ \beta_i &= (A_i - \varphi A_{i-1})\mu + \varphi\beta_{i-1} + v_i \\ &= c_{d_{i-1}} + \varphi\beta_{i-1} + v_i. \end{aligned}$$

Without loss of generality it can be assumed the data consist of complete business weeks so that the sequence of days follows the usual deterministic pattern; when missing data exists, those values can always be imputed to obtain full weeks. That is $d_i = 1 + (i-1) \mod 5$ so that there is a one to one relationship between $c_1, \ldots, c_5$ and $\mu_1, \ldots, \mu_5$ given by:

$$\begin{aligned} G(\varphi)\mu &= \mathbf{c} \\ \begin{bmatrix} -\varphi & 1 & 0 & 0 & 0 \\ 0 & -\varphi & 1 & 0 & 0 \\ 0 & 0 & -\varphi & 1 & 0 \\ 0 & 0 & 0 & -\varphi & 1 \\ 1 & 0 & 0 & 0 & -\varphi \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \end{bmatrix} &= \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} \end{aligned}$$

With this formulation, instead of having to invoke the more complicated Yule-Walker approach to solve for the μ and φ parameters, OLS can be used to solve for $\mathbf{c}$ and φ and then μ is found via the above relation. To do this, define an $(n-1) \times d = 5$ matrix $\mathbf{A}$ with elements $\mathbf{A}_{id} = 1$ when $d = 1 + (i-1) \mod 5$ and 0 otherwise and denote vectors like $[\beta_2, \ldots, \beta_n]'$ as $\{\beta\}_2^n$. Because cross products of the β_{ik} must be distinguished from scalars, the M-Step solution is best represented by the partitioned regression solution below. Reverting to the k subscript again to emphasize the solution/method is for each of the K factors:

$$\begin{bmatrix} \hat{\mathbf{c}}_k \\ \hat{\varphi}_k \end{bmatrix}_{(d+1)\times 1} = \begin{bmatrix} \mathbf{A}'\mathbf{A}_{d\times d} & \mathbf{A}'\{\boldsymbol{\beta}_k\}_1^{n-1} \\ (A'\{\boldsymbol{\beta}_k\}_1^{n-1})' & \sum_{i=1}^{n-1}\beta_{ik}^2 \end{bmatrix}^{-1}_{(d+1)\times(d+1)} \begin{bmatrix} [\mathbf{A}'\{\boldsymbol{\beta}_k\}_2^{n}]_{d\times 1} \\ \sum_{i=2}^{n}\beta_{i-1,k}\beta_{ik} \end{bmatrix} \tag{3.9}$$

Then as usual per the E-step:

1. β_{ik} is replaced by $E[\beta_{ik}|\mathbf{X}]$,

2. β_{ik}^2 is replaced by $E[\beta_{ik}^2|\mathbf{X}]$, and

3. $\beta_{ik}\beta_{i-1,k}$ is replaced by $E[\beta_{ik}\beta_{i-1,k}|\mathbf{X}]$.

Again, terms from the latter two of these come from the $E[\boldsymbol{\beta}\boldsymbol{\beta}|\mathbf{X}]$ matrix.

3.4 The One Factor AR(1) Model

In Section 2.5.1, a sequential EM approach for the FDFM was introduced where a K factor model could be estimated by sequentially estimating K single factor models. Because of this and because the one factor model arises frequently enough, an abridged treatment of the one factor model is provided below. The format is similar to that of Chapter 2. First the model is specified, followed by the joint distribution and likelihood. Then the steps of the EM are discussed, concluding with a description of the GCV selection for the smoothing parameter.

Referring to the original model Equation (2.4), the one factor model is just a special case of that:

$$\begin{cases} x_{ij} = \beta_i f_j + \epsilon_{ij}, \quad \epsilon_{ij} \overset{i.i.d.}{\sim} N(0,\sigma^2) \\ \beta_i - A_i\mu = \varphi(\beta_{i-1} - A_{i-1}\mu) + v_i, \quad v_i \overset{i.i.d.}{\sim} N(0,\sigma_1^2) \\ Ev_t\epsilon_{sj} = 0 \quad \text{for} \quad t,s = 1,\ldots,n; \quad j = 1,\ldots,m, \end{cases} \tag{3.10}$$

where for date i, the $1\times d$ regressor vector for the factor is A_i, with $d\times 1$ coefficient vector μ. As in the multi-factor case, an identification condition is required for the model. Since it is a single factor model, all that is necessary is that the factor loading curve is of unit length: $\mathbf{ff}' = 1$.

Using the prior notation introduced in Section 2.1, this model is denoted as an *FDFM(1,1)* model.

3.4.1 The Joint Distribution and Likelihood of $\mathbf{X}$ and $\boldsymbol{\beta}$

The one factor model is represented in matrix form by

$$\mathbf{X}_{n\times m} = \boldsymbol{\beta}_{n\times 1}\mathbf{f}_{1\times m} + \epsilon_{n\times m}, \tag{3.11}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_n]'$ and $\mathbf{f} = [f_1, \ldots, f_m]'$. As in Section 2.2, it is assumed for the moment that the factor time series $\boldsymbol{\beta}$ has no regressors for the sake of notational parsimony.

Then the joint distribution of $\mathbf{X}$ and $\boldsymbol{\beta}$ is given in the following proposition:

Proposition 3.4.1. *Joint Distribution of* $\mathbf{X}$ *and* $\boldsymbol{\beta}$

$$\begin{aligned} f(\mathbf{X}, \boldsymbol{\beta}) &= f(\boldsymbol{\beta}) \times \prod_{i=1}^{n}\prod_{j=1}^{m} f(x_{ij}|\beta_i) \\ &= f(\beta_1) \\ &\times \left\{\prod_{i=2}^{n} \frac{1}{\sqrt{2\pi\sigma_1^2}} exp\left[-\frac{(\beta_i - c - \varphi\beta_{i-1})^2}{2\sigma_1^2}\right]\right\} \\ &\times \prod_{i=1}^{n}\prod_{j=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[-\frac{(x_{ij} - \beta_i f_j)^2}{2\sigma^2}\right]. \end{aligned} \tag{3.12}$$

The penalized log-likelihood expression is then derived by taking the natural log of the joint distribution, multiplying by -2, and adding a single penalty term for the one factor loading curve:

$$\begin{aligned} PL &= -2\ln[f(\beta_1)] + (n-1)\ln(2\pi\sigma_1^2) + nm\ln(2\pi\sigma^2) \\ &+ \frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - \beta_i f_j)^2 + \sum_{i=2}^{n}\frac{1}{\sigma_1^2}(\beta_i - c - \varphi\beta_{i-1})^2 \\ &+ \lambda\mathbf{f}\Omega\mathbf{f}'. \end{aligned} \tag{3.13}$$

As in the multiple factor case, β_1 is assumed as given.

3.4.2 Maximum Likelihood Estimation with the EM

Maximum likelihood estimation using the EM proceeds as before with the specification of some initial values. From these, initial ML estimates are calculated which facilitate the E-step. In the M-step, updated parameter and factor loading curve estimates are computed. the process continues until estimates from one iteration to the next are sufficiently close.

Step 0: Preliminary Estimates via SVD

As in the multi-factor case, singular value decomposition provides initial values for the factors and factor loadings to begin the EM. Recall from Section 2.4.2 the SVD of $\mathbf{X}$:

$$\mathbf{X} \overset{SVD}{=} \mathbf{U}_{n\times m}\mathbf{D}_{m\times m}\mathbf{V}'_{m\times m}.$$

In this single factor case, $\mathbf{X}$ is simply approximated by:

$$\mathbf{X} \approx d_1\mathbf{u}_1\mathbf{v}'_1.$$

The initial values for the factors and factor loading curves are designated as $\boldsymbol{\beta}_{(0)} = d_1\mathbf{u}_1$ and $\mathbf{f}_{(0)} = \mathbf{v}'_1$. From these, initial parameter estimates are computed for σ^2 and the set of factor parameters $\{\sigma_1^2, c, \varphi\}$.

The solution for the error variance, σ^2, is as follows. The penalized log-likelihood (3.13) is differentiated with respect to σ^2. Setting the resulting expression equal to zero, and solving for σ^2 results in the following MLE for σ^2:

$$\hat{\sigma}^2 = \frac{\sum_{i,j}(x_{ij} - \beta_i f_j)^2}{nm}.$$

Recall the assumption that the first value of the factor time series is assumed as given. Then solutions for the AR(1) parameters reduce to an ordinary least squares problem. Holding the $\{\sigma_1^2\}$ fixed for the moment, maximization of (3.13) with respect to φ and c is equivalent to a minimization of a sum of squared errors. This result is summarized in a specialized version of Lemma 2.4.1 from Section 2.4.2:

Lemma 3.4.1. *Define $SSE = \sum_{i=2}^{n}(\beta_i - c - \varphi\beta_{i-1})^2$, and let*

$$\begin{aligned}
\mathbf{y} &\equiv [\beta_2, \ldots, \beta_n]', \\
\mathbf{W} &\equiv \begin{bmatrix} 1 & \beta_1 \\ 1 & \beta_2 \\ \vdots & \vdots \\ 1 & \beta_{n-1} \end{bmatrix}_{(n-1)\times 2}, \\
\textit{and } \phi &\equiv [c, \varphi]'.
\end{aligned}$$

Then $SSE = \|\mathbf{y} - \mathbf{W}\phi\|^2$ and the MLEs for the auto-regressive parameters are the OLS solutions $\hat{\phi} = [\mathbf{W}'\mathbf{W}]^{-1}\mathbf{W}'\mathbf{y}$.

With the estimates $\hat{\phi}$, the individual variances can be found by differentiating the penalized log-likelihood (3.13) with respect to σ_1^2, then setting the result equal to zero and finally solving for σ_1^2:

$$\hat{\sigma}_1^2 = \frac{\sum_{i=2}^{n}(\beta_i - c - \varphi\beta_{i-1})^2}{n-1}. \tag{3.14}$$

The E-Step

The model (3.10) can be rewritten in a vector-ized form similar to the form in Section 2.4.3:

$$\boldsymbol{X} = (\mathbf{f}' \otimes \mathbf{I}_n)\boldsymbol{\beta} + vec(\epsilon).$$

With $\boldsymbol{X} \equiv vec(\mathbf{X})$. Derivation of the conditional moments requires the expressions of the unconditional moments, $\{\mu_{\mathbf{X}}, \mu_{\boldsymbol{\beta}}, \Sigma_{\mathbf{X}}, \Sigma_{\boldsymbol{\beta}}, \Sigma_{\boldsymbol{\beta},\mathbf{X}}\}$.

First consider the mean and the variance matrix for the single factor time series $\boldsymbol{\beta}$. From Hamilton (1994):

$$E[\beta_i] = \frac{c}{1-\varphi},$$

and the i, lth entry of the covariance matrix $\Sigma_{\boldsymbol{\beta}}$ is

$$[\Sigma_{\boldsymbol{\beta}}]_{i,l} = \sigma_1^2 \left(\frac{\varphi^{|i-l|}}{1-\varphi^2} \right).$$

Both of these follow immediately from Lemma 2.4.3.

This yields the special cases for the single factor model based on Proposition 2.4.1 for the unconditional moments of $\boldsymbol{X}$ and $\boldsymbol{\beta}$:

$$\begin{aligned}
\mu_{\boldsymbol{\beta}} &= \frac{c}{1-\varphi} \otimes \mathbf{1}_n \\
\mu_{\mathbf{X}} &= (\mathbf{f}' \otimes \mathbf{I}_n)\mu_{\boldsymbol{\beta}} \\
\Sigma_{\mathbf{X}} &= (\mathbf{f}'\mathbf{f} \otimes \Sigma_{\beta} + \sigma^2 \mathbf{I}_{nm} \\
\Sigma_{\beta,\mathbf{X}} &= (\mathbf{f} \otimes \Sigma_{\beta}).
\end{aligned}$$

Using properties of multivariate normal random vectors, the conditional distribution of $\beta|\boldsymbol{X}$ is the same, notationally at least, as in Proposition 2.4.2:

$$\begin{aligned}
\mu_{\beta|\mathbf{X}} \equiv E[\boldsymbol{\beta}|\boldsymbol{X}] &= \mu_{\beta} + \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}(\mathbf{X} - \mu_{\mathbf{X}}), \\
\Sigma_{\beta|\mathbf{X}} \equiv Var[\boldsymbol{\beta}|\boldsymbol{X}] &= \Sigma_{\beta} - \Sigma_{\beta,\mathbf{X}}\Sigma_{\mathbf{X}}^{-1}\Sigma_{\mathbf{X},\beta}.
\end{aligned}$$

and

$$E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}] = \Sigma_{\beta|\mathbf{X}} + \mu_{\beta|\mathbf{X}}\mu'_{\beta|\mathbf{X}}.$$

The M-Step

Differentiating the penalized log-likelihood (3.13) with respect to $\mathbf{f}$, setting the resulting expression equal to zero and solving for $\mathbf{f}$ yields:

$$\hat{\mathbf{f}}' = \left[||\boldsymbol{\beta}||^2 \mathbf{I}_m + \lambda\sigma^2\Omega \right]^{-1} \left[\sum_{i=1}^{n} \beta_i X_i \right]. \tag{3.15}$$

Because there is only the single factor loading curve, there is no need for any sort of sequential solution as in Section 2.4.4. This further simplifies the GCV selection for the single smoothing parameter λ.

GCV Selection

GCV selection for the single smoothing parameter λ is based on finding a value of the parameter that minimizes the GCV criterion. Again, the similarities of the solution for $\mathbf{f}$ in (3.15) to the solution for a ridge regression are apparent. The GCV criterion is

$$\text{GCV}(\lambda) \quad = \quad \frac{\|(\mathbf{I}_{nm} - H_\lambda)\tilde{X}\|^2/nm}{[1 - \text{tr}(H_\lambda)/nm]^2},$$

with

$$H_\lambda \quad = \quad \left\{ \frac{1}{\sigma^2}(\boldsymbol{\beta} \otimes \mathbf{I}_m) \left[\frac{\|\boldsymbol{\beta}\|^2}{\sigma^2}\mathbf{I}_m + \lambda\Omega \right]^{-1} (\boldsymbol{\beta}' \otimes \mathbf{I}_m) \right\}.$$

GCV(λ) is calculated over a grid of possible values during the estimation of each factor loading curve. The smoothing parameter that corresponds to the minimum value of GCV($\cdot$) is selected as the optimal one.

After the $\mathbf{f}/\lambda$ pair is determined, orthogonal-ization in this case is equivalent to mere normalization. Recall this is required in order to maintain the assumption of orthogonality; that $\mathbf{f}_{(l)}\mathbf{f}'_{(l)} = 1$, for each iteration l of the EM. This is done by:

- Updating the factor by the multiple of its norm-ed factor loading: $\boldsymbol{\beta} \times \|\mathbf{f}\|$.
- Updating factor cross products by the squared norm of the factor loading: $(\boldsymbol{\beta}\boldsymbol{\beta}') \times \|\mathbf{f}\|^2$.
- Finally, dividing the factor loading curve by its norm: $\mathbf{f}/\|\mathbf{f}\|$.

With the updated orthogonal factor and factor loadings, the M-step concludes with the calculation of the overall model variance (σ^2), and the auto-regressive factor parameters; see Section 2.4.5 for details on replacing $\boldsymbol{\beta}$ with the corresponding conditional expectations for these.

Finally, after these parameter estimates are updated, the EM continues on to the next E-step where the factor time series are updated and fed into another M-step. The process continues until the factor loading curve and other parameters from one EM iteration to the next are sufficiently close.

This concludes Chapter 3. The remaining chapters of this dissertation cover some more theoretical derivations, simulation results, selection methods and the FDFM applied to real data; finally concluding remarks including possible future research directions.

Chapter 4

GCV and NCS Derivations

4.1 Review: The Functional Dynamic Factor Model

Abstracting for a moment from the present setting of yield curve forecasting, consider the more general process of a time series of curves $\{x_i(t) : t \in \mathcal{T}; i = 1, \ldots, n\}$, where $\mathcal{T}$ is some continuous interval and i indexes discrete time. Following the classical FDA development as discussed in Ramsay and Silverman (2005), it is hypothesized that each curve is composed of a forecastable smooth underlying curve, $y_i(t)$, plus an error component, $\epsilon_i(t)$, that is,

$$x_i(t) = y_i(t) + \epsilon_i(t). \tag{4.1}$$

There are two primary goals of a functional time series model: provide an accurate description of the dynamics of the series; accurately forecast the smooth curve $y_{n+h}(t)$ for some forecast horizon $h > 0$.

In practice, of course, only a discrete sampling of each curve is observed. Specifically, for $t \in \mathcal{T}$, consider a sample of discrete points $\{t_1, t_2, \ldots, t_m\}$ with $t_j \in \mathcal{T}$ for $j \in \{1, \ldots, m\}$. Then denote

$$x_{ij} \equiv x_i(t_j),$$

as an observed data at time i evaluated at t_j. That is, the observed data x_{ij} is a point sampled from the process $x_i(t)$ at the specific value $t = t_j$.

In terms of forecasting, if the cross-sectional dimension m is small enough, the observed data $\{x_{ij}\}$ can be modeled directly as a multivariate time series using, say, a VAR specification. Otherwise, DFMs can be used to reduce dimension. Here, coefficients known as factor loadings with forecasted factor scores will provide discrete forecasts at t_j for the functional time series. Yet there is nothing inherent or implicit in the DFM framework to provide direction in terms of forecasting an entire function for all $t \in \mathcal{T}$. Thus, to forecast $y_{n+h}(t)$, we can synthesize the DFM framework with methods from functional data analysis (FDA).

4.1.1 The Model

Via this synthesis, we propose a model referred to as the functional dynamic factor model (FDFM). The formulation is similar to that of a DFM where the observed data $\{x_{ij}\}$ is a function of a small set of K latent dynamic factors $\{\beta_{ik}; k = 1, \ldots, K\}$ and their corresponding factor loadings. But in this setting the factor loadings $f_{kj} \equiv f_k(t_j)$ are discrete samples from continuous, unobserved though non-random factor loading curves $f_k(\cdot)$. Together, the dynamic factors with their functional coefficients generate the forecastable part of the time series of curves $\{x_i(t)\}$.

In theory, the dynamic factors can follow any type of time series process such as (V)ARIMA, but for the purpose of this paper we focus on factors with stationary and independent AR(p) errors. These factors can include explanatory variables[1] or just a constant. In the former case, we have a $1 \times d$ regressor vector A_{ik} having the $d \times 1$ coefficient vector μ_k. We need not assume these nor the number of them are the same for each factor; nor the order of the AR process p, for that matter. Rather, for notational convenience we simply define $p = \max\{p_1, \ldots, p_K\}$ and $d = \max\{d_1, \ldots, d_K\}$ and for any case where $p_k < p$ or $d_k < d$ we use the appropriate placement of zeros. We retain the option for the regressors themselves to differ; thus we continue to use the k subscript per factor. Finally, for the model to be identified, we require that the functional

[1] These could be economic indicators, or seasonal effects, e.g.

coefficients are orthonormal[2]. The model is explicitly stated as

$$\begin{cases} x_i(t_j) = \sum_{k=1}^{K} \beta_{ik} f_k(t_j) + \epsilon_i(t_j), \\ \beta_{ik} - A_{ik}\mu_k = \sum_{r=1}^{p} \varphi_{rk}(\beta_{i-r,k} - A_{i-r,k}\mu_k) + v_{ik}, \\ \int_T f_k(t) f_l(t) dt = \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{otherwise,} \end{cases} \end{cases} \tag{4.2}$$

with $\epsilon_i(t_j) \equiv \epsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma^2)$, $v_{ik} \overset{i.i.d.}{\sim} N(0, \sigma_k^2)$ and $E[v_{ik}\epsilon_{i'j}] = 0$ for $i, i' = 1, \ldots, n$. Should we require only a constant in place of regressors, then $A_{ik}\mu_k$ is a scalar μ_k for all i. With the assumption of stationarity this yields the constant $c_k = \mu_k(1 - \sum_{r=1}^{p} \varphi_{rk})$. This is a broad framework that includes the standard versions of both DFMs and FPCA models: when the coefficients $\{f_k(t)\}$ are non-functional, Model (4.2) reduces to the standard DFM; when the factors $\{\boldsymbol{\beta}_k\}$ are non dynamic, the model is similar to FPCA.

4.1.2 Estimation

With the error assumptions for Model (4.2) we propose estimation via maximum likelihood (ML). To ensure smooth and functional estimates for the factor loading curves, we augment the likelihood expression with "roughness" penalties (Green and Silverman, 1994) and maximize a *penalized* likelihood expression. Because our dynamic factors are unobserved, we consider this a problem of missing data, and use the expectation maximization (EM) algorithm (Dempster et al., 1977) to estimate model parameters and smooth curves.

Penalized Likelihood

Let the $n \times m$ matrix $\mathbf{X}$ denote collectively the observed data where the (i, j)th element of $\mathbf{X}$ is x_{ij} for $i = 1, \ldots, n$, $j = 1, \ldots, m$. The rows of $\mathbf{X}$ correspond to yield curves for a fixed date; the columns are the time series of yield for a specific maturity. Next, we denote $f_{kj} = f_k(t_j)$,

[2] Other types of constraints may be employed to ensure identification, such as conditions on the covariance function of the factor loading curves.

the *row* vector $\mathbf{f}_k = [f_{k1}, \ldots, f_{km}]$, and

$$\mathbf{F}' = \left[\mathbf{f}_1', \ldots, \mathbf{f}_K'\right].$$

In a similar manner, we define $\boldsymbol{\beta}_k = [\beta_{1k} \ldots \beta_{nk}]'$ and the matrix $\mathbf{B}_{n \times K} = [\boldsymbol{\beta}_1 \ldots \boldsymbol{\beta}_K]$. Thus the columns of $\mathbf{B}$ are the time series factors $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$. Then, the Model (4.2) is represented in matrix form as

$$\mathbf{X}_{n \times m} = \mathbf{B}_{n \times K}\mathbf{F}_{K \times m} + \boldsymbol{\epsilon}_{n \times m} = \sum_{k=1}^{K} \boldsymbol{\beta}_k \mathbf{f}_k + \boldsymbol{\epsilon}, \tag{4.3}$$

where $\boldsymbol{\epsilon} = [\epsilon_{ij}]_{n \times m}$ with $\epsilon_{ij} = \epsilon_i(t_j)$.

To derive the log-likelihood expression, we rely on successive conditioning of the joint distribution for $\mathbf{X}$ and $\mathbf{B}$:

$$l(\mathbf{X}, \mathbf{B}) = l(\mathbf{B}) + l(\mathbf{X}|\mathbf{B}). \tag{4.4}$$

Because we have assumed that the K factors of AR(p) series are independent, their joint distribution is the product of the individual distributions. To each of those, we further condition on the first p values of each factor time series; thus our likelihood (4.4) is a *conditional* one. For ease of notation we assume there are no regressors in the factor time series. Then

$$l(\mathbf{B}) = (n-p)\sum_{k=1}^{K} \ln(2\pi\sigma_k^2) + \sum_{i=p+1}^{n}\sum_{k=1}^{K} \frac{1}{\sigma_k^2}(\beta_{ik} - c_k - \sum_{r=1}^{p} \varphi_{rk}\beta_{i-r,k})^2, \tag{4.5}$$

and

$$l(\mathbf{X}|\mathbf{B}) = nm\ln(2\pi\sigma^2) + \frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - \sum_{k=1}^{K}\beta_{ik}f_{kj})^2. \tag{4.6}$$

To ensure the underlying factor loading curve $f_k(\cdot)$ is smooth, following Green and Silverman (1994), we include roughness penalties to (4.6) in solving for the K factor loading curves $\mathbf{f}_k$. These terms place a condition on the second derivative of each function $f_k(\cdot)$ over its domain $\mathcal{T}$ to ensure that the function is not too "rough." Combining Equation (4.4) with the roughness penalties, we obtain the following penalized log-likelihood:

$$\begin{aligned} l_p(\mathbf{X}, \mathbf{B}) &= l(\mathbf{B}) + l_p(\mathbf{X}|\mathbf{B}), \\ &\equiv l(\mathbf{B}) + \left[l(\mathbf{X}|\mathbf{B}) + \sum_{k=1}^{K} \lambda_k \int \left[f_k''(t) \right]^2 dt \right]. \end{aligned} \tag{4.7}$$

The penalty parameter λ_k controls how strictly the roughness penalty is enforced, and we allow it to differ for each loading curve (thus the "k" subscript). The selection process for the penalty parameters is discussed in Section 4.1.2. We refer to the latter term in Equation (4.7), $l_p(\mathbf{X}|\mathbf{B})$, as the penalized sum of squares (PSS). Intuitively, optimization of PSS balances a familiar goodness-of-fit criterion with a smoothness requirement for the resulting estimates of $f_k(t)$.

Thus, to estimate the model, we will optimize the penalized, conditional log-likelihood $l_p(\mathbf{X}, \mathbf{B})$ with respect to the set of parameters and factor loading curves: $\Theta \equiv \sigma^2 \cup \{\mathbf{f}_k, \sigma_k^2, c_k, \varphi_{1,k}, \ldots, \varphi_{p,k}\}_{k=1}^{K}$. Note that the technical difficulty is that the dynamic factors $\mathbf{B}$ are unobserved, which makes it infeasible to directly optimize the penalized likelihood (4.7). Our proposal is to treat their absence as a missing value problem and use the expectation maximization algorithm (Dempster et al., 1977) to optimize Expression (4.7); this will be discussed in Section 4.1.2.

Below we assume the dynamic factors are known and discuss how to estimate the AR model parameters and the smooth factor loading curves.

When the dynamic factors have no regressors the conditional MLEs for the AR parameters ($\{\sigma_k^2, c_k, \varphi_{1,k}, \ldots, \varphi_{p,k}\}$) are the same as the ordinary least squares (OLS) solutions. In the case where the factors do have regressors, an additional step is required to alternatively solve for the AR parameters $\{\varphi_{1,k}, \ldots, \varphi_{p,k}\}$ and the regressor coefficient vectors $\{\mu_k\}$. The resulting solutions are the (feasible) generalized least squares (GLS) solution; see Judge (1985) for a detailed discussion. We do consider this general formulation in the simulation studies reported in 5

Now we discuss how to estimate the loading curves $f_k(t)$. In order to allow the curves to have their own smoothness, through allowing different λ_k, we proceed in a sequential manner to estimate $f_k(t)$ one at a time, incorporating penalty parameter selection for that loading curve through cross-validation, as discussed in Section 4.1.2.

According to Theorem 2.1 of Green and Silverman (1994), for fixed k, the minimizer $\hat{f}_k(\cdot)$ of PSS is a natural cubic spline with knot locations $t_1, \ldots, t_m$. Further, this NCS interpolates the discrete row vector $\hat{\mathbf{f}}_k$ which is the solution to the minimization problem

$$\min_{\mathbf{f}_k} \left[l(\mathbf{X}|\mathbf{B}) + \lambda_k \mathbf{f}_k \boldsymbol{\Omega} \mathbf{f}_k' \right], \tag{4.8}$$

where $\boldsymbol{\Omega_{m \times m}}$ is a matrix determined solely by the spline knot locations; the explicit formulation of $\boldsymbol{\Omega}$ is deferred until Section 4.3.1.

Let $\boldsymbol{X}_T \equiv vec(\mathbf{X}')$ which stacks the columns of $\mathbf{X}'$ into an $nm \times 1$ vector. Then using the Kronecker product $\otimes$, Model (4.3) can be rewritten in vector form as

$$\boldsymbol{X}_T = (\mathbf{B} \otimes \mathbf{I}_m) vec(\mathbf{F}') + vec(\boldsymbol{\epsilon}') = \sum_{k=1}^{K} (\boldsymbol{\beta}_k \otimes \mathbf{I}_m) \mathbf{f}_k' + vec(\boldsymbol{\epsilon}'). \tag{4.9}$$

Consider the solution $\hat{\mathbf{f}}_k$ for fixed $k \in \{1, \ldots, K\} \equiv \mathbb{K}$. For the remaining $h \in \mathbb{K}$, we define $\boldsymbol{X}_T^* = \boldsymbol{X}_T - \sum_{h \neq k} (\boldsymbol{\beta}_h \otimes \mathbf{I}_m) \mathbf{f}_h'$. Then the minimization problem (4.8) is equivalent to

$$\min_{\mathbf{f}_k} \left\| \frac{1}{\sigma} \boldsymbol{X}_T^* - \frac{1}{\sigma} (\boldsymbol{\beta}_k \otimes \mathbf{I}_m) \cdot \mathbf{f}_k' \right\|^2 + \lambda_k \mathbf{f}_k \boldsymbol{\Omega} \mathbf{f}_k', \tag{4.10}$$

where $\| \cdot \|$ is the Euclidean norm.

Although not immediately apparent, minimization of (4.10) is equivalent to a ridge regression problem. Let $\mathbf{Y} = \frac{1}{\sigma} \boldsymbol{X}_T^*$, and $\mathbf{W} = \frac{1}{\sigma} (\boldsymbol{\beta}_k \otimes \mathbf{I}_m)$. Then $\min_{\mathbf{f}_k} \left[\|\mathbf{Y} - \mathbf{W} \mathbf{f}_k'\|^2 + \lambda_k \mathbf{f}_k \boldsymbol{\Omega} \mathbf{f}_k' \right]$ has the solution:

$$\hat{\mathbf{f}}_k' = \left[\mathbf{W}' \mathbf{W} + \lambda_k \boldsymbol{\Omega} \right]^{-1} \mathbf{W}' \mathbf{Y}. \tag{4.11}$$

Further, the ridge regression formulation suggests a generalized cross validation (GCV) procedure for the selection of each λ_k; this is covered with more detail in Section 4.1.2.

EM Algorithm

First introduced by Dempster et al. (1977), then refined by Meng and Rubin (1993), the EM is an iterative method by which to impute missing data with values based on conditional expectations using the observed data. First, the EM is inaugurated with initial values for the factors and factor loading curves. From these initial values, maximum likelihood estimates for the remaining parameters from Θ are calculated based on Equations (4.5), (4.6) and (4.7); we call this *Step 0.* Then the algorithm alternates between the E-step and the M-step. In the *E-step*, values for the factor time series are calculated as conditional expectations given the observed data and current values for the MLEs. In the *M-step*, MLEs are calculated for the factor loading curves and other parameters based on the factor scores from the conditional expectations in the E-step. After the initial step, the E-step and the M-step are repeated until differences in the estimates from one iteration to the next are sufficiently small. More details are given below.

Step 0: Akin to the method used in Shen (2009), initial values for $\mathbf{B}$ are composed of the first K singular values and left singular vectors from the singular value decomposition (SVD) of the data matrix $\mathbf{X}$. Initial values for $\mathbf{F}$ are the corresponding right singular vectors. From these, initial parameter estimates are computed for σ^2 and the set of factor parameters $\{\sigma_k^2, c_k, \varphi_{1,k}, \ldots, \varphi_{p,k}\}$.

The E-Step: Derivation of the conditional moments for the E-Step requires the expressions of some of the unconditional moments. Let $\boldsymbol{X} \equiv vec(\mathbf{X})$ and $\boldsymbol{\beta} \equiv vec(\mathbf{B})$. Then Equation (4.3) can be rewritten as

$$\boldsymbol{X} = (\mathbf{F}' \otimes \mathbf{I}_n)\boldsymbol{\beta} + vec(\boldsymbol{\epsilon}) = \sum_{k=1}^{K} (\mathbf{f}_k \otimes \mathbf{I}_n)\boldsymbol{\beta}_k + vec(\boldsymbol{\epsilon}). \tag{4.12}$$

Define the $n \times n$ variance matrix for $\boldsymbol{\beta}_k$ as $\boldsymbol{\Sigma}_k$, and let $\mathbf{c}$ be the $K \times 1$ vector with elements

$c_k/[1-(\sum_{r=1}^{p}\varphi_{r,k})]$. Then

$$\begin{aligned} E[\boldsymbol{\beta}] \equiv \boldsymbol{\mu}_{\boldsymbol{\beta}} = \mathbf{c}\otimes\mathbf{1}_n \quad E[\boldsymbol{X}] \equiv \boldsymbol{\mu}_{\mathbf{X}} = (\mathbf{F}'\otimes\mathbf{I}_n)\boldsymbol{\mu}_{\boldsymbol{\beta}} \\ Var[\boldsymbol{\beta}] \equiv \boldsymbol{\Sigma}_{\boldsymbol{\beta}} = diag\{\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_K\} \quad Cov[\boldsymbol{\beta},\boldsymbol{X}] \equiv \boldsymbol{\Sigma}_{\boldsymbol{\beta},\mathbf{X}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}}(\mathbf{F}\otimes\mathbf{I}_n) \\ Var[\boldsymbol{X}] \equiv \boldsymbol{\Sigma}_{\mathbf{X}} = (\mathbf{F}'\otimes\mathbf{I}_n)\boldsymbol{\Sigma}_{\boldsymbol{\beta}}(\mathbf{F}\otimes\mathbf{I}_n) + \sigma^2\mathbf{I}_{nm}. \end{aligned} \tag{4.13}$$

Next, using properties of multivariate normal random vectors, the conditional distribution of $\boldsymbol{\beta}|\boldsymbol{X}$ can be found. Let

$$\begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{X} \end{pmatrix} \sim N\left[\begin{pmatrix} \boldsymbol{\mu}_{\boldsymbol{\beta}} \\ \boldsymbol{\mu}_{\mathbf{X}} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{\boldsymbol{\beta}} & \boldsymbol{\Sigma}_{\boldsymbol{\beta},\mathbf{X}} \\ \boldsymbol{\Sigma}_{\mathbf{X},\boldsymbol{\beta}} & \boldsymbol{\Sigma}_{\mathbf{X}} \end{pmatrix}\right].$$

Then

$$\begin{cases} \boldsymbol{\mu}_{\boldsymbol{\beta}|\mathbf{X}} \equiv E[\boldsymbol{\beta}|\boldsymbol{X}] = \boldsymbol{\mu}_{\boldsymbol{\beta}} + \boldsymbol{\Sigma}_{\boldsymbol{\beta},\mathbf{X}}\boldsymbol{\Sigma}_{\mathbf{X}}^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_{\mathbf{X}}), \\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}|\mathbf{X}} \equiv Var[\boldsymbol{\beta}|\boldsymbol{X}] = \boldsymbol{\Sigma}_{\boldsymbol{\beta}} - \boldsymbol{\Sigma}_{\boldsymbol{\beta},\mathbf{X}}\boldsymbol{\Sigma}_{\mathbf{X}}^{-1}\boldsymbol{\Sigma}_{\mathbf{X},\boldsymbol{\beta}}, \\ E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}] = \boldsymbol{\Sigma}_{\boldsymbol{\beta}|\mathbf{X}} + \boldsymbol{\mu}_{\boldsymbol{\beta}|\mathbf{X}}\boldsymbol{\mu}_{\boldsymbol{\beta}|\mathbf{X}}'. \end{cases} \tag{4.14}$$

From a computational standpoint there is concern over the inversion of $\boldsymbol{\Sigma}_{\mathbf{X}}$ which is of order nm. Because the EM is an iterative procedure, this could be especially problematic. However, we can use the following result based on the Sherman–Morrison–Woodbury factorization (Press et al., 1992, e.g.) to simplify the computation:

Proposition 4.1.1.

$$\boldsymbol{\Sigma}_{\mathbf{X}}^{-1} = \sigma^{-2}\mathbf{I}_{nm} - \sigma^{-4}(\mathbf{F}'\otimes\mathbf{I}_n)\left[\sigma^{-2}\mathbf{I}_{nK} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\right]^{-1}(\mathbf{F}\otimes\mathbf{I}_n). \tag{4.15}$$

The form itself is not so important as what it means. Instead of inverting $\boldsymbol{\Sigma}_{\mathbf{X}}$ directly, which is an $nm \times nm$ matrix, only the middle matrix $\left[\sigma^{-2}\mathbf{I}_{nK} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\right]$ needs to be inverted. This matrix is of smaller size $nK \times nK$. Further, as $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$ is block diagonal, then $\sigma^{-2}\mathbf{I}_{nK} + \boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}$ is as well. Thus, using this factorization, the inversion of $\boldsymbol{\Sigma}_{\mathbf{X}}$ is reduced from an $nm \times nm$

inversion to K, $n \times n$ inversions.

With the conditional moments, the E-step of the EM posits that the missing data (the time series factors) are replaced with the known values of the conditional distribution given $\boldsymbol{X}$. Thus in the following M-step, in solving for the MLEs, expressions involving $\boldsymbol{\beta}_k$ will utilize values from $\boldsymbol{\mu}_{\boldsymbol{\beta}|\mathbf{X}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\beta}|\mathbf{X}}$, and $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}]$.

The M-Step: For each EM iteration, the M-step optimizes the conditional penalized log-likelihood in Equation (4.7) given the observed data and the current parameter estimates for Θ. It is clear from Equations (4.5) and (4.6) that in the MLEs, the factor time series appear either singly or in terms of cross products both within and between factors. Values for terms like β_{ik} come directly from the vector $\boldsymbol{\mu}_{\boldsymbol{\beta}|\mathbf{X}}$. But because a term like $\beta_{ik'}\beta_{hk}$, $k, k' = 1, \ldots, K$, $i, h = 1, \ldots, n$, is a conditional expectation of a product, its replacement values are obtained from the $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}]$ matrix. We will show in Section 4.1.2 some rather fortunate results to simplify computation of the conditional expectation of the factor products.

The M-step, then, is just a matter of making these substitutions into the likelihood, and solving for the MLEs. After the M-Step, we return to the E-Step to update the values for the factor time series. This procedure is repeated until the parameter estimates from one iteration of the EM are sufficiently close to those of the next.

Computational Efficiency

This section presents results intended to ease some of the computational aspects of the estimation for the functional dynamic factor model. The reason for this being that the EM algorithm is an iterative procedure; and each iteration is rife with large matrix inversions and manipulations. Further, given the results of Section 4.1.2, we propose to sequentially solve for each factor loading curve $\mathbf{f}_k$; $k = 1, \ldots, K$. Finally, the smoothing parameter λ_k needs to be selected in a data-adaptive manner for each k. Below, we derive a (generalized) cross-validation (GCV) procedure to achieve this. Efficient implementation allows us to easily evaluate the GCV score over many candidate values of λ_k.

GCV Selection: More specifically, using the notation from Section 4.1.2, the ridge regression formulation (4.10) with solution (4.11) suggests a generalized cross validation (GCV) criterion for selecting the λ_k which minimizes

$$\begin{aligned} \mathrm{GCV}(\lambda_k) &= \frac{\|(\mathbf{I}_{nm} - \mathbf{H}_{\lambda_k})\mathbf{Y}\|^2/nm}{[1 - \mathrm{tr}(\mathbf{H}_{\lambda_k})/nm]^2} \text{ for }, \\ \mathbf{H}_{\lambda_k} &= \mathbf{W}\left[\mathbf{W}'\mathbf{W} + \lambda_k \mathbf{\Omega}\right]^{-1}\mathbf{W}'. \end{aligned} \tag{4.16}$$

$\mathrm{GCV}(\lambda_k)$ is calculated over a grid of possible values during the M-Step of each EM iteration for each factor loading curve. The smoothing parameter that corresponds to the least value of $\mathrm{GCV}(\cdot)$ is selected as the optimal one. It is worthwhile to note that this can be a computationally intensive procedure: calculating $\mathrm{GCV}(\lambda)$ for several values for λ during each EM iteration and for each factor. It can be shown that $\mathbf{H}_{\lambda_k}$ depends on the inversion of the matrix $\left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k \mathbf{\Omega}\right]$. Using the eigen-decomposition of $\mathbf{\Omega}$, a method exists for which the only inversion required is the inversion of a diagonal matrix. Consider the following proposition:

Proposition 4.1.2. *Let* $\mathbf{S}(\lambda_k) = \left[\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\mathbf{I}_m + \lambda_k \mathbf{\Omega}\right]^{-1}$ *so that* $\hat{\mathbf{f}}_k' = [\mathbf{S}(\lambda_k)]\left[\frac{1}{\sigma^2}(\boldsymbol{\beta}_k' \otimes \mathbf{I}_m)\boldsymbol{X}_T^*\right]$. *Given the eigen-decomposition of the* $m \times m$ *penalty matrix* $\mathbf{\Omega} = \mathbf{\Gamma}\mathbf{\Delta}\mathbf{\Gamma}'$ *with* $\mathbf{\Delta}_{m\times m} = diag\{\delta_j\}_{j=1}^m$*, then*

$$\mathbf{S}(\lambda_k) = \mathbf{\Gamma} \cdot diag\left\{\left(\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2} + \lambda_k\delta_j\right)^{-1}\right\}\mathbf{\Gamma}',$$

and

$$tr\{\mathbf{H}_{\lambda_k}\} = \frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2}\sum_{j=1}^{m}\frac{1}{\frac{\|\boldsymbol{\beta}_k\|^2}{\sigma^2} + \lambda_k\delta_j}.$$

Thus a single eigen-decomposition, followed by a diagonal matrix inversion for each of the factors, circumvents performing an $m \times m$ inversion for each of the K factors and each of the candidate values for λ_k.

Block Diagonality: In the M-step, when products of the factors appear, such as $\langle\boldsymbol{\beta}_k, \boldsymbol{\beta}_h\rangle = E[\langle\boldsymbol{\beta}_k, \boldsymbol{\beta}_h\rangle|\boldsymbol{X}]$, then the imputation comes from the $E[\boldsymbol{\beta}\boldsymbol{\beta}'|\boldsymbol{X}]$ matrix. It can be shown that $\mathbf{\Sigma}_{\boldsymbol{\beta}|\mathbf{X}}$ is block diagonal; this property facilitates a rather convenient result regarding between-factor cross products:

Proposition 4.1.3. $\mathbf{\Sigma}_{\boldsymbol{\beta}|\mathbf{X}}$ *is block diagonal with* K $n \times n$ *blocks. Further, for* $h \neq k$, $E[\langle \boldsymbol{\beta}_k, \boldsymbol{\beta}_h \rangle | \boldsymbol{X}] = \langle \boldsymbol{\mu}_{\boldsymbol{\beta}_k|\boldsymbol{X}}, \boldsymbol{\mu}_{\boldsymbol{\beta}_h|\boldsymbol{X}} \rangle$.

Therefore, the conditional expectation of a product of two (distinct) factors is simply the product of their individual expectations. This greatly simplifies the M-step calculations.

4.2 Cross Validation

In the Review Section 4.1.2 the method and expression for the determination of the optimal smoothing parameter λ_k is summarized based on Equations (4.10) and (4.11). There is an equivalent and more intuitive formulation that better facilitates the derivation. Further, without loss of generality, because the GCV method is sequential for each k, we can focus the derivation on a one factor model and drop the k subscript. Finally, we may also assume $\sigma^2 = 1$ to further ease the notation.

First this equivalence is shown. Next based on this equivalence, the result is formally derived. Finally, a serendipitous result of the derivation is a simplifed expression for GCV criterion (4.16). In any case, Proposition 4.1.2 continues to hold.

4.2.1 Alternate Formulation

Derivation of the CV and GCV criteria is based on calculating the leave-out residual from time point/column-at-a-time deletion of the observed data matrix $\mathbf{X}$. Although a popular method for GCV in FDA is row/curve deletion, because the present setting involves a dynamic system of curves, deletion of a curve removes an entire time point from the data and destroys the time dependency structure. Presentation of the ridge regression formulation in Equations (4.10) and (4.11) was originally formulated as such to heuristically justify the GCV criterion based on ridge regression: a vector of "dependent variables" ($\boldsymbol{X}_T^*$), a matrix of "explanatory variables" ($\boldsymbol{\beta}_k \otimes \mathbf{I}_m$), and a vector of "coefficients" ($\mathbf{f}_k$).

But time point-deletion in this setting requires deleting every mth component of $\boldsymbol{X}_T^*$, rather then deleting consecutive $n \times 1$ blocks of $\boldsymbol{X}^*$. This makes the derivation awkward and less intuitive than the method presented in Huang et al. (2008). It is thus preferred to follow that approach. To do this, in this section we denote $\mathbf{f}$ as an $m \times 1$ column vector.

For each k, the GCV method in Section 4.1 is based on holding the remaining $k \neq K$ factor loading curves fixed: $\boldsymbol{X}_T^* = \boldsymbol{X}_T - \sum_{h \neq k} (\boldsymbol{\beta}_h \otimes \mathbf{I}_m)\mathbf{f}_h'$. Thus here we can focus on the one factor model in order to drop the k subscript. To further ease notation, we will assume $\sigma^2 = 1$.

Proposition 4.2.1. *The following minimization problems with respective GCV criteria are equivalent:*

$$\min_{\mathbf{f}} \|\boldsymbol{X}_T - (\boldsymbol{\beta} \otimes \mathbf{I}_m) \cdot \mathbf{f}\|^2 + \lambda \mathbf{f}' \boldsymbol{\Omega} \mathbf{f},$$
$$\min_{\mathbf{f}} \|\boldsymbol{X} - (\mathbf{I}_m \otimes \boldsymbol{\beta}) \cdot \mathbf{f}\|^2 + \lambda \mathbf{f}' \boldsymbol{\Omega} \mathbf{f}.$$

Proof:

From Equation (4.11), the solution to the first problem is

$$\hat{\mathbf{f}} = \left[(\boldsymbol{\beta} \otimes \mathbf{I}_m)'(\boldsymbol{\beta} \otimes \mathbf{I}_m) + \lambda \boldsymbol{\Omega}\right]^{-1} (\boldsymbol{\beta} \otimes \mathbf{I}_m)' \text{vec}(\mathbf{X}') = \left[\|\boldsymbol{\beta}\|^2 + \lambda \boldsymbol{\Omega}\right]^{-1} (\boldsymbol{\beta}' \otimes \mathbf{I}_m) \text{vec}(\mathbf{X}').$$

The solution to the second problem is

$$\hat{\mathbf{f}} = \left[(\mathbf{I}_m \otimes \boldsymbol{\beta})'(\mathbf{I}_m \otimes \boldsymbol{\beta}) + \lambda \boldsymbol{\Omega}\right]^{-1} (\mathbf{I}_m \otimes \boldsymbol{\beta})' \text{vec}(\mathbf{X}) = \left[\|\boldsymbol{\beta}\|^2 + \lambda \boldsymbol{\Omega}\right]^{-1} (\mathbf{I}_m \otimes \boldsymbol{\beta})' \text{vec}(\mathbf{X}).$$

Clearly the inverted matrices in either solution are identical. To show $(\boldsymbol{\beta}' \otimes \mathbf{I}_m)\text{vec}(\mathbf{X}') = (\mathbf{I}_m \otimes \boldsymbol{\beta})'\text{vec}(\mathbf{X})$, we use an identity involving the vec$(\cdot)$ operator (Magnus and Neudecker, 1999). For matrices $\mathbf{A}$ and $\mathbf{C}$ with vector $\mathbf{d}$ of conformable sizes such that $\mathbf{ACd}$ is well defined, then:

$$\mathbf{ACd} = (\mathbf{d}' \otimes \mathbf{A})\mathbf{C} = (\mathbf{A} \otimes \mathbf{d}')\mathbf{C}'.$$

Let $\mathbf{d} = \boldsymbol{\beta}$, $\mathbf{A} = \mathbf{I}_m$, and $\mathbf{C} = \mathbf{X}'$. Then

$$(\boldsymbol{\beta}' \otimes \mathbf{I}_m)\text{vec}(\mathbf{X}') = \mathbf{I}_m \mathbf{X}' \boldsymbol{\beta} = (\mathbf{I}_m \otimes \boldsymbol{\beta})'\text{vec}(\mathbf{X}). \tag{4.17}$$

Note that the same result for the K factor case follows based on Equations (4.9) and (4.12) and the fact that the vec$(\cdot)$ operator is distributive under addition. Denoting $\mathbf{S} = \left[\|\boldsymbol{\beta}\|^2 \mathbf{I}_m + \lambda \boldsymbol{\Omega}\right]^{-1}$.

A result from Equation (4.17) that will prove useful in Section 4.2.2 is that

$$\hat{\mathbf{f}} = \mathbf{S}\mathbf{X}'\boldsymbol{\beta}. \tag{4.18}$$

To show equivalence of the GCV criteria, expanding the numerator of Equation (4.16) for $\mathbf{H}_1 = (\boldsymbol{\beta} \otimes \mathbf{I}_m)\mathbf{S}(\boldsymbol{\beta}' \otimes \mathbf{I}_m)$ yields:

$$\|(\mathbf{I} - \mathbf{H}_1)\text{vec}(\mathbf{X}')\|^2 = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}^2 - 2\text{vec}(\mathbf{X}')'\mathbf{H}_1\text{vec}(\mathbf{X}') + \text{vec}(\mathbf{X}')'\mathbf{H}_1'\mathbf{H}_1\text{vec}(\mathbf{X}').$$

Clearly the first term is also equal to $\text{vec}(\mathbf{X})'\text{vec}(\mathbf{X})$. Expanding the second term:

$$\text{vec}(\mathbf{X}')'\mathbf{H}_1\text{vec}(\mathbf{X}') = \boldsymbol{\beta}'\mathbf{X}\mathbf{S}\mathbf{X}'\boldsymbol{\beta} = \text{vec}(\mathbf{X})'\mathbf{H}_2\text{vec}(\mathbf{X}),$$

from Equation (4.17) for $\mathbf{H}_2 \equiv (\mathbf{I}_m \otimes \boldsymbol{\beta})\mathbf{S}(\mathbf{I}_m \otimes \boldsymbol{\beta}')$. For the third term, it is worth noting that both $\mathbf{H}_1$ and $\mathbf{H}_2$ are symmetric. For $\mathbf{H}_1$:

$$\mathbf{H}_1' = \left[(\boldsymbol{\beta} \otimes \mathbf{I}_m)\mathbf{S}(\boldsymbol{\beta}' \otimes \mathbf{I}_m)\right]' = (\boldsymbol{\beta}' \otimes \mathbf{I}_m)'\mathbf{S}'(\boldsymbol{\beta} \otimes \mathbf{I}_m)' = \mathbf{H}_1,$$

because $\mathbf{S}$ is symmetric. Therefore

$$\begin{aligned}
\mathbf{H}_1'\mathbf{H}_1 &= \mathbf{H}_1^2 = (\boldsymbol{\beta} \otimes \mathbf{I}_m)\mathbf{S}(\boldsymbol{\beta}' \otimes \mathbf{I}_m)(\boldsymbol{\beta} \otimes \mathbf{I}_m)\mathbf{S}(\boldsymbol{\beta}' \otimes \mathbf{I}_m) \\
&= \|\boldsymbol{\beta}\|^2(\boldsymbol{\beta} \otimes \mathbf{I}_m)\mathbf{S}^2(\boldsymbol{\beta}' \otimes \mathbf{I}_m).
\end{aligned}$$

It is easily shown that equivalent results hold for $\mathbf{H}_2$. Then

$$\begin{aligned}
\text{vec}(\mathbf{X}')'\mathbf{H}_1'\mathbf{H}_1\text{vec}(\mathbf{X}') &= \|\boldsymbol{\beta}\|^2\text{vec}(\mathbf{X}')'(\boldsymbol{\beta} \otimes \mathbf{I}_m)\mathbf{S}^2(\boldsymbol{\beta}' \otimes \mathbf{I}_m)\text{vec}(\mathbf{X}') \\
&= \|\boldsymbol{\beta}\|^2\boldsymbol{\beta}'\mathbf{X}\mathbf{S}^2\mathbf{X}'\boldsymbol{\beta} = \boldsymbol{\beta}'\mathbf{X}\mathbf{S}'(\mathbf{I}_m \otimes \boldsymbol{\beta})'(\mathbf{I}_m \otimes \boldsymbol{\beta})\mathbf{S}\mathbf{X}'\boldsymbol{\beta} \\
&= \text{vec}(\mathbf{X})'\mathbf{H}_2'\mathbf{H}_2\text{vec}(\mathbf{X}),
\end{aligned}$$

again due to Equation (4.17).

To show equivalence of the denominator of GCV criterion (4.16), we just need to show that

$\text{tr}(\mathbf{H}_1) = \text{tr}(\mathbf{H}_2)$:

$$\begin{aligned}
\text{tr}(\mathbf{H}_1) &= \text{tr}\left[(\boldsymbol{\beta} \otimes \mathbf{I}_m)\mathbf{S}(\boldsymbol{\beta}' \otimes \mathbf{I}_m)\right] \\
&= \text{tr}(\|\boldsymbol{\beta}\|^2\mathbf{S}) = \text{tr}((\mathbf{I}_m \otimes \boldsymbol{\beta}')(\mathbf{I}_m \otimes \boldsymbol{\beta})\mathbf{S}) \\
&= \text{tr}((\mathbf{I}_m \otimes \boldsymbol{\beta})\mathbf{S}(\mathbf{I}_m \otimes \boldsymbol{\beta}')) = \text{tr}(\mathbf{H}_2).
\end{aligned}$$

Therefore, the GCV criteria for either formulation in Proposition 4.2.1 are also equivalent. Thus, going forward we present the estimated factor loading curves as the solution to the minimization problem

$$\min_{\mathbf{f}} \|\text{vec}(\mathbf{X}) - (\mathbf{I}_m \otimes \boldsymbol{\beta}) \cdot \mathbf{f}\|^2 + \lambda \mathbf{f}'\boldsymbol{\Omega}\mathbf{f}, \tag{4.19}$$

with corresponding GCV criterion as

$$\begin{aligned}
\hat{\mathbf{f}} &= \mathbf{S}\mathbf{X}'\boldsymbol{\beta}\ , \mathbf{H} = (\mathbf{I}_m \otimes \boldsymbol{\beta})\mathbf{S}(\mathbf{I}_m \otimes \boldsymbol{\beta}'), \qquad (4.20)\\
\text{GCV}(\lambda) &= \frac{\|(\mathbf{I}_{nm} - \mathbf{H})\text{vec}(\mathbf{X})\|^2/nm}{[1 - \text{tr}(\mathbf{H})/nm]^2}.
\end{aligned}$$

To further ease notation, we resume denoting $\text{vec}(\mathbf{X}) = \boldsymbol{X}$. To this end, using Formulation (4.20) it is worth noting the following equivalences among expressions involving $\mathbf{X}$ and $\boldsymbol{X}$:

$$\begin{aligned}
\hat{\boldsymbol{X}} &= \mathbf{H}\boldsymbol{X} = (\mathbf{I}_m \otimes \boldsymbol{\beta})\mathbf{S}\mathbf{X}'\boldsymbol{\beta}, \qquad (4.21)\\
\text{tr}(\mathbf{H}) &= \|\boldsymbol{\beta}\|^2\text{tr}(\mathbf{S}), \\
\|(\mathbf{I}_{nm} - \mathbf{H})\boldsymbol{X}\|^2 &= \|\boldsymbol{\beta}\|^{-2}\|(\mathbf{I}_m - \|\boldsymbol{\beta}\|^2\mathbf{S})\mathbf{X}'\boldsymbol{\beta}\|^2 + \boldsymbol{X}'\boldsymbol{X} - \|\boldsymbol{\beta}\|^{-2}\boldsymbol{\beta}'\mathbf{X}\mathbf{X}'\boldsymbol{\beta}.
\end{aligned}$$

The last relation is obtained by expanding $\|(\mathbf{I}_{nm} - \mathbf{H})\boldsymbol{X}\|^2$, adding and subtracting $\|\boldsymbol{\beta}\|^{-2}\boldsymbol{\beta}'\mathbf{X}\mathbf{X}'\boldsymbol{\beta}$, then using the results of this section to simplify and rearrange terms. Using these relations, the GCV criterion in Equations (4.20) can just as easily be expressed in terms of $\mathbf{S}_{m\times m}$ and $[\mathbf{X}'\boldsymbol{\beta}]_{m\times 1}$ rather than $\mathbf{H}_{nm\times nm}$ and $\boldsymbol{X}_{nm\times 1}$. Either form is used in Section 4.2.2, but the benefit of the former formulation is evident in terms of dimensions of the matrices as compared to the latter. A simplified GCV criterion is therefore presented below. Explicitly, because choice of divisors m vs. nm only adjust the scale of GCV(λ); and $\|\boldsymbol{\beta}\|^{-2}\boldsymbol{\beta}'\mathbf{XX}'\boldsymbol{\beta}$ is conditional on $\boldsymbol{\beta}$ and thus not dependent upon λ GCV criterion (4.20) is equivalent to

$$\mathrm{GCV}(\lambda) \quad = \quad \frac{\|(\mathbf{I}_m - \|\boldsymbol{\beta}\|^2\mathbf{S})\mathbf{X}'\boldsymbol{\beta})\|^2/m}{[1-\mathrm{tr}(\|\boldsymbol{\beta}\|^2\mathbf{S})/m]^2}. \tag{4.22}$$

4.2.2 (G)CV Derivation

In general, cross-validation is based on sequentially leaving out sections of the observed data, estimating a model for each "leave-out" and computing some metric for how well the model predicts the left out sections. Obviously, this procedure can be costly in terms of computation. Therefore, it is preferable to be able to compute the CV or GCV metric without needing to re-estimate a model for each leave-out. In this section it is shown that the CV and GCV criteria for leave-out column deletion of $\mathbf{X}$ results in a closed for expression that obviates re-estimation of the FDFM for each leave-out.

The columns of $\mathbf{X}$ are denoted as $n \times 1$ vectors $\mathbf{x}_j$ for $j = 1, \ldots, m$. Suppose the $n \times 1$ block $\mathbf{x}_j$ is deleted from $\boldsymbol{X}$ and Equation (4.19) is minimized. Let $\hat{\mathbf{f}}^{(-j)} = [\hat{f}_1^{(-j)}, \ldots, \hat{f}_m^{(-j)}]'$ denote the estimate for factor loading curve $\mathbf{f}$ based on the deletion of this jth block of $\boldsymbol{X}$, with $\hat{\mathbf{x}}_j^{(-j)} = \boldsymbol{\beta}\hat{f}_j^{(-j)}$ as the resulting predicted value for $\mathbf{x}_j$. We base our CV criterion on the sequence of "leave-out residuals," the $(-j)$th of which is defined as

$$\|\hat{\mathbf{x}}_j^{(-j)} - \mathbf{x}_j\|^2 = \|\boldsymbol{\beta}\hat{f}_j^{(-j)} - \mathbf{x}_j\|^2, \tag{4.23}$$

for $(-j), j = 1, \ldots, m$. A convenient form exists for this residual exists which obviates re-estimation of the FDFM for each j:

Lemma 4.2.1. *The jth leave-out error sum of squares is*

$$\|\boldsymbol{\beta}\hat{f}_j^{(-j)} - \mathbf{x}_j\|^2 = \mathbf{x}_j'\mathbf{x}_j - \frac{\mathbf{x}_j'\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|^2} + \frac{\left(\|\boldsymbol{\beta}\|\hat{f}_j - \boldsymbol{\beta}'\mathbf{x}_j/\|\boldsymbol{\beta}\|\right)^2}{(1-\|\boldsymbol{\beta}\|^2\mathbf{S}_{jj})^2}, \tag{4.24}$$

where $\mathbf{S}_{jj}$ is the jjth element of the matrix $\mathbf{S} = \left[\|\boldsymbol{\beta}\|^2\mathbf{I}_m + \lambda\boldsymbol{\Omega}\right]^{-1}$.

The derivation of Lemma 4.2.1 is deferred to the next section. Because the cross validation criterion is a function of λ, the first two terms in Equation (4.24) do not affect the optimal choice for λ. Therefore, to obtain a CV criterion, we average the third term in Equation (4.24) over $j = 1, \ldots, m$. Note that the jth element of $[\hat{\mathbf{f}}]_j = \hat{f}_j$ can be expressed as the jth element of $\mathbf{S}\mathbf{X}'\boldsymbol{\beta}$ (from Equation (4.18)). Similarly, $\boldsymbol{\beta}'\mathbf{x}_j = \mathbf{x}_j'\boldsymbol{\beta}$ is the jth element of $\mathbf{X}'\boldsymbol{\beta}$. Then $\hat{f}_j = [\mathbf{S}\mathbf{X}'\boldsymbol{\beta}]_j = \mathbf{S}_{jj}\mathbf{x}_j'\boldsymbol{\beta}$. Thus

$$\begin{aligned}\Big(\|\boldsymbol{\beta}\|\hat{f}_j - \boldsymbol{\beta}'\mathbf{x}_j/\|\boldsymbol{\beta}\|\Big)^2 &= \frac{1}{\|\boldsymbol{\beta}\|^2}\left(\mathbf{x}_j'\boldsymbol{\beta} - \|\boldsymbol{\beta}\|^2\mathbf{S}_{jj}\mathbf{x}_j'\boldsymbol{\beta}\right)^2 = \frac{1}{\|\boldsymbol{\beta}\|^2}\left[(1 - \|\boldsymbol{\beta}\|^2\mathbf{S}_{jj})\mathbf{x}_j'\boldsymbol{\beta}\right]^2 \\ &= \frac{1}{\|\boldsymbol{\beta}\|^2}\left\{[(\mathbf{I}_m - \|\boldsymbol{\beta}\|^2\mathbf{S})\mathbf{X}'\boldsymbol{\beta}]_j\right\}^2.\end{aligned}$$

Therefore, the cross validation criterion (CV) is expressed as

$$\frac{1}{m}\sum_{j=1}^{m}\frac{\left\{[(\mathbf{I}_m - \|\boldsymbol{\beta}\|^2\mathbf{S})\mathbf{X}'\boldsymbol{\beta}]_j\right\}^2}{\|\boldsymbol{\beta}\|^2(1 - \|\boldsymbol{\beta}\|^2\mathbf{S}_{jj})^2}. \tag{4.25}$$

By replacing the denominator weights $\|\boldsymbol{\beta}\|^2(1 - \|\boldsymbol{\beta}\|^2\mathbf{S}_{jj})^2$ of Equation (4.25) with the average value of $\mathbf{S}_{jj}$ which is $\frac{1}{m}\mathrm{tr}(\mathbf{S})$, we get the generalized cross validation criterion

$$\mathrm{GCV}(\lambda) = \frac{\|(\mathbf{I}_{nm} - \|\boldsymbol{\beta}\|^2\mathbf{S})\mathbf{X}'\boldsymbol{\beta})\|^2/m}{\|\boldsymbol{\beta}\|^2[1 - \mathrm{tr}(\|\boldsymbol{\beta}\|^2\mathbf{S})/m]^2}.$$

Which, irrelevant of the scale factor $\|\boldsymbol{\beta}\|^2$, is equivalent to the criterion in Expression (4.22)

4.2.3 Proof of Lemma 4.2.1

From Equations (4.20) and (4.21), we define the prediction error for the jth block of $\boldsymbol{X}$ as

$$\hat{\mathbf{x}}_j - \mathbf{x}_j = \boldsymbol{\beta}\hat{f}_j - \mathbf{x}_j.$$

Define $\hat{\mathbf{f}}^{(-j)}$ as the factor loading curve estimate from the minimization problem (4.19) where the jth block $\mathbf{x}_j$ is deleted from $\boldsymbol{X}$. Suppose we delete $\mathbf{x}_j$ from $\boldsymbol{X}$ and replace it with $\boldsymbol{\beta}\hat{f}_j^{(-j)}$, denote this new vector as $\boldsymbol{X}^*$ with estimate $\hat{\mathbf{x}}_j^*$. Then $\hat{\mathbf{f}}^{(-j)} = \mathbf{S}(\mathbf{I}_m \otimes \boldsymbol{\beta})\boldsymbol{X}^*$ (see Theorem 3.1

from Green and Silverman, 1994) and $\hat{\mathbf{x}}_j^* = \boldsymbol{\beta}\hat{f}^{(-j)}$ and we define the leave-out residual as

$$\boldsymbol{\beta}\hat{f}_j^{(-j)} - \mathbf{x}_j.$$

Denote the $n \times n$ blocks of the $\mathbf{H}$ matrix as $\mathbf{H}_{j,l}$ for $j, l = 1, \ldots, m$, and the jth block of the vector $\hat{\boldsymbol{X}}^* = \mathbf{H}\boldsymbol{X}^*$ as $[\mathbf{H}\boldsymbol{X}^*]_j$; $\hat{\boldsymbol{X}} = \mathbf{H}\boldsymbol{X}$ as $[\mathbf{H}\boldsymbol{X}]_j$. Then

$$\begin{aligned}
\boldsymbol{\beta}\hat{f}_j^{(-j)} &= [\mathbf{H}\boldsymbol{X}^*]_j = \sum_{l \neq j} \mathbf{H}_{j,l}\mathbf{x}_l + \mathbf{H}_{j,j}\boldsymbol{\beta}\hat{f}_j^{(-j)}, \\
\boldsymbol{\beta}\hat{f}_j &= [\mathbf{H}\boldsymbol{X}]_j = \sum_{l=1}^{m} \mathbf{H}_{j,l}\mathbf{x}_l.
\end{aligned}$$

Subtracting $\mathbf{x}_j$ from each side of $\boldsymbol{\beta}\hat{f}_j^{(-j)} = \sum_{l \neq j} \mathbf{H}_{j,l}\mathbf{x}_l + \mathbf{H}_{j,j}\boldsymbol{\beta}\hat{f}_j^{(-j)}$ yields

$$\begin{aligned}
\boldsymbol{\beta}\hat{f}_j^{(-j)} - \mathbf{x}_j &= \sum_{l \neq j} \mathbf{H}_{j,l}\mathbf{x}_l + \mathbf{H}_{j,j}\boldsymbol{\beta}\hat{f}_j^{(-j)} - \mathbf{x}_j \pm \mathbf{H}_{jj}\mathbf{x}_j \\
&= \boldsymbol{\beta}\hat{f}_j - \mathbf{x}_j + \mathbf{H}_{jj}\left[\boldsymbol{\beta}\hat{f}_j^{(-j)} - \mathbf{x}_j\right].
\end{aligned}$$

Therefore, the leave-out residual is expressed in the form

$$\boldsymbol{\beta}\hat{f}_j^{(-j)} - \mathbf{x}_j = (\mathbf{I}_n - \mathbf{H}_{jj})^{-1}(\boldsymbol{\beta}\hat{f}_j - \mathbf{x}_j).$$

It is worth noting that

$$\mathbf{H}_{jj} = [(\mathbf{I}_m \otimes \boldsymbol{\beta})\mathbf{H}(\mathbf{I}_m \otimes \boldsymbol{\beta}')]_{jj} = \mathbf{S}_{jj}\boldsymbol{\beta}\boldsymbol{\beta}',$$

and that

$$(\mathbf{I}_n - \mathbf{H}_{jj})^{-1} = \mathbf{I}_n + \frac{\mathbf{S}_{jj}}{1 - \mathbf{S}_{jj}\|\boldsymbol{\beta}\|^2}\boldsymbol{\beta}\boldsymbol{\beta}'. \tag{4.26}$$

This result is verified by multiplying the right-hand side of the expression by the inverse if the left-hand side.

Let $\mathbf{p} = \boldsymbol{\beta}\hat{f}_j - \mathbf{x}_j$. Then

$$\begin{aligned} \|\mathbf{p}\|^2 &= \mathbf{x}_j'\mathbf{x}_j - 2\mathbf{x}_j\boldsymbol{\beta}\hat{f}_j + \|\boldsymbol{\beta}\|^2\hat{f}_j^2 \pm \frac{(\mathbf{x}_j'\boldsymbol{\beta})^2}{\|\boldsymbol{\beta}\|^2} \\ &= \mathbf{x}_j'\mathbf{x}_j - \frac{(\mathbf{x}_j'\boldsymbol{\beta})^2}{\|\boldsymbol{\beta}\|^2} + \left(\|\boldsymbol{\beta}\|\hat{f}_j - \frac{\boldsymbol{\beta}'\mathbf{x}_j}{\|\boldsymbol{\beta}\|}\right)^2, \qquad (4.27) \end{aligned}$$

for Euclidean norm $\|\cdot\|^2$. Also, with $\boldsymbol{\beta}'\mathbf{p} = \|\boldsymbol{\beta}\|^2(\hat{f}_j - \boldsymbol{\beta}'\mathbf{x}_j/\|\boldsymbol{\beta}\|^2)$, then

$$\frac{(\boldsymbol{\beta}'\mathbf{p})^2}{\|\boldsymbol{\beta}\|^2} = (\|\boldsymbol{\beta}\|\hat{f}_j - \frac{\boldsymbol{\beta}'\mathbf{x}_j}{\|\boldsymbol{\beta}\|})^2. \qquad (4.28)$$

With Identity (4.26), the squared norm of the leave-out residual, $\boldsymbol{\beta}\hat{f}_j^{(-j)} - \mathbf{x}_j$, is expressed as

$$\begin{aligned} \|(\mathbf{I}_n - \mathbf{H}_{jj})^{-1}(\boldsymbol{\beta}\hat{f}_j - \mathbf{x}_j)\|^2 &= \left\|\left(\mathbf{I}_n + \frac{\mathbf{S}_{jj}}{1 - \mathbf{S}_{jj}\|\boldsymbol{\beta}\|^2}\boldsymbol{\beta}\boldsymbol{\beta}'\right)\mathbf{p}\right\|^2 \\ &= \left\|\mathbf{p} + \frac{\mathbf{S}_{jj}}{1 - \mathbf{S}_{jj}\|\boldsymbol{\beta}\|^2}\boldsymbol{\beta}\boldsymbol{\beta}'\mathbf{p}\right\|^2. \end{aligned}$$

Expanding these terms, combined with Equations (4.27) and (4.28), culminates in Equation (4.24) of Lemma (4.2.1).

4.3 Natural Cubic Splines

Previously, in 2.3, the penalty matrix $\boldsymbol{\Omega}$ was introduced as the product of matrices involving first and second differences of the observed $\{t_j\}$. The motivation was that a penalty based on the squared second derivative of the factor loading curve $f_k(t)$ is approximated by an expression involving differences of the discrete f_{kj}.

By introducing a slightly different formulation of $\boldsymbol{\Omega}$ that is still based only on the observed $\{t_j\}$, it can be shown that the estimated $\hat{\mathbf{f}}_k$ form natural cubic splines. This fortunate result readily facilitates a method for interpolation which in turn verifies that the estimated $\hat{f}_k(\cdot)$ are true functions.

4.3.1 The Penalty Matrix

Specifically, following Green and Silverman (1994), let $h_j = t_{j+1} - t_j$. For $j = 1, \dots, m$, we define the banded matrix $Q_{m\times(m-2)}$ with columns numbered in a non-standard way: elements $q_{jj'}$ denote the $j = 1, \dots, m$th row and $j' = 2, \dots, m-1$st column of Q. These elements in particular for $|j - j'| < 2$ are given by

$$q_{j-1,j} = h_{j-1}^{-1}, \quad q_{jj} = -h_{j-1}^{-1} - h_j^{-1}, \quad q_{j+1,j} = h_j^{-1}, \tag{4.29}$$

and are 0 otherwise. Further, we define the symmetric matrix $R_{(m-2)\times(m-2)}$ with elements $r_{jj'}; j, j' = 2, \dots, (m-1)$ such that $r_{jj'} = 0$ for $|j - j'| \geq 2$ and otherwise

$$\begin{cases} r_{jj} = \frac{1}{3}(h_{j-1} - h_j) \text{ for } j = 2, \dots, m-1, \\ r_{j,j+1} = r_{j+1,j} = \frac{1}{6}(h_{j-1} - h_j) \text{ for } j = 2, \dots, m-2. \end{cases} \tag{4.30}$$

Note that R is diagonal dominant and thus it is positive definite and invertible. Let

$$\mathbf{\Omega} = QR^{-1}Q'. \tag{4.31}$$

The following result is based on Theorem 2.1 of Green and Silverman (1994).

Proposition 4.3.1. *For fixed k, the $\hat{f}_k(\cdot)$ optimizing* PSS *in* (4.7) *is a natural cubic spline with knot locations at t_j, and*

$$\int \left[f_k''(t)\right]^2 dt \quad = \quad \mathbf{f}_k \mathbf{\Omega} \mathbf{f}_k'.$$

4.3.2 Proof of Proposition 4.3.1

For each $k = 1, \dots, K$, the solution for the optimal factor loading curve, from Equations (4.6) and (4.7), $\hat{f}_k(\cdot)$ solves the minimization problem

$$\min_{f_k(\cdot)} \left[\frac{1}{\sigma^2} \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - \sum_{k=1}^{K} \beta_{ik} f_{kj})^2 + \sum_{k=1}^{K} \lambda_k \int \left[f_k''(t)\right]^2 dt \right],$$

which is equivalent to

$$\min_{f_k(\cdot)} \left[\frac{1}{\sigma^2} \sum_{i=1}^{n} \sum_{j=1}^{m} (x^*_{ij} - \beta_{ik} f_{kj})^2 + \lambda_k \int \left[f''_k(t) \right]^2 dt \right],$$

for

$$x^*_{ij} \equiv x_{ij} - \sum_{h \neq k} \beta_{ih} f_{hj},$$

in solving for a specific $\hat{f}_k(\cdot)$. We denote $\hat{f}_{kj} = \hat{f}_k(t_j)$ and $f_{kj} = f_k(t_j)$. From Theorem 2.3 of Green and Silverman (1994), a natural cubic spline (NCS) that interpolates coordinates $(t_j, \hat{f}_j)$ is the unique minimizing function of $\int [f''_k(t)]^2 \, dt$ over all functions which interpolate $(t_j, \hat{f}_j)$. Therefore, the minimizing function $\hat{f}_k(\cdot)$ of (4.32) is an NCS with knot locations $t_1, \ldots, t_j, \ldots, t_m$. Because an NCS which interpolates (t_j, f_{kj}) is unique, $\hat{f}_k(\cdot)$ is the unique NCS minimizer of (4.32). With $\mathbf{\Omega}$ defined as in Equations (4.29), (4.30) and (4.31), Theorem 2.1 of Green and Silverman (1994) asserts that $\int [f''_k(t)]^2 \, dt = \mathbf{f}_k \mathbf{\Omega} \mathbf{f}'_k$. with $\mathbf{f}'_k = [f_{k1}, \ldots, f_{km}]$. Therefore, the argument to the minimization problem in Equation (4.32) is equivalently written as

$$\frac{1}{\sigma^2} \sum_{i=1}^{n} \sum_{j=1}^{m} (x^*_{ij} - \beta_{ik} f_{kj})^2 + \lambda_k \mathbf{f}_k \mathbf{\Omega} \mathbf{f}'_k.$$

Extending this result to *each* $k = 1, \ldots, K$, we can equivalently write the penalized sum of squares in (4.7) as

$$l_p(\mathbf{X}, \mathbf{B}) = l(\mathbf{B}) + l(\mathbf{X}|\mathbf{B}) + \sum_{k=1}^{K} \lambda_k \mathbf{f}_k \mathbf{\Omega} \mathbf{f}'_k.$$

4.3.3 Forecasting and Curve Synthesis

Recall that the goal of our Functional Dynamic Factor Model (FDFM) is to provide forecasts of an *entire* curve from an observed time series of sampled curves. Once the FDFM has been estimated, it is a straightforward exercise to do just this. Further, due to the functional nature of the model, we are not restricted to forecasts for only the observed knot locations; the natural cubic spline (NCS) results of Section 4.3.1 allow us to forecast to any degree of fineness between knot locations. Indeed, Proposition 4.3.1 even allows within sample imputation of an entire time series.

Forecasting is straightforward: for illustrative purposes, suppose we estimate our FDFM with K factors following an AR(1) process with constants $\{c_k\}$, $k = 1, \ldots, K$. Then the h-step ahead forecasted curve $\hat{x}_{n+h|n}(t)$ is based on the components of the forecast of the factor time series $\hat{\beta}_{n+h|n,k}$ and the estimated factor loading curves $\hat{f}_k(t)$:

$$\begin{cases} \hat{x}_{n+h|n}(t) = \sum_{k=1}^{K} \hat{\beta}_{n+h|n,k} \hat{f}_k(t) \\ \hat{\beta}_{n+h|n,k} = \hat{c}_k + \hat{\varphi}_k \hat{\beta}_{n+h-1,k} = \sum_{r=0}^{h-1} \hat{\varphi}^r \hat{c}_k + \hat{\varphi}_k^h \beta_{nk}. \end{cases} \tag{4.32}$$

The NCS result of Section 4.3.1 ensures that $\hat{f}_k(t)$ is indeed a function rather than a discrete set of points. Thus, we can interpolate $\hat{f}_k(t)$ to any degree of fineness between any two knot locations t_j and t_{j+1}.

Specifically, consider $t \in [t_j, t_{j+1}]; j = 1, \ldots, m$. We can compute values for an entire time series $\{\hat{x}_1(t)\}_{i=1}^{n}$ because each $\hat{f}_k(t)$ is an NCS. Denote $\gamma_{kj} \equiv f_k''(t_j)$. It can be shown (Green and Silverman, 1994)

$$\begin{aligned} \hat{f}_k(t) \quad = \quad & \frac{(t - t_j) f_{k,j+1} + (t_{j+1} - t) f_{kj}}{h_j} + \frac{1}{6}(t - t_j)(t_{j+1} - t) \times \\ & \left[\left(1 + \frac{t - t_j}{h_j}\right) \gamma_{k,j+1} + \left(1 + \frac{t_{j+1} - t}{h_j}\right) \gamma_{kj} \right], \end{aligned} \tag{4.33}$$

for each $k = 1, \ldots, K$. For $t < t_1$, or $t > t_m$, the $\hat{f}_k(t)$ is a linear extrapolation; we illustrate this limitation in Section 7.2.3. Using this method together with Equations (4.32) we can just as easily impute *and* forecast at the same time; a result that enables, for example, yield forecasts for bonds of maturities that *have not been observed.*

Chapter 5

Simulation Results

This chapter focuses on model performance based on simulated data. SVD and the smooth factor model (SFM) from Shen (2009) are used as comparison models for the FDFM. The added benefit of using simulated data over actual data is that knowing the true value of the underlying parameters allows inference as to the bias and variability of the estimated parameters. Further, knowledge of the error component of the simulated data permits forecast inference based on both the raw simulated data, and also just the deterministic component of the data.

5.1 Introduction

One benefit of simulated data is that we are able to assess parameter accuracy as well as forecast accuracy. In the simulation study discussed in Section 5.2 below, we show that our FDFM achieves both of these objectives. By illustrating goodness-of-fit of the FDFM in a simulated setting, we demonstrate ours is a viable model robust to use for various intents; not one unduly tailored to a single specific application, be it call volume or yield curve forecasting. In the latter instance, where there does exist a substantive meaning for parameter values based on some underlying theory, our positive results regarding parameter accuracy lend credence to the interpretability of our model estimates.

The set up for this simulation is motivated by call volumes from a call center – an important application in workforce management of service systems. In call centers, the majority of operating expenses are due to capacity (Gans et al., 2003). Thus, optimal staffing is of critical concern, and this requires accurate call volume forecasting. Our simulation stems in particular from Shen and Huang (2008) who proposed a method for producing forecasts of call arrival volumes from a U.S. financial services firm. Call volumes are recorded at multiple intervals throughout the business day; and volumes corresponding to each of those intervals are recorded from day to day. The authors consider forecasting both within day call arrivals for the intervals throughout the day as well as forecasting the entire within day call arrival pattern from day to day.

5.2 Simulation Studies

Call volume data is essentially a very long time series with multiple periodicities. Volumes are collected in intervals occurring throughout the day, and collected each business day throughout the year. Given the length and frequency, this type of data typically exhibit multiple periodic components (Taylor, 2008). To mimic this behavior, we simulate a very long univariate time series characterized by: 1. high frequency of measurement and 2. multiple periodicity.

The idea is to "block" the high frequency univariate time series into a multivariate time series of sparser frequency. An example outside of the call data scenario is to consider a single time series of monthly data with strong annual seasonality. This can be converted to a multivariate time series consisting of twelve annual time series; each of the twelve representing a particular month.

This idea is hardly a new one: Hurd and Miamee (2007), for example, present a comprehensive treatment of such periodic time series data in their text. The reason for this type of data in regard to the FDFM, however, is to model the periodic cycle as *functional*. Thus, we decompose a seasonal or periodic time series into a *functional time series.*

5.2.1 Simulation Design

We design our simulation based on the actual call center data analyzed in Shen and Huang (2008) where the number of incoming calls N_{ij} is recorded for each of $j = 1, \ldots, m = 68$ *intra-day* intervals over $i = 1, \ldots, n = 210$ consecutive business days (Monday to Friday). In addition to the periodicity of the intra-day call volumes, we further introduce a day-of-the-week cycle in the *inter-day* call values which was exhibited in the real data from Shen and Huang (2008). To this end we will use indicator variables to capture this effect in our factor time series. Earlier, in Section 2.5.2, we briefly discussed how our FDFM is perfectly capable of using explanatory variables in the dynamic factors; our present simulation study illustrates how easily this addition is implemented.

Data for a two factor design is generated, where each independent factor follows an AR(1) process with five intercepts to mimic a day-of-the-week effect. Because call volumes – hypothetical or otherwise – are count data, we use the convention that the square root transform of Poisson data is approximately normal: $x_{ij} = \sqrt{N_{ij} + 1/4}$ (see Brown et al., 2010). Then we generate our data as follows:

$$\begin{cases} x_{ij} = \sum_{k=1}^{2} \beta_{ik} f_{kj} + \epsilon_{ij} \text{ ,with } \epsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma^2) \\ \beta_{ik} = c_{d_{i-1},k} + \varphi_k \beta_{i-1,k} + v_{ik} \end{cases} \tag{5.1}$$

with $v_{ik} \overset{i.i.d.}{\sim} N(0, \sigma_k^2)$ and $E[\epsilon_{ij} v_{sk}] = 0$ for $i, s = 1, \ldots, n, j = 1, \ldots, m, d_i = 1, 2, 3, 4, 5$ (denoting the day of the week), and $k = 1, 2$. Selection of the true parameter values for the simulation will be discussed below.

We compare our model with two others. The first one is the method used by Shen and Huang (2008). They use singular value decomposition of the observed data matrix $\mathbf{X}$ to obtain values for the time series factors $\boldsymbol{\beta}_k$ and factor loading curves $\mathbf{f}_k$; we refer to this method as the SVD model hereafter. This is the method introduced in Chapter 2 to obtain starting values for EM estimation. This is also the model used in Shen (2009) as a benchmark for their smooth factor model (SFM); that model is the second by which we compare ours. The SFM uses the first K left singular vectors and singular values as the dynamic factors $\boldsymbol{\beta}_k$. The corresponding K right singular vectors are smoothed in a sequential matter to form factor loading curves $\mathbf{f}_k$. The first factor loading curve is found with a penalized sum of squares criterion like ours and akin to Green and Silverman (1994). The second curve is found in a similar manner but based on the residual values given the first factor and factor loading curve.

Although both methods (SFM and FDFM) give smooth factor loading curve estimates, the key difference between them is that SFM is estimated in two steps. The SFM estimates the parameters of the time series models for the dynamic factors separately from the factor loading curves. Our FDFM, however, estimates them simultaneously in a single step using the EM formulation. As a result, our simulation results in Tables 5.1 and 5.2 show that the FDFM gives estimates with smaller bias and overall mean squared error (MSE). In addition, another distinction between the two models is that the SFM extracts the loading curves sequentially using residual matrices.

For a fair comparison between SFM and FDFM, the true parameter values for the simulation are based on both SFM and FDFM estimates of the actual call center data in Shen and Huang (2008). The true factor loading curves are provided by SFM estimates from the call data; true dynamic factor parameter values (c_{11}, φ_1, etc.) are furnished by FDFM estimates of the same data. In our design, only σ^2, the overall model variance, is varied in the simulations over three values. All other parameters remain fixed. Those parameter values and factor loading curves are shown in Figure 6.2.

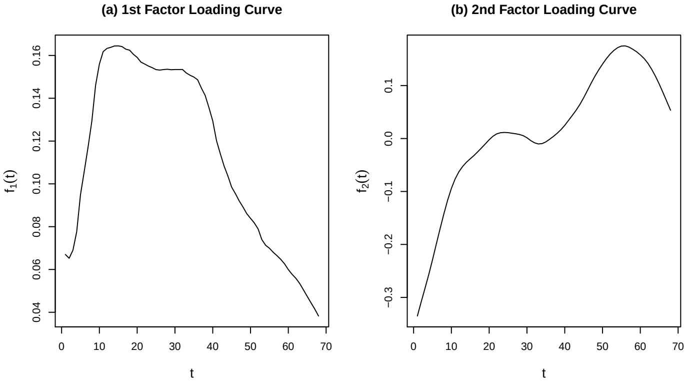
We generate 100 data sets of size $n = 210$ by $m = 68$ per value of σ^2 (300 total data sets). For each set, the three models are estimated on a 205 × 68 rolling window to produce 5 rolling 1-step-ahead forecasts in order to represent each day of the week.

5.2.2 Parameter Accuracy

The top panels of Figure 5.2 show the estimates for the second factor loading curve from the SVD, SFM and FDFM models for the case where $\sigma = 2$. The solid black curve shows the true loading curve; the solid colored curve shows the mean of the estimates over the 100 simulations and the dashed curves represent the first and third quartiles. The panel directly below each models' estimates depicts the average squared bias (red/long and short dash), variance (black/dashed) and MSE (blue/solid) of the estimated factor loading curves. In these simulations, the first factor is so dominant that there is little difference, between the three models' estimates for $f_1(t)$.

However, a great deal of difference exists among the second factor loading curve $f_2(t)$ estimates. The first and third quartile bands are much closer for the FDFM and SFM. The squared

Figure 5.1: (a)-(b) Factor Loading Curves. Factor loading curves used to simulate the data are based on SFM estimates from the actual call center data. (c) Simulation Parameters. Parameter values are based on FDFM estimates of call center data.

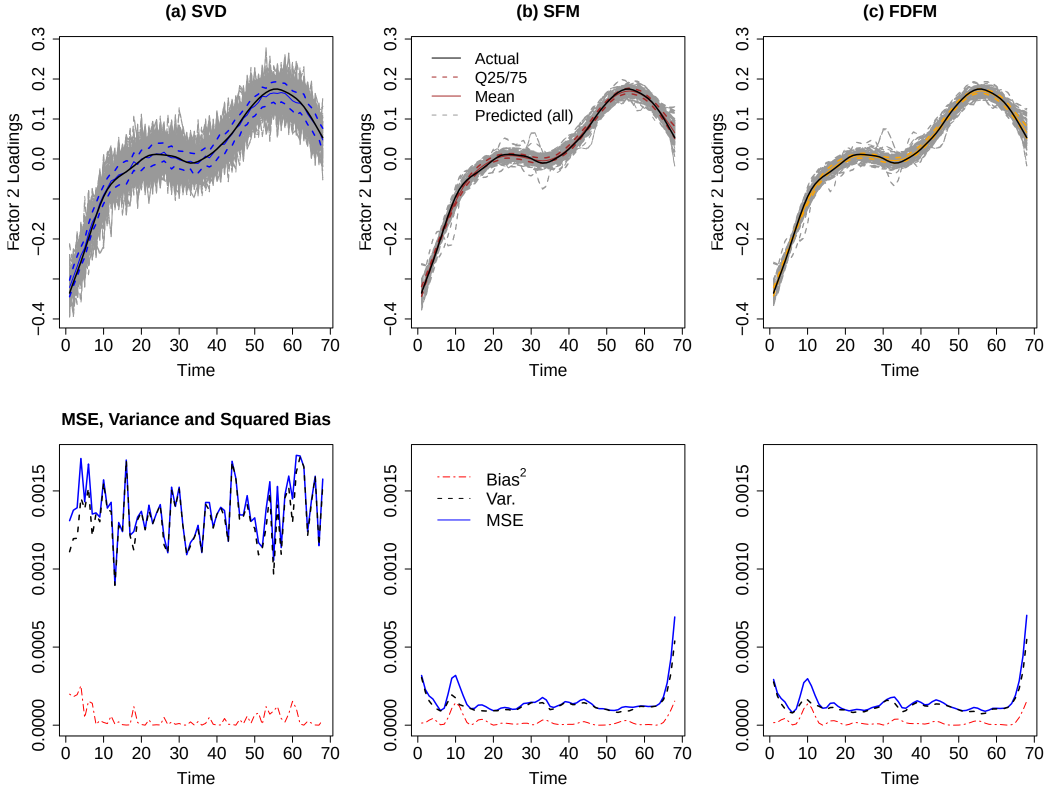


(c) Simulation Parameters

Parameter	First Factor	Second Factor
Intercept 1	$c_{11} = 58.24$	$c_{12} = -2.48$
Intercept 2	$c_{21} = 64.04$	$c_{22} = 0.92$
Intercept 3	$c_{31} = 65.65$	$c_{32} = 0.39$
Intercept 4	$c_{41} = 68.53$	$c_{42} = -6.50$
Intercept 5	$c_{51} = 86.36$	$c_{52} = 7.48$
Slope	$\varphi_1 = 0.72$	$\varphi_2 = 0.52$
Factor Std.	$\sigma_1 = 4.18$	$\sigma_2 = 1.90$
Overall Std.	$\sigma \in \{0.75, 1.5, 2.0\}$	

bias among the methods, though close in magnitude, is smaller and less variable for the FDFM and SFM. Further, both the MSE and variance are drastically smaller for the FDFM and SFM as well. This reduction in variance is due to the smooth regularization inherent in the SFM and FDFM models; SVD has no requirement for smoothness. Recalling the estimation method for the SFM from Section 5.2.1, we see that it is quite similar to the method for the FDFM from Chapter 2; the key difference being the incremental nature of the SFM smoothing. It is because of this similarity that there is little distinction between the models' estimates.

Figure 5.2: Factor Loading Curve Estimates for $f_2(t)$ from the SVD, SFM, and FDFM models where $\sigma = 2$. The top row shows the estimated factor loading curves with mean and quartile bands. The second row shows the MSE, squared bias and variance. The FDFM produces much less variable estimates than SVD, while FDFM and SFM estimates are quite comparable.



However, harken back to the other distinguishing feature of the FDFM from the SFM: the simultaneous estimation of the FDFM as opposed to the two step estimation of SFM. In comparing the other parameter estimates of the models, we do indeed see markedly better estimates produced by our FDFM over SFM. Specifically, in Table 5.1, we see the bias and standard deviation for the factor parameters and overall standard deviation σ for the case where $\sigma = 2$. For Factor 1 parameters, FDFM estimates display uniformly lower bias (in magnitude) than the SVD and SFM estimates. Standard deviation is also lower for all but error standard deviation σ_1. For Factor 2 estimates, the FDFM again achieves the lowest bias while the SFM typically shows a lower standard deviation in its estimates. To reconcile the second factor results, refer to Table 5.2 which displays mean squared error (MSE) of the parameter estimates which is equal to the sum of the squared bias and variance. FDFM MSE is lowest for all Factor 1 estimates, the overall standard deviation and four of the seven Factor 2 parameters. Thus, the FDFM displays the greatest accuracy in parameter estimation among the models.

This is a key result in settings such as in yield curve forecasting where there exist substantive interpretations for the factors, their loadings and parameters (see Chapter 7). Further, in cases where observed regressors are included like the macroeconomic indicators in Diebold et al. (2006), we can be confident the associated coefficients predicted by our FDFM are accurate.

5.2.3 Forecast Performance

Root mean squared forecast error (RMSFE) is computed to assess performance among the methods on forecasting the 206-210th observed curves in each of the simulated data sets with the forecasts computed by Equations (4.32). Recall from Chapter 2 that the time series of curves consist of a smooth underlying curve, $y_i(t)$, plus an error component, $\epsilon_i(t)$:

$$x_i(t) = y_i(t) + \epsilon_i(t). \tag{5.2}$$

The goal of our FDFM is to forecast the smooth curve.

A benefit of a simulation study is knowledge of the exact data generating process. Therefore, in our assessment, we focus on the fit of the model predictions $\hat{x}_{ij}$ compared with the non-random component of x_{ij}. Using Equation (5.1), we can further extract the error components of the dynamic factors. Consider the following decomposition from substituting the second

Table 5.1: Parameter Bias and (Std. Dev.) for the case where $\sigma = 2$. Smallest bias (in magnitude) and standard deviation are indicated in **bold**. Despite comparable performance of SFM and FDFM on factor loading curves, FDFM estimation results in nearly uniform lower bias and standard deviation for first factor parameters. For second factor estimates, FDFM produces lowest bias while for the majority of parameters SFM displays lower standard deviation. See Table 5.2 for reconciliation.

	First Factor			Second Factor		
Parameter	SVD	SFM	FDFM	SVD	SFM	FDFM
Intercept 1	-24.236	-24.235	**-5.474**	-1.036	-1.04	**0.004**
(c_{1k})	(14.35)	(14.352)	(**13.507**)	(0.464)	(**0.434**)	(0.583)
Intercept 2	-22.983	-22.983	**-5.267**	0.037	0.063	**0.026**
(c_{2k})	(13.596)	(13.598)	(**12.843**)	(0.426)	(**0.406**)	(0.469)
Intercept 3	-22.502	-22.501	**-5.115**	-0.221	-0.176	**0.034**
(c_{3k})	(13.359)	(13.362)	(**12.609**)	(0.443)	(**0.411**)	(0.474)
Intercept 4	-22.358	-22.358	**-5.104**	-0.024	-0.242	**-0.045**
(c_{4k})	(13.266)	(13.268)	(**12.52**)	(0.401)	(**0.394**)	(0.473)
Intercept 5	-22.675	-22.674	**-5.283**	1.304	1.442	**-0.026**
(c_{5k})	(13.348)	(13.35)	(**12.624**)	(0.587)	(**0.568**)	(0.776)
Slope	0.094	0.094	**0.021**	0.243	0.23	**-0.015**
(φ_k)	(0.056)	(0.056)	(**0.052**)	(0.075)	(**0.074**)	(0.111)
Factor Std.	-0.541	-0.553	**0.092**	-1.095	-0.937	**0.07**
(σ_k)	(**0.268**)	(0.269)	(0.299)	(**0.146**)	(0.15)	(0.221)
Overall Std.	0.043	0.036	**0.007**			
(σ)	(0.011)	(0.011)	(0.011)			

Table 5.2: MSE for the case where $\sigma = 2$. Smallest MSE indicated in **bold**. FDFM estimates result in uniformly lower MSE for factor one parameters. Despite lower standard deviation of SFM factor two estimates (Table 5.1), the majority of FDFM estimates achieve lower MSE.

	First Factor			Second Factor		
Parameter	SVD	SFM	FDFM	SVD	SFM	FDFM
Intercept 1 (c_{1k})	793.3	793.3	**212.4**	1.287	1.269	**0.34**
Intercept 2 (c_{2k})	713.1	713.1	**192.7**	0.183	**0.169**	0.221
Intercept 3 (c_{3k})	684.8	684.8	**185.1**	0.245	**0.2**	0.226
Intercept 4 (c_{4k})	675.9	675.9	**182.8**	**0.161**	0.214	0.225
Intercept 5 (c_{5k})	692.3	692.3	**187.3**	2.045	2.401	**0.603**
Slope (φ_k)	0.012	0.012	**0.003**	0.064	0.059	**0.013**
Factor Std. (σ_k)	0.365	0.378	**0.098**	1.219	0.901	**0.054**
Overall Std. (σ)	0.002	0.001	**0.000**			

equation of (5.1) into the first:

$$x_{ij} = z_{ij} + \nu_{ij} + \epsilon_{ij}, \tag{5.3}$$

with $\nu_{ij} = \sum_{k=1}^{K} v_{ik} f_{jk}$ and $z_{ij} = \sum_{k=1}^{K} (c_{d_{i-1},k} + \varphi_k \beta_{i-1,k}) f_{kj}$.

With this decomposition, the RMSFE is then based on $E[x_{n+1,j}] = z_{n+1,j}$ and the forecast $\hat{x}_{n+1,j}$. Because the motivation for our simulation is based on call center data, when we compare RMSFE, we convert back to the count data metric for call volumes from Section 5.2.1. Let $\hat{N}_{n+1,j} = \hat{x}^2_{n+1,j} - 1/4$ and $N_{n+1,j} = z^2_{n+1,j} - 1/4$. Then

$$\text{RMSFE}_{n+1} = \sqrt{\frac{1}{m} \sum_{j=1}^{m} (N_{n+1,j} - \hat{N}_{n+1,j})^2},$$

which we average over the five rolling forecasts.

The results are shown in Figure 5.3 for the simulated data sets for the three values of σ. As σ increases, the SFM and FDFM outperform SVD by an increasingly larger margin. The FDFM, however displays slightly lower RMSFE and less variability than SFM.

Thus, despite the performance of the SFM model in producing smooth factor loading curves and more accurate forecasts than the benchmark SVD model, the FDFM produces smooth curves, better forecasts, and considerably more accurate parameter estimates than SFM.

5.3 Conclusion

In our simulation study, we have shown that our FDFM produces accurate forecasts as well as accurate parameter estimates. Further, we have done so in a simulated setting separate from the various true applications discussed in this thesis. In any of those settings, exceptional forecast accuracy is of obvious importance. However, we also must underscore the relevance of producing accurate parameter estimates. In the yield context, this permits reliable interpretation of the factor loading curves akin to Diebold and Li (2006) and dynamic factor coefficients as in Diebold et al. (2006). In Section 5.1 we also emphasized the importance of call center forecasting for workforce management of service systems. Accurate prediction of the day-of-the week effect, then, is of great concern for reliable forecasts; through our simulation, we have shown our FDFM is capable of this.

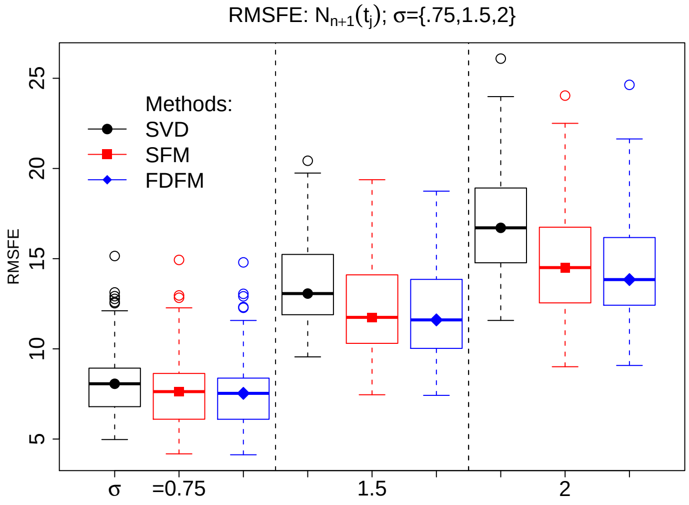


Figure 5.3: Forecast Performance. RMSFE based on N_{ij} from Section 5.2.3 for the SVD, SFM and FDFM models by the three values of σ. As σ increases, the SFM and FDFM outperform SVD by a greater margin.

Chapter 6

Selection and Inference

In this chapter we detail criteria for the selection of the number of factors in the FDFM, methods to select to order of the dynamic factors, and outline a bootstrap procedure to provide confidence intervals for model estimates. Results for each of these are illustrated based on simulated data. The design of the simulation employs an innovative approach in creating data that does not have an overwhelmingly dominant first factor. The method to this end is outlined first, followed by the actual simulation design. Each section thereafter then provides a description of a selection or inference procedure with illustrations the application to the simulated data.

6.1 Factor Weighting

Before we proceed with a simulation to test and implement the methods proposed in the following sections, it is worth noting that in application of SVD or even the FDFM to real data, the percentage of variance explained by the first factor is typically very large. In some cases 99%. If simulated data is based on entirely on model estimates from actual data, then surely this result translates to the analysis of that simulated data. In this setting, that any of the proposed methods above will correctly assess that $K = 1$ is hardly surprising. Therefore, in the following simulation, we will specify our simulated data to have a more equal weighting among the factors. For example, in the yield data analyzed in Chapter 7, SVD decomposition of the observed data reveals the first factor accounts for approximately 99% of the variance in the data, while the next two factors compose most of the minimally remaining 1%.

Ideally, a more representative weighting may be $\{0.65, 0.25, 0.10\}$ or $\{0.75, 0.20, 0.05\}$. Such a weighting would provided better opportunity to assess the testing of $K > 1$. Unfortunately, simulating this type of data from scratch requires many considerations for how to select the orthogonal functions $F(t)$ and generating corresponding time series factors that have reasonable variance and still achieve this weighting. At best the simulated data may be very specific and not linked to any practical true application of the model. However, an intermediate method exists based on real data that can *approximately* achieve a more proportional weighting. Consider the following approach:

1. Real Data. If we are interested in designing a simulation for a specific number of factors K, then given some real data of interest $\mathbf{X}_{n\times m}$, either from Step 0 of the EM, or full estimation of the FDFM, we get predicted values for the factors $\hat{\mathbf{B}}_{n\times K}$ and factor loading curves $\hat{\mathbf{F}}_{m\times K}$. Then, recalling the original model $\mathbf{X} = \mathbf{B}\mathbf{F}' + \boldsymbol{\epsilon} = \mathbf{Y} + \boldsymbol{\epsilon}$ the predicted underlying smooth curves are given by

$$\hat{\mathbf{Y}} = \hat{\mathbf{B}}\hat{\mathbf{F}}'.$$

2. Simulate Data. For the simulation, we can use $\hat{\mathbf{F}}$ as the orthonormal factor loading curves, and generate $\boldsymbol{\epsilon}$ based on values of σ^2 that are reasonably consistent with the real data residuals $\mathbf{X} - \hat{\mathbf{B}}\hat{\mathbf{F}}'$. From each column of $\hat{\mathbf{B}}$, we can estimate an initial AR (or ARMA) model, and use the estimated parameters as the true values for simulated data. Specifically, fit

$$\hat{\beta}_{ik} = c_k + \sum_{r=1}^{P} \varphi_{rk}\hat{\beta}_{i-r,k} + v_{ik},$$

for $k = 1, \ldots, K$. Then use the estimates for $\{c_k, \varphi_{1k}, \ldots, \varphi_{Pk}, \sigma_k^2\}$ to simulate a new set of K independent time series:

$$\alpha_{ik} = c_k + \sum_{r=1}^{P} \varphi_{rk}\alpha_{i-r,k} + v_{ik},$$

for $v_{ik} \overset{iid}{\sim} N(0, \sigma_k^2)$, $i = 1, \ldots, n$. Finally, we create a single preliminary simulated data set $\tilde{\mathbf{X}}$ as

$$\tilde{\mathbf{X}} = \tilde{\mathbf{Y}} + \boldsymbol{\epsilon},$$

for $\tilde{\mathbf{Y}} = \mathbf{A}\hat{\mathbf{F}}'$ where the columns of $\mathbf{A}_{n \times K}$ contain the K simulated AR(P) processes: $\{\alpha_{ik}\}_{i=1}^{n}$.

3. Re-scale Simulated data. $\tilde{\mathbf{X}}$ will resemble the original raw data $\mathbf{X}$, including the possibly non-optimal quality of possessing a single dominant factor with $K-1$ other considerably less dominant factors. But with our prototype simulated data set, we can re-scale to get a more equal weighting. Revisiting Step 0 from the EM consider the singular value decomposition of $\tilde{\mathbf{Y}}$:

$$\tilde{\mathbf{Y}} = \mathbf{U}_{n \times K}\mathbf{D}_{K \times K}\mathbf{V}'_{K \times m},$$

where $\mathbf{D}$ is a diagonal matrix containing the singular values of $\tilde{\mathbf{Y}}$ in descending order $d_1 > \ldots > d_K$ [1]. Because $\hat{\mathbf{F}}$ and $\mathbf{V}$, are both orthonormal, approximately, we can expect $\mathbf{A} \approx \mathbf{UD}$. The percentage of variance in $\tilde{\mathbf{Y}}$ explained by singular vector $\mathbf{u}_k$ is $p_k \equiv d_k^2 \times SS$, where $SS = \sum_{h=1}^{K} d_h^2$. If $\{p_1, \ldots, p_K\}$ is not representative of the weighting of factors we would like in our simulation we designate a new set of percent explained variance $\{p_1^*, \ldots, p_K^*\}$ then we can re-weight to a new set of dynamic factors $\mathbf{B}$ by:

$$\mathbf{B} \equiv \mathbf{A}\mathbf{D}^{-1}\boldsymbol{\Delta},$$

where $\mathbf{D}^{-1}\boldsymbol{\Delta}$ is a diagonal matrix with elements $w_k \equiv \frac{\sqrt{p_k^* SS}}{d_k}$ along the diagonal.

4. Simulate New Data. With the weighting elements $\{w_k\}$ we can simulate new dynamic factors $\{\boldsymbol{\beta}_k\}$ that combined with $\hat{\mathbf{F}}$ will approximate our preferred percent of explained variance per factor $\{p_1^*, \ldots, p_K^*\}$:

$$w_k(\alpha_{ik} - \mu_{\alpha;k}) = \sum_{r=1}^{P} \varphi_{rk}[w_k(\alpha_{i-r,k} - \mu_{\alpha;k})] + w_k v_{\alpha;ik},$$

or

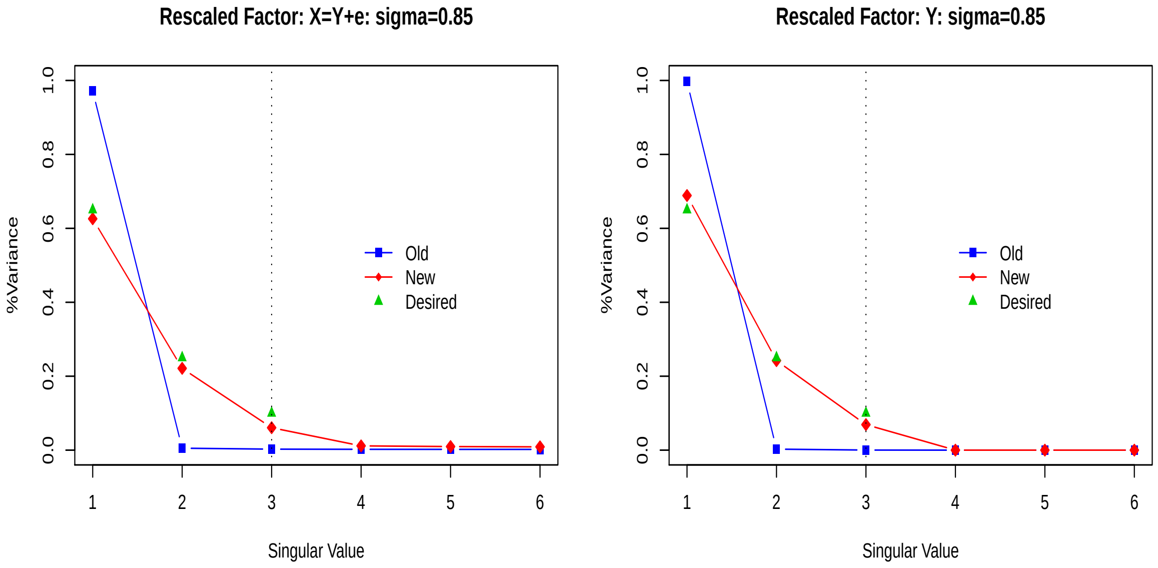
$$(\hat{\beta}_{ik} - \mu_k) = \sum_{r=1}^{P} \varphi_{rk}(\hat{\beta}_{i-r,k} - \mu_k) + v_{ik},$$

with $v_{ik} \sim N(0, w_k^2, \sigma_{\alpha;k}^2)$. With these new simulated factors $\mathbf{B}$, existing factor loading curves $\hat{\mathbf{F}}$ and simulated independent errors $\boldsymbol{\epsilon}$ we construct our new data $\mathbf{X}$. Note that we can adjust the parameter σ^2 to more adequately reflect the scale of the new simulated data. An example

[1] Given that $rank(\tilde{\mathbf{Y}}) = K$, singular values with left and right singular vectors $K+1, \ldots, m$ are equal to 0.

of de-constructing then reconstructing the yield curve data is shown in Figure 6.1 for both the **X** and **Y** data. The blue lines show the original percent of variance of data simulated based on FDFM estimates of the yield data. The green points display our target re-weighting to obtain a three factor data set with $\{0.65, 0.25, 0.10\}$ percent weighting. The red lines then show the results of the method described in this section. Clearly there is some merit to this approach.

Figure 6.1: Example of Factor Re-scaling. We simulate data based on estimated parameters from the true yield curve data in Chapter 7 for 3 factors. Using the method described in Section 6.1, we re-weight the data by scaling the original singular values (blue) to form a new dataset. Those singular values (red) resemble the desired weighting (green).

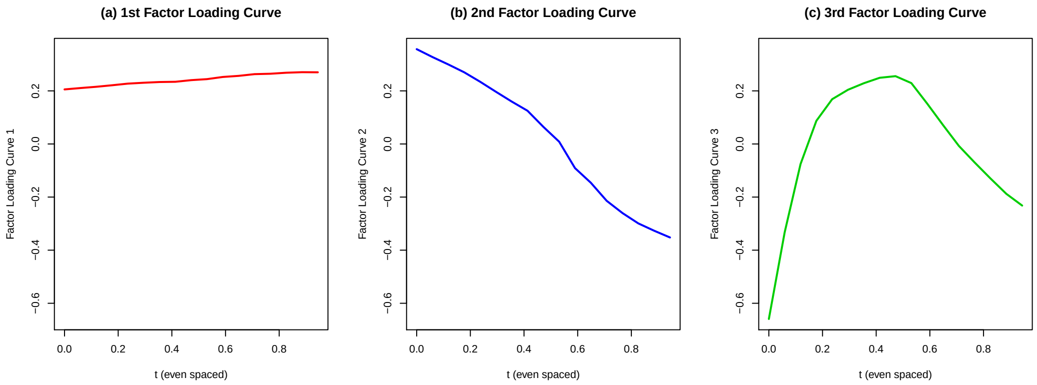


6.2 Simulation Design

To illustrate the re-weighting scheme proposed in Section 6.1 we create some simulated data inspired by the true yield curve data thoroughly examined in Chapter 7. Factor loading curves are selected as those estimated for the period May 1985 to April 1994 of that data. This period is as representative of that data as any other, and was chosen primarily for consistency with the presentation of those estimates in Chapter 7. However, unlike the model estimated in that chapter, we simulate the three dynamic factors as AR(2) processes in order to better illustrate the order selection procedure introduced in Section 6.4.

Preliminary values for σ are chosen based on those estimated in Chapter 7, and initial parameters for the dynamic factors are obtained from AR(2) estimation of $E[\boldsymbol{\beta}|\boldsymbol{X}]$ from the E-step of the yield data. Via the method described in Section 6.1, these parameters are then modified to achieve and approximate 0.65, 0.25, 0.10 weighting among the three factors. These modified parameters, along with factor loading curves, are presented in Figure 6.2. We use these true model parameters to simulate 100 data sets for each of three values for σ for size $n = 105$ and $m = 17$. An example data set is illustrated in Figure 6.3.

Figure 6.2: (a)-(c) Factor Loading Curves. Factor loading curves used to simulate the data are based on FDFM estimate actual yield curve data from Chapter 7. (d) Simulation Parameters. Parameter values are also based on FDFM estimates of the yield curve data.



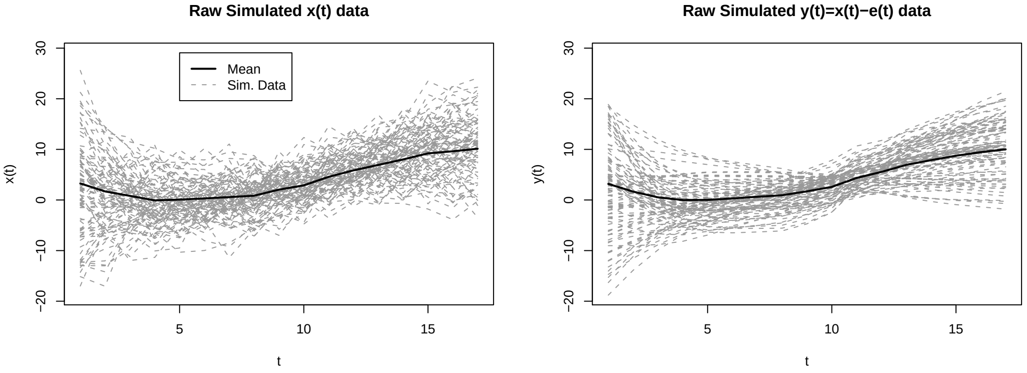
(d) Simulation Parameters

Parameter	First Factor	Second Factor	Third Factor
Mean	$\mu_1 = 20.04$	$\mu_2 = -7.76$	$\mu_3 = -1.54$
AR(1)	$\varphi_{11} = 1.15$	$\varphi_{12} = 1.25$	$\varphi_{13} = 1.16$
AR(2)	$\varphi_{21} = -0.19$	$\varphi_{22} = -0.27$	$\varphi_{23} = -0.32$
Factor Std.	$\sigma_1 = 0.91$	$\sigma_2 = 4.54$	$\sigma_3 = 4.47$
Overall Std.	$\sigma \in \{2.55, 3.75, 5.25\}$		
	$n = 105,\ m = 17$		

6.3 Factor Selection

Until this point, the method for selecting the number of factors has not been considered. As the hypothesis is that the unobserved time series factors drive the observed data, it is not so much an issue of *sel*–ecting K as it is a matter of *det*–ecting K.

Figure 6.3: Example simulated data set according to the design in Figure 6.2. In grey are $x_i(t_j)$ and $y_i(t_j)$ for each $i = 1, \ldots, n$. The mean over all i is shown in black.



6.3.1 Discrete DFM Results

Drawing from the DFM literature Pena and Box (1987) and then Pena and Poncela (2006) prescribe detection methods based on the structure of the observed lagged autocovariance matrices. Namely, that for a non-functional DFM, the rank of such matrices should be K and not m–the number of observed time series. In particular, recall the formulation of the classical DFM from Section 2.1.1:

$$\begin{cases} \mathbf{x}_i = \mathbf{F}_{m\times K}\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i, \\ \Phi(L)\boldsymbol{\beta}_i = \Theta(L)\mathbf{v}_i, \\ \Phi(L) = \mathbf{I}_K - \Phi_1 L - \ldots - \Phi_P L^P, \\ \Theta(L) = \mathbf{I}_K - \Theta_1 L - \ldots - \Theta_Q L^Q, \\ \begin{bmatrix} \boldsymbol{\epsilon}_i \\ \mathbf{v}_i \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0}_{m\times 1} \\ \mathbf{0}_{K\times 1} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_\epsilon & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_v \end{bmatrix} \right), \\ \mathbf{F}'\mathbf{F} = \mathbf{I}_K \end{cases} \tag{6.1}$$

In this setting, the autocovariance matrices $Cov(\mathbf{x}_i, \mathbf{x}_{i-l}) \equiv \boldsymbol{\Gamma}_{\mathbf{x}}(l)$ for lags $l = 0, 1, \ldots$ are functions of the autocovariance matrices for the dynamic factors $Cov(\boldsymbol{\beta}_i, \boldsymbol{\beta}_{i-l}) \equiv \boldsymbol{\Gamma}_{\boldsymbol{\beta}}(l)$:

$$\boldsymbol{\Gamma}_{\mathbf{x}}(l) = \begin{cases} \mathbf{F}\boldsymbol{\Gamma}_{\boldsymbol{\beta}}(l)\mathbf{F}' + \boldsymbol{\Sigma}_\epsilon & \text{if } l = 0, \\ \mathbf{F}\boldsymbol{\Gamma}_{\boldsymbol{\beta}}(l)\mathbf{F}' & \text{otherwise.} \end{cases} \tag{6.2}$$

Therefore, the (short) rank of $\boldsymbol{\Gamma}_{\mathbf{x}}(l)$ is equal to the number of latent dynamic factors K.

In the case of independent dynamic factors, $\boldsymbol{\Gamma}_{\boldsymbol{\beta}}(l)$ is diagonal. Then the orthonormality condition in Model (6.1) implies eigen-decomposition of $\boldsymbol{\Gamma}_{\mathbf{x}}(l)$ is equivalent to Equation (6.2) for $l > 0$: the K non-zero eigenvalues of $\boldsymbol{\Gamma}_{\mathbf{x}}(l)$ correspond to the diagonal elements of $\boldsymbol{\Gamma}_{\boldsymbol{\beta}}(l)$; the corresponding eigenvectors are the columns of the $\mathbf{F}$ matrix. This representation yields two important details regarding the number of factors K. First, only the first K eigenvalues of $\boldsymbol{\Gamma}_{\mathbf{x}}(l)$ are non-zero. Second, under the assumption that the dynamic factors are stationary, the first K eigenvectors do not depend on time i, and are thus identical regardless of lag l (respectively, $k = 1, \ldots, K$). Pena and Box (1987) provides an ad hoc methodology for identifying the dimensionality of the dynamic factors based on this relationship. However, it is unclear how they form an estimate of $\boldsymbol{\Gamma}_{\mathbf{x}}(l)$. Further, it is implied that some foreknowledge of the time series process is required (knowing $\Phi(L)$ and $\Theta(L)$ in Equation (6.1)) since the authors propose an additional assessment for K based on the ranks of the lag polynomial matrices $\Phi(L)$ and $\Theta(L)$. Finally, there is some subjectivity in the assessment of K as there are no definitive test criteria described in these methods.

However, this does provide a starting point for assessing K and even the identification of the (V)ARMA processes for dynamic factors $\boldsymbol{\beta}_i$. Eigen-decomposition of the *sample* autocovariance matrices of the observed data can be used to infer the number of factors K. The sample autocovariance matrices are

$$\mathbf{G}_{\mathbf{x}}(l) \equiv \frac{1}{n-l} \sum_{i=l+1}^{n} (\mathbf{x}_{i-l} - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})', \tag{6.3}$$

for $\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ and $l \geq 0$. Let the eigen-decomposition of Equation (6.3) yield eigenvalues $[e_{1;l}, \ldots, e_{m;l}]$ and eigenvectors $[\mathbf{e}_{1;l}, \ldots, \mathbf{e}_{m;l}]$. Then, K is given by the number of nonzero eigenvalues for lags $l = 1, \ldots$. Those corresponding eigenvectors should be "stable" and nonzero across lags.

Going a step beyond the Pena and Box (1987), methodology, the decay pattern of the K eigenvalues across lags can be determined using traditional time series techniques. Continuing with the stationary and independent factor specification, examination of the sample autocorrelation (ACF) and partial autocorrelation (PACF) functions can reveal the number of AR and MA terms for each independent time series.

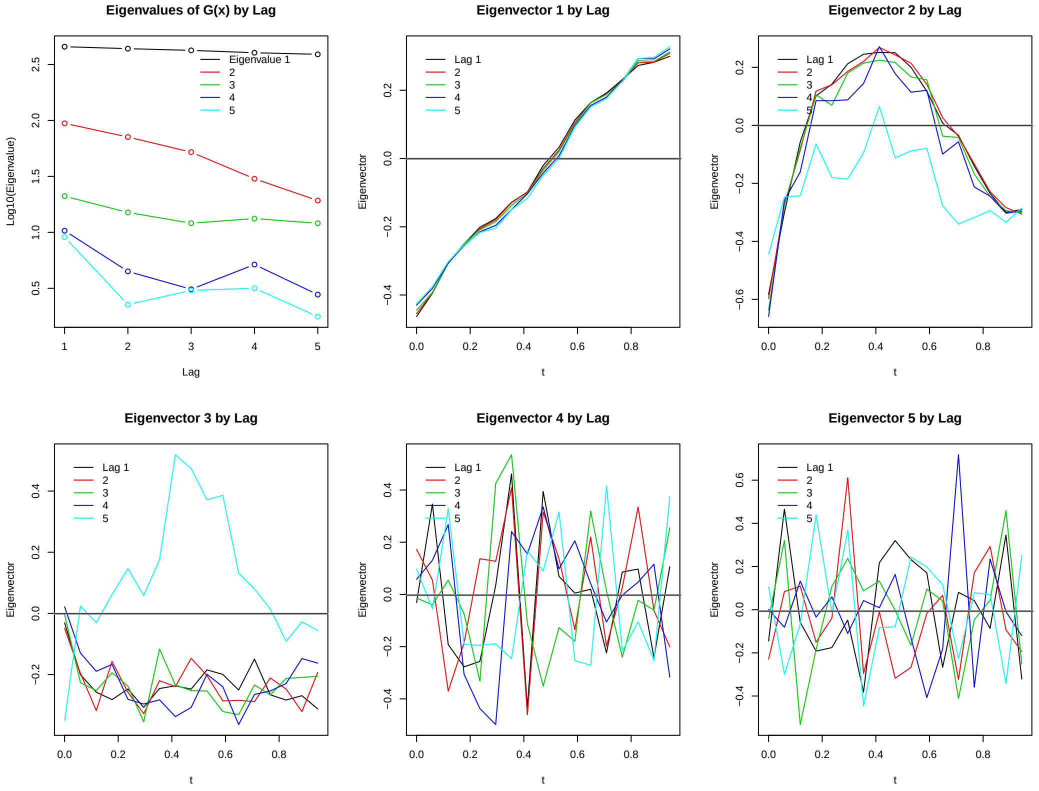
Pursuing dependent dynamic factors is beyond the scope of this thesis, but for illustration, a similar approach can be used. Again following Pena and Box (1987), it is still the case that $\mathbf{\Gamma_x}(l) = \mathbf{F}\mathbf{\Gamma}_{\boldsymbol{\beta}}(l)\mathbf{F}'$ for $l > 0$, but the eigenvectors of $\mathbf{\Gamma_x}(l)$ are no longer contained in $\mathbf{F}$ alone. Rather, via eigen-decomposition of $\mathbf{\Gamma}_{\boldsymbol{\beta}}(l) = \mathbf{U}_l\mathbf{D}_l\mathbf{U}_l^{-1}$, the eigenvectors of $\mathbf{\Gamma_x}(l)$ are now $\mathbf{FU}_l$ and depend on lag l. However, $\mathbf{\Gamma_x}(l)$ and $\mathbf{\Gamma}_{\boldsymbol{\beta}}(l)$ continue to have the same eigenvalues: those contained in the diagonal matrix $\mathbf{D}_l$. Could an estimate for $\mathbf{\Gamma}_{\boldsymbol{\beta}}(l)$ be formed, identification techniques for time series such as sample ACFs and PACFs along with Cross-correlation functions (CCFs) could be utilized to assess the VARMA dependence. Unfortunately, due to this dependence, it is now not immediately clear how to extract this estimate from the sample autocovariance expression in Equation (6.3).

Regardless, to illustrate this approach for independent AR(2) dynamic factors we create some simulated data inspired by the true yield curve data thoroughly examined in Chapter 7. For a detailed description of how the parameters were chosen, see Sections 6.1 and 6.2; for the model parameters, see Figure 6.2. Then examine the resulting eigenvectors and eigenvalues for the sample autocovariance matrices $\mathbf{G_x}(l)$ for lags $l = 1, \ldots, 5$. Although this is not a rigorous testing procedure, it is a method that can provide some initial insight as to the number of factors. Further, it is relatively inexpensive in regard to computation, in that all we require for $\mathbf{G_x}(l)$ is the raw data.

As an example, consider Figure 6.4 which is generated from a simulated data set from the design outlined in Figure 6.2. For lags $l = 1, \ldots, 5$ we examine the resulting components of eigen-decomposition of the sample auto-covariance matrices given in Equation (6.3).

For the $K = 3$ factor model with independent AR(2) factors, the estimated eigenvalues correspond to the univariate autocovariances of the of the time series factors and we expect to

Figure 6.4: Eigenvalues and eigenvectors by lag $l = 1, \ldots, 5$ for $\mathbf{G}_{\mathbf{x}}(l)$, based on Pena and Box (1987). For simulated data consisting of 3 factors, the first 3 eigenvalues of the sample autocovariance matrix are large consistently across lags. For the 5 lags shown, the first 3 eigenvectors noticeably deviate from zero, indicating 3 factors.



see their decay over increasing lags. Further, as in Equation 6.2, per lag we should only observe 3 "large" eigenvalues and the remaining $m - K$ will be close to zero. Observing the first panel of Figure, we observe exactly this: of the five sequences of eigenvalues shown, the fourth and fifth are visibly smaller than the first three.

The next five panels in 6.4 illustrate the first five eigenvectors of $\mathbf{G}_{\mathbf{x}}(l)$ by lag l. Here, for a 3 factor model, in the first three sets we expect some non-zero pattern in the vectors over $\{t_j\}$, and a pattern relatively consistent among lags. Greater-than-Kth eigenvectors will appear "noisy" about 0, and exhibit no pattern among lags. This is what we see. Clearly for the first and second eigenvectors, there is consistency in pattern among lags, and they are not uniformly zero. The third eigenvector, though noisy still noticeably differs from 0, and outside of the fifth lag the pattern is consistent. Note that the eigenvectors are unique up to sign.

The results presented in Pena and Box (1987) are, of course, for the traditional DFM. However, for the functional case, the concept is the same. For the FDFM(K,p) specification

$$x_i(t) = \vec{\mathrm{F}}(t)\boldsymbol{\beta}_i + \epsilon_i(t), \tag{6.4}$$

where $\vec{\mathrm{F}}(t)$ is the $1 \times K$ vector of functions $[f_1(t), \ldots, f_K(t)]$, the autocovariance *functions* $Cov(x_i(t), x_{i-l}(t)) \equiv \Gamma_{x(t)}(l)$ are given by:

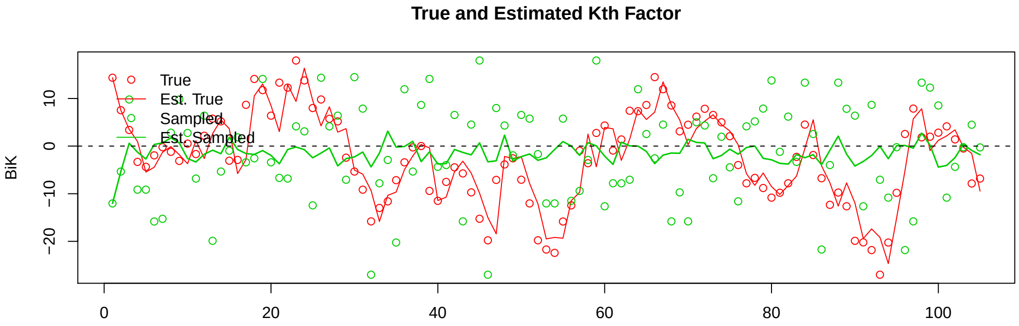
$$\Gamma_{x(t)}(l) = \mathrm{F}(t)\boldsymbol{\Gamma}_{\boldsymbol{\beta}}(l)\mathrm{F}(t)^{\mathrm{T}} + \sigma^2_{I\{l=0\}}. \tag{6.5}$$

In the same manner that there are $K < m$ non-zero eigen-values for $\boldsymbol{\Gamma}_{\mathbf{x}}(l); l > 0$ in the classical DFM (6.1), this is also the case for the FDFM. Continuing with the assumption that the observed data $x_i(t_j)$ is a sampling of the true curves $x_i(t)$, we can use the same methods as those outlined above to infer an initial guess as to the true number of factors. As illustrated in the example, even in the case of $m = 17$ observed series, the underlying number of $K = 3$ factors is still identifiable. However, as mentioned, here there is no hypothesis or test criteria.

6.3.2 A Bootstrap Approach

Consider another approach to more succinctly identify the true number of factors. Based on the simulated data discussed in Section 6.2, we note in Figure 6.5 an example of the true and resampled 3rd factor with corresponding AR(2) estimates. It is obvious that resampling destroys the temporal dependence of the factor. Therefore, perhaps a method exists that can exploit this. Specifically, let

Figure 6.5: True and estimated factors. For K=3 simulated data, the 3rd dynamic factor is plotted with AR(2) estimates (red points/line). The factor is then re-sampled and plotted (green points) with its AR(2) estimates. The re-sample destroys the dynamic dependence, thus resulting AR(2) estimates are close to zero.



$$\mathbf{X} \overset{\text{SVD}}{=} \sum_{j=1}^{m} \mathbf{b}_j \mathbf{v}_j',$$

for $\mathbf{b}_j \equiv d_j \mathbf{u}_j$ as the jth singular value multiplied by the jth right singular vector. For the K_0 factor model, we would like to reject the hypothesis that the true number of factors is some $K < K_0$, and fail to reject the hypothesis that the number of factors is $K < K_0 + 1$. Since we will not know K_0 at the outset, we certainly will not know $K_0 + 1$ either. Thus, consider a sequence of hypotheses tests that the true number of factors K_0 is less than $K + 1$ for $K = 1, 2, \ldots, m$ (or the inaugural K could be chosen based on some initial investigation such as the one described in Section 6.3.1). Without knowledge of the true dynamic factors and corresponding factor loading curves, we can use SVD components as proxies for these. For the hypothesis test $\mathrm{H}_0 : K_0 < K+1$, denote

$$\tilde{\mathbf{X}} = \sum_{k=1}^{K+1} \mathbf{b}_k \mathbf{v}_k'.$$

If there are indeed only K factors, then there should be no dynamic dependence in the $(K+1)$st left singular vector and value $\mathbf{b}_{K+1}$. So some model that predicts dynamic dependence should be indifferent to the ordering of the n elements of $\mathbf{b}_{K+1}$ if it is just noise. Therefore, we model the $(K+1)$st singular value and left singular vector as $\mathbf{b}_{K+1} \sim \mathrm{ARMA}(p,q)$ with resulting estimate $\hat{\mathbf{b}}_{K+1}$. Then define

$$\begin{aligned} \hat{\mathbf{X}} &= \sum_{k=1}^{K} \mathbf{b}_k \mathbf{v}_k' + \hat{\mathbf{b}}_{K+1} \mathbf{v}_{K+1}', \\ \hat{\boldsymbol{\epsilon}} &= \mathbf{X} - \hat{\mathbf{X}}, \end{aligned}$$

intentionally including dynamic component $\hat{\mathbf{b}}_{K+1}\mathbf{v}_{K+1}'$ as part of the residual. We sample $\hat{\boldsymbol{\epsilon}}$ with replacement to obtain $\boldsymbol{\epsilon}^*$ and construct

$$\tilde{\mathbf{X}} = \sum_{k=1}^{K} \mathbf{b}_k \mathbf{v}_k' + \boldsymbol{\epsilon}^*.$$

From $\tilde{\mathbf{X}}$ we extract the $(K+1)$st left singular value and singular vector $\tilde{\mathbf{b}}_{K+1}$. We then estimate the same order $\mathrm{ARMA}(p,q)$ for $\tilde{\mathbf{b}}_{K+1}$, then replace it with the fitted model $\hat{\mathbf{b}}_K^*$ to obtain

$$\mathbf{X}^* = \sum_{k=1}^{K} \mathbf{b}_k \mathbf{v}_k' + \hat{\mathbf{b}}_{K+1}^* \mathbf{v}_{K+1}'.$$

We perform a final SVD on $\mathbf{X}^*$. The idea is that if there are only K factors, then the $(K+1)$st singular value for each of $\mathbf{X}$, $\hat{\mathbf{X}}$, and $\mathbf{X}^*$ should be similar and markedly smaller than the previous ones, if not close to zero. The reason being that in the case of K factors, the $(K+1)$st right singular vector and value should be white noise:

$$\mathbf{X} = \sum_{k=1}^{K} \boldsymbol{\beta}_k \mathbf{f}_k' + \boldsymbol{\epsilon},$$

and

$$\mathbf{X} = \sum_{j=1}^{m} \mathbf{b}_j \mathbf{v}'_j,$$

so that

$$\begin{aligned} \mathbf{X} &\approx \sum_{k=1}^{K} \boldsymbol{\beta}_k \mathbf{f}'_k + \sum_{j=K+1}^{m} \mathbf{b}_j \mathbf{v}'_j, \\ \boldsymbol{\epsilon} &\approx \sum_{j=K+1}^{m} \mathbf{b}_j \mathbf{v}'_j. \end{aligned}$$

Under this hypothesis, since $\mathbf{b}_{K+1}$ is a component of an i.i.d. $N(0, \sigma^2)$ process, an ARMA estimate of it versus an ARMA estimate of its re-sampled version should return similar results that are effectively estimates of the mean. Then singular values of the reconstructed $\hat{\mathbf{X}}$ and $\mathbf{X}^*$ data sets would resemble those from the true data $\mathbf{X}$.

Alternatively, if there are indeed $K_0 > K$ factors, then re-sampling does affect the reconstruction. ARMA prediction of $\mathbf{b}_{K+1}$ will provide meaningful dynamic estimates of the true underlying $\boldsymbol{\beta}_{K+1}$ process. However, a re-sampled $\boldsymbol{\epsilon}^*$ destroys the dynamic dependence, and we would expect only the first K singular values of $\mathbf{X}^*$ to be large, as opposed to the $K+1$ singular values of $\mathbf{X}$ and $\hat{\mathbf{X}}$ being large.

By repeatedly re-sampling $\hat{\boldsymbol{\epsilon}}$ we can form an empirical distribution based on the resulting singular values from each $\mathbf{X}^*$ data matrix. Again we do this sequentially to test the hypotheses of $K_0 < K + 1$ factors for $K = 1, 2, 3, \ldots, m - 1$ until we are no longer able to reject the hypothesis. To do this, against the bootstrapped distribution, we compare the $(K + 1)$st singular value from $\hat{\mathbf{X}}$ by way of the percent variance:

$$\iota^{(K+1)} \equiv \frac{d_{K+1}^2}{\sum_{j=1}^{m} d_j^2}.$$

We perform this method for 100 simulated data sets for three values of σ and 200 bootstraps per simulated data set; all under the simulation setup in Figure 6.2. For the $K_0 = 3$ model, Figure 6.6 shows an example scree plot of $\iota^{(k)}$ from one of the simulated data sets for testing

$K = 1, 2, 3$ and 4 factors. Plotted are the singular values in terms of $\iota^{(k)}$ of $\mathbf{X}$ (red), $\hat{\mathbf{X}}$ (green), and the average over 200 bootstraps for $\mathbf{X}^*$ (blue). In the first panel, for testing $K_0 < 2$ factors there is a clear difference in $\iota^{(2)}$ between $\hat{\mathbf{X}}$ and $\mathbf{X}^*$. This motivates a subsequent test for no more than 2 factors, shown in panel 2. Again there is a clear discrepancy in $\iota^{(3)}$ between $\hat{\mathbf{X}}$ and $\mathbf{X}^*$, which in turn motivates a test for 3 factors, or $K_0 < 4$. This is shown in the third panel. Here, estimation and resampling of a hypothetical 4th factor does not affect the values of $\iota^{(4)}$ in either $\hat{\mathbf{X}}$ or $\mathbf{X}^*$. For completeness, we test for 4 factors, although the failure to reject the previous hypothesis of fewer than 4 factors implies this result. To confirm this, $\iota^{(5)}$ between $\hat{\mathbf{X}}$ and $\mathbf{X}^*$ is shown in the fourth panel.

Figure 6.6: Example "scree" plots for bootstrap approach. For a single simulated data set we examine singular values in terms of % of variance explained for the true data (red), the $(K+1)$st left singular vector and value replaced with AR(2) estimate (green), and mean over 200 bootstraps of resampled $(K+1)$st left singular vector and value replaced with AR(2) estimate (blue). For true $K = 3$ factors, resample of second and third left singular vector and value affects singular values.

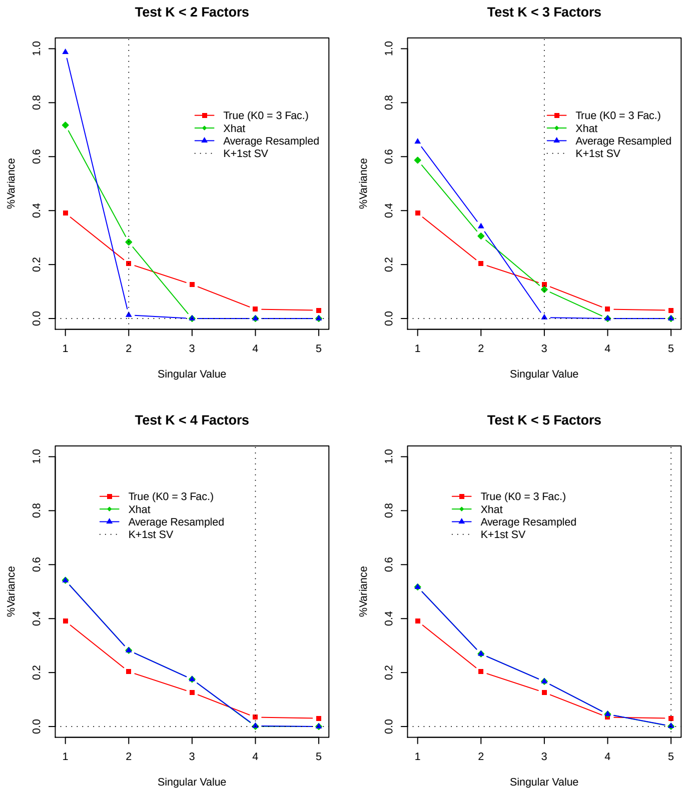


Figure 6.7 summarizes the results for the 3×100 simulated data sets. The top 3 boxplots depict the empirical p-values from the bootstrap distribution for each value of σ for each test of K_0 less than $K+1 = 2, 3, 4$ and 5 factors. For the first two tests, in nearly all cases $\iota^{(\cdot)}$ is so large for $\hat{\mathbf{X}}$ compared with the 200 values for $\mathbf{X}^*$ that the p-value is essentially 0. In the case of testing $K_0 < 4$ factors, the p-value is distributed across the $[0, 1]$ range indicating similar $\iota^{(4)}$ values for both $\hat{\mathbf{X}}$ and $\mathbf{X}^*$. Finally, for testing the hypothesis of 4 factors, (or $K_0 < 5$), we have a similar result to confirm the 3 factor model.

This evidence of a useful test is almost too convincing, so for further illustration, presented directly below each test boxplot are boxplots showing the range of values of $\iota^{(\cdot)}$ for $\hat{\mathbf{X}}$ in green, and the range of the averages over 200 bootstraps for $\mathbf{X}^*$ in blue. The first two plots confirm the result of zero p-values for tests of fewer than 2 and 3 factors ($\iota^{(2)}$ and $\iota^{(3)}$,respectively); the third and fourth illustrate the large overlap in $\iota^{(4)}$ and $\iota^{(5)}$ for testing fewer than 4 and 5 factors.

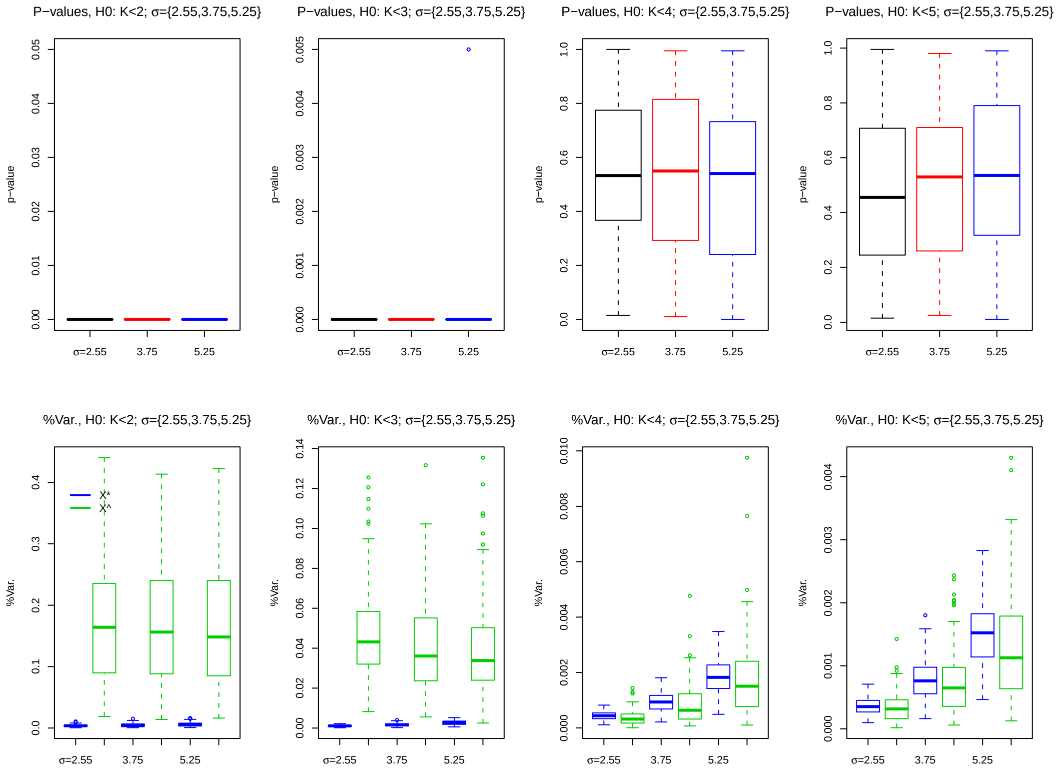
Thus, based on these results, we have identified a useful tool in Section 6.3.1 for developing an initial guess as to the number of factors K, and here we have found an implementable procedure for more rigorously determining this.

6.3.3 Additional Methods

Canonical Autocorrelation

In addition to the ad hoc methods described in Pena and Box (1987) for determining the number of factors for a traditional DFM, a more rigorous test procedure for determination of K is presented in Pena and Poncela (2006). Although their emphasis is on a method that can be used for unit-root non-stationary factors, the method is just as easily applied to a stationary DFM. The idea is that the number of of non-zero canonical correlations between $\mathbf{x}_{i-l}$ and $\mathbf{x}_i$ is equal to the number of factors K. This is based on the relationship between $\mathbf{\Gamma}_{\mathbf{x}}(l)$ and $\mathbf{\Gamma}_{\boldsymbol{\beta}}(l)$ presented in Equation (6.2). Presuming the dynamic factors have mean 0 (for notational

Figure 6.7: P-values and %Variance for Hypothesis Tests for the simulated data of 3 dynamic factors. Top: We reject the hypotheses of fewer than 2 and 3 factors, but are unable to reject fewer than 4 (and thus 5) factors. Bottom: Boxplots of $\iota^{(\cdot)}$ corresponding to each test over all simulations for each choice of σ. In blue, the average over 200 bootstraps for $\mathbf{X}^*$; in green the values for $\hat{\mathbf{X}}$. The first two plots confirm the result of zero p-values for tests of fewer than 2 and 3 factors ($\iota^{(2)}$ and $\iota^{(3)}$,respectively); the third and fourth illustrate the large overlap in $\iota^{(4)}$ and $\iota^{(5)}$ for testing fewer than 4 and 5 factors.



simplicity), the squared sample canonical correlations are given by the eigenvalues of the matrix

$$\hat{\mathbf{M}}(l) = \left[\sum_{i=l+1}^{n} \mathbf{x}_i \mathbf{x}_i'\right]^{-1} \sum_{i=l+1}^{n} \mathbf{x}_i \mathbf{x}_{i-l}' \left[\sum_{i=l+1}^{n} \mathbf{x}_{i-l} \mathbf{x}_{i-l}'\right]^{-1} \sum_{i=l+1}^{n} \mathbf{x}_{i-l} \mathbf{x}_i', \tag{6.6}$$

for lags $l = 1, 2, \ldots$. Theorem 3 in Pena and Poncela (2006) cites that $m - K$ of these eigenvalues converge in probability to zero. For each l, denote the ordered eigenvalues of $\hat{\mathbf{M}}(l)$ as $e_{1;l} \leq e_{2;l} \leq \ldots \leq e_{m;l}$; then Lemma 1 states that the test statistic

$$S_{m-k} = -(n-l) \sum_{j=1}^{m-k} \ln\left(1 - e_{j;l}\right) \tag{6.7}$$

is asymptotically $\chi^2_{(m-k)^2}$.

Based on this statistic, a hypothesis test can be formed for the maximum number of factors. Since rejecting the hypothesis of $\mathrm{H}_0 : k \leq K$ factors implies rejection of the hypothesis $\mathrm{H}_0 : k - 1 \leq K$ for $k - 1 \geq 0$ factors, it makes sense to apply the test sequentially for increasing values of k up to the first failure to reject. This is done for each of a a reasonable number of lags $l = 1, \ldots, q$ (Pena and Poncela, 2006, use 5).

However, based on simulation, here is an instance of where a discrete method does not translate well to a functional DFM setting. In the traditional DFM framework, the idea is to model a "large" number of time series that share some relation using a smaller set of latent factors; for example, using 2 dynamic factors to explain 6 time series. In the case of functional time series, two observed points x_{ij} and $x_{ij'}$; $j, j' = 1, \ldots, m$ are related through a function. We have a high dimension m of observed time series that are less a set of disparate time series than they are sequence of points at each i that are intimately connected to each other as a specific function evaluated at certain points.

Indeed, in the examples below, we see that as the sampling size m of points along the domain $\mathcal{T}$ of $x_i(t)$ grows large (or sampling grows dense), so increase the number of canonical correlations between some $\mathbf{x}_i$ and $\mathbf{x}_{i-l}$ for lags $l = 1, 2, \ldots$. Further, as the $\{x_i(t_j)\}$ are functionally related, the canonical correlations remain large enough to confound test statistic $S_{m-k} = -(n-l) \sum_{j=1}^{m-k} \ln\left(1 - e_{j;l}\right)$. S_{m-k} will increase with m as long as the squared canonical correlations are greater than zero. This is to be expected with functionally related data having a high frequency of measurement. Therefore, when m is large enough, the value for S_{m-k} will almost always reject the null hypothesis, regardless of the true number of factors.

Using the design outline in Figure 6.2, for 100 simulated data sets we plot the mean of the jth squared canonical correlations in Figure 6.8 for 4 choices of m:

$m = 3$	$j = 1, 9, 17$
$m = 5$	$j = 1, 5, 9, 13, 17$
$m = 9$	$j = 1, 3, 5, 7, 9, 11, 13, 15, 17$
$m = 17$	$j = 1, \dots, 17$

According to Pena and Poncela (2006), per lag we expect $K = 3$ large squared canonical correlations, with remaining $j > K$ being near zero. In the first two cases, where m is small (3 and 5), we do see a decay in squared correlations (there are only 3 canonical correlations in the first case). However, for $m = 9$, or 17, although we do see a noticeable decline in squared canonical correlation (SCC) from the Kth to the $(K + 1)$st, the remaining $m - K$ SCCs are still much larger than zero.

To ensure this result is not endemic to this particular simulation, we replicate this method of the simulation design of Bathia et al. (2010) (see next section) in Figure 6.9 and observe the same pattern. Although the reason for the problem in this discrete method when translated to a functional setting is intuitively plausible, an area of future research is to derive this result rigorously.

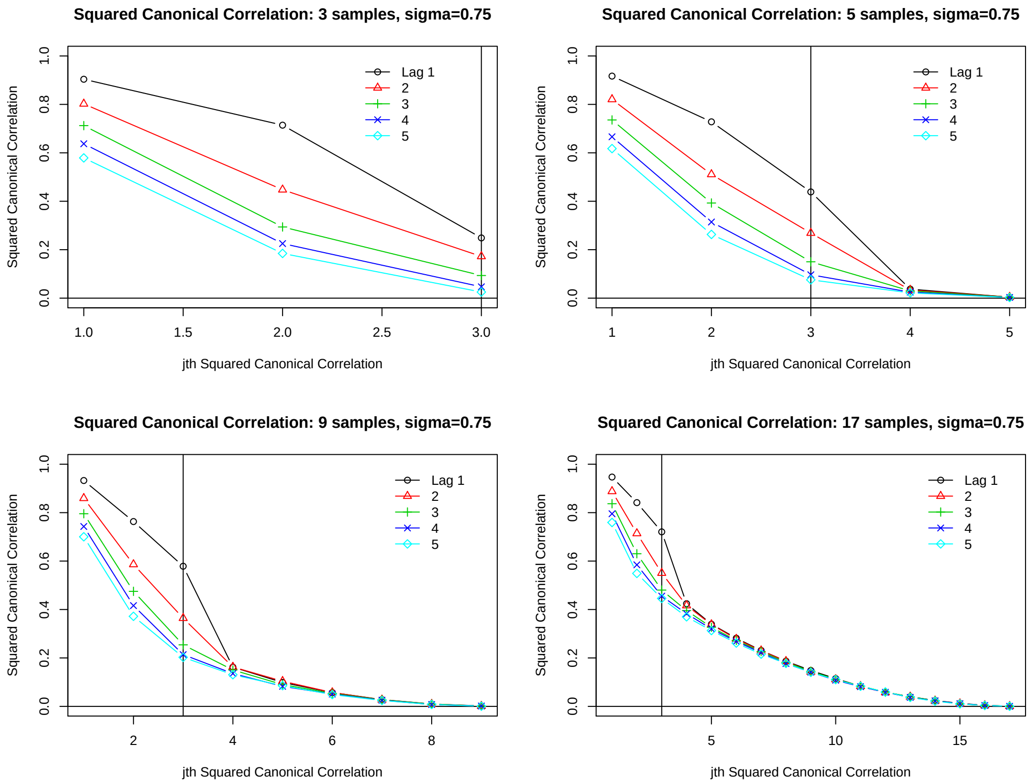
Functional Results

Bathia et al. (2010) present a detailed methodology for determining the "Finite Dimensionality" of functional time series. Essentially, the hypothesis is that an observed functional time series consists of the sum of a noise component and "curve component" or underlying smooth curve:

$$x_i(t) = y_i(t) + \epsilon_i(t).$$

The underlying smooth curve $y_i(t)$ can further be decomposed via the Karhunen–Loeve expan-

Figure 6.8: Canonical correlations for increasing number of series considered, by lag. Yield simulation design: $K = 3$ factors. As m increases – equivalent to a denser sampling – the decay in squared canonical correlations is more persistent. This inflates the test statistic of Pena and Poncela (2006), making rejection rare even when the number of factors being tested well exceeds the true number K.



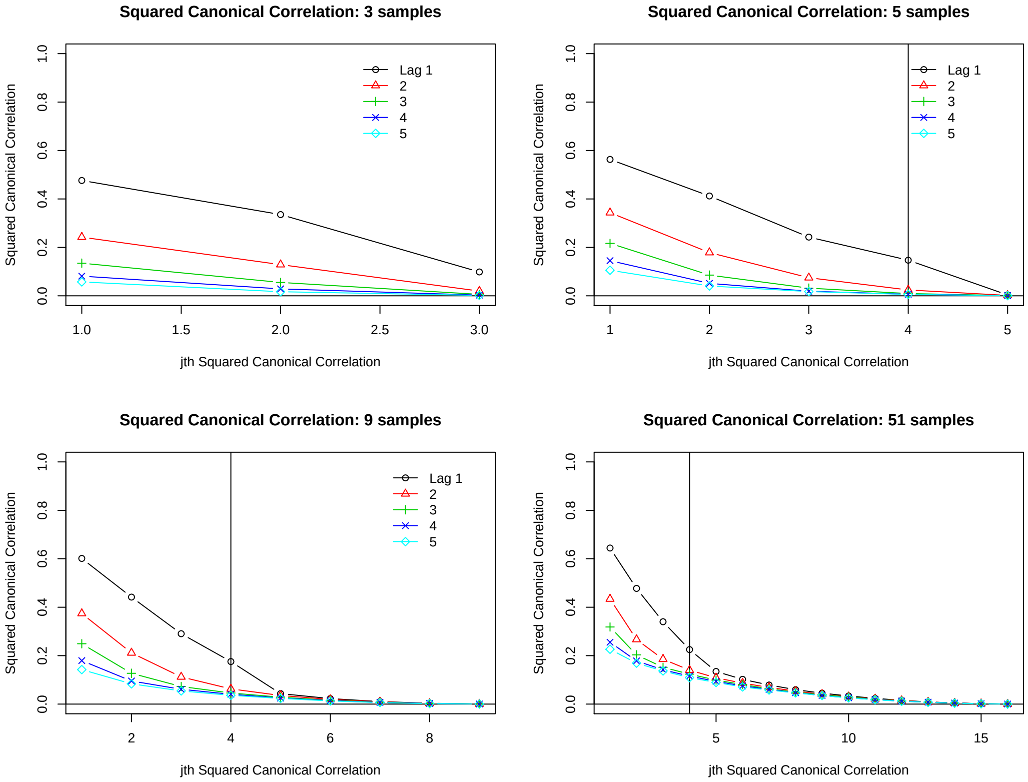
Squared Canonical Correlation: 3 samples, sigma=0.75

Squared Canonical Correlation: 5 samples, sigma=0.75

Squared Canonical Correlation: 9 samples, sigma=0.75

Squared Canonical Correlation: 17 samples, sigma=0.75

Figure 6.9: Canonical correlations for increasing number of series considered, by lag. Bathia et al. (2010) simulation design: $K = 4$ factors. We see the same pattern of slow decay in SCCs when m, the rate of sampling of a functional time series, grows large.



sion to an infinite sum of scalar random variables. The idea is that for some $K \ll \infty$, the variance of the $(K+1)$st random variable is 0, thus identifying a finite dimensionality of the functional time series. This is almost entirely consistent with the current setting of our FDFM, and thus relates well to the determination of the number of dynamic factors and corresponding factor loading curves.

Because the underlying smooth curves $y_i(t)$ are unobserved, Bathia et al. (2010) form an auto-covariance operator based on the observed curve time series $x_i(t)$. The complexity that arises is that for the lag 0 auto-covariance operator, a Karhunen–Loeve expansion is not directly applicable due to the presence of the noise term $\epsilon_i(t)$ (for lags $l > 0$, the noise term is assumed as $E[\epsilon_i(t)\epsilon_{i+l}(t)] = 0$). To get around this complication, Bathia et al. (2010) form another operator that incorporates the auto-covariance functions for all lags l including zero. Eigenfunction decomposition then reveals that the first sharp decline or appearance of a zero corresponding eigenvalue determines the dimensionality K.

To obviate determining eigenfunctions of their operator only to find these eigenvalues, the authors show that the same eigenvalues can be determined by an eigen-decomposition of discrete matrix with elements that are inner-products of the observed functional time series at various lags. The corresponding eigenvectors combined with the observed functional time series then determine the dynamic factors.

Bathia et al. (2010) propose a bootstrap approach of re-sampling residuals from the estimated model to form an empirical distribution of eigenvalues by which to compare with those obtained from the estimated model. Despite the obvious appeal of the approach, in both the simulation design discussed here in Section 6.2 and the design illustrated in Bathia et al. (2010) itself, we were not able to confirm the practical implementation of the method. The findings were not unlike the canonical correlation method of Pena and Poncela (2006); we observed a slow decay of eigenvalues making it difficult to infer the true number of factors.

Error Resampling

Drawing on some of the ideas presented in Bathia et al. (2010) and Pena and Box (1987) another attempt was to combine concepts from each of these. Consider observed data $\mathbf{X}$ which is the sum of a random component $\boldsymbol{\epsilon}$ and dynamic factors with factor loading curves evaluated at $\{t_1, \ldots, t_m\}$:

$$\begin{aligned}\mathbf{X} &= \mathbf{Y} + \boldsymbol{\epsilon} \\ \mathbf{Y} &= \sum_{k=1}^{K} \boldsymbol{\beta}_k \mathbf{f}'_k.\end{aligned}$$

To test the number of factors, consider an SVD based approach. To develop this idea, we continue to use simulated data from the design outlined in Figure 6.2 for $K = 3$ dynamic factors following independent AR(2) processes. Although in practice we will not have the $\mathbf{Y}$ data, it stands to reason that if a method does not work for $\mathbf{Y}$, it most likely will not work for $\mathbf{X}$.

Y Models

Model Y1: Let $\hat{\mathbf{Y}}_{y1}$ be the first $K-1$ true factors and factor loading curves:

$$\begin{aligned}\hat{\mathbf{Y}}_{y1} &= \sum_{k=1}^{K-1} \boldsymbol{\beta}_k \mathbf{f}'_k, \\ \hat{\boldsymbol{\epsilon}}_{y1} &= \mathbf{Y} - \hat{\mathbf{Y}}_{y1} = \boldsymbol{\beta}_K \mathbf{f}'_K.\end{aligned}$$

We sample $\hat{\boldsymbol{\epsilon}}_{y1}$ with replacement to get $\boldsymbol{\epsilon}^*_{y1} = \boldsymbol{\beta}^*_K \mathbf{f}'_K$ and $\mathbf{Y}^*_{y1} = \sum_{k=1}^{K-1} \boldsymbol{\beta}_k \mathbf{f}'_k + \boldsymbol{\beta}^*_K \mathbf{f}'_K$, where $\boldsymbol{\beta}^*_K$ are the sampled (with replacement) elements of $\boldsymbol{\beta}_K$. We can compare SVD($\mathbf{Y}$) with SVD($\mathbf{Y}^*_{y1}$) to examine if there is a detectable difference in the Kth singular value which could indicate a practical method for testing the number of factors.

Model Y2: Suppose that $\mathbf{Y}$ is (somehow) observable, but that the component factors and factor loading curves are not. As a proxy for these, we examine the singular values and singular vectors:

$$\begin{aligned}
\mathbf{Y} &\overset{\text{SVD}}{=} \sum_{j=1}^{m} d_j \mathbf{u}_j \mathbf{v}_j', \\
\hat{\mathbf{Y}}_{y2} &= \sum_{k=1}^{K-1} d_k \mathbf{u}_k \mathbf{v}_k', \\
\hat{\boldsymbol{\epsilon}}_{y2} &= \mathbf{Y} - \hat{\mathbf{Y}}_{y2}.
\end{aligned}$$

We then sample $\hat{\boldsymbol{\epsilon}}_{y2}$ with replacement to get $\boldsymbol{\epsilon}^*_{y2}$ and $\mathbf{Y}^*_{y2} = \hat{\mathbf{Y}}_{y2} + \boldsymbol{\epsilon}^*_{y2}$. In a similar manner, we compare SVD($\mathbf{Y}$) with SVD($\mathbf{Y}^*_{y2}$) and SVD($\mathbf{Y}^*_{y1}$).

In any of these cases – SVD($\mathbf{Y}$), SVD($\mathbf{Y}^*_{y2}$) or SVD($\mathbf{Y}^*_{y1}$) – we still expect to see exactly 3 factors, but with differing singular values d_k from SVD, particularly for the Kth factor. This is summarized by the percentage of variance:

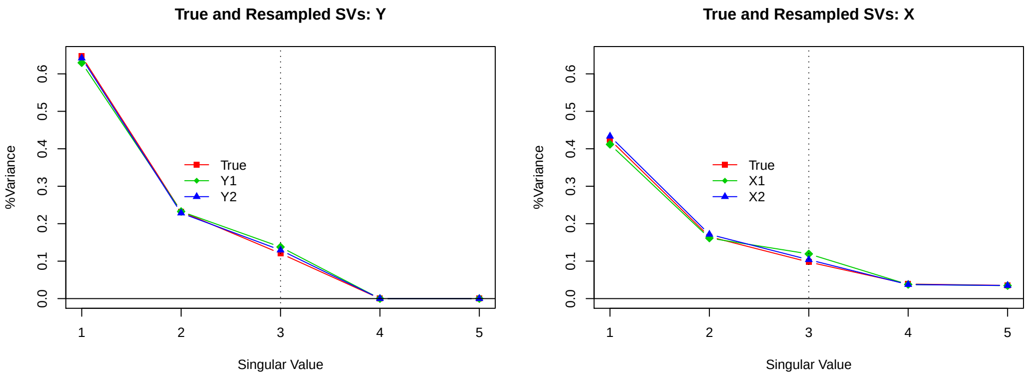
$$\iota^{(k)} \equiv \frac{d_k^2}{\sum_{j=1}^{m} d_j^2},$$

denoted as $\iota^{(k)}_{y0}$, $\iota^{(k)}_{y1}$, and $\iota^{(k)}_{y2}$ for the true, Model Y1, and Model Y2 decompositions, respectively. Because each method still supports a K factor model, it leaves to question what can be inferred from the differing values of $\{\iota^{(k)}_{y0}, \iota^{(k)}_{y1}, \iota^{(k)}_{y2}\}$ for the first K singular values. In fact, the re-sampling of $\hat{\boldsymbol{\epsilon}}_{y\cdot}$ is effectively a re-sampling of $\boldsymbol{\beta}_K$ or some estimate thereof. The AR(p) time dependency is then destroyed, but this has little to no effect on a method (SVD) that is indifferent to the ordering of the values ($i = 1, \ldots, n$) of the factors. An example of this is shown in in the first panel of Figure 6.10.

X Models

By exploring similar methodologies for the "noisy" data $\mathbf{X} = \mathbf{Y} + \boldsymbol{\epsilon}$, we should expect to see more pronounced differences in the $\iota^{(k)}$ resulting from re-sampled residuals under the assumption of $K-1$ factors. Ideally, a re-sampled data set $\mathbf{X}^*$ will result in a markedly lower $\iota^{(K)}_{x\cdot}$ than the true $\iota^{(K)}_{x0}$, and a markedly larger $\iota^{(K+1)}_{x\cdot}$ than the true $\iota^{(K+1)}_{x0}$, which should be close to zero.

Figure 6.10: For simulated data of $K = 3$ factors, we examine the third singular value of the true data (red) and two reconstructed data sets for both the raw $\mathbf{X}$ data and the $\boldsymbol{\epsilon}$ error free $\mathbf{Y}$ In each case, we re-sample either the third factor (green) or left singular vector and value, then reconstruct the data. Though the dynamic dependence of the third factor is destroyed in either case, this does not affect the singular values in the reconstructed data sets.



Consider two different representations of the $\mathbf{X}$ data matrix:

$$\begin{aligned}\mathbf{X} &= \sum_{k=1}^{K} \boldsymbol{\beta}_k \mathbf{f}'_k + \boldsymbol{\epsilon}, \\ \mathbf{X} &\overset{\text{SVD}}{=} \sum_{j=1}^{m} d_j \mathbf{u}_j \mathbf{v}'_j.\end{aligned}$$

Because of error $\boldsymbol{\epsilon}$ we can still expect a sharp decline in $\iota_{x0}^{(K+1)}$ from $\iota_{x0}^{(K)}$, but not necessarily to the point where $\iota_{x0}^{(K+1)} \approx 0$. Therefore, there should exist a greater effect to re-sampling residuals from $K-1$ factor models to an extent that distinguishes them from the true K factor model.

Model X1: Let

$$\begin{aligned}\hat{\mathbf{X}}_{x1} &= \sum_{k=1}^{K-1} \boldsymbol{\beta}_k \mathbf{f}'_k + \boldsymbol{\epsilon}, \\ \hat{\boldsymbol{\epsilon}}_{x1} &= \mathbf{X} - \hat{\mathbf{X}}_{x1} = \boldsymbol{\beta}_K \mathbf{f}'_K.\end{aligned}$$

We sample $\hat{\boldsymbol{\epsilon}}_{x1}$ with replacement to get $\boldsymbol{\epsilon}^*_{x1} = \boldsymbol{\beta}^*_K \mathbf{f}'_K$ and $\mathbf{X}^*_{x1} = \sum_{k=1}^{K-1} \boldsymbol{\beta}_k \mathbf{f}'_k + \boldsymbol{\beta}^*_K \mathbf{f}'_K + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}^*_K$ are the sampled (with replacement) elements of $\boldsymbol{\beta}_K$. Next we compare SVD($\mathbf{X}$) with

SVD($\mathbf{X}^*_{x1}$).

Model X2: Similarly, we can approximate the factors and factor loading curves with singular values and singular vectors:

$$\begin{aligned}
\mathbf{X} &\overset{\text{SVD}}{=} \sum_{j=1}^{m} d_j \mathbf{u}_j \mathbf{v}_j', \\
\hat{\mathbf{X}}_{x2} &= \sum_{k=1}^{K-1} d_k \mathbf{u}_k \mathbf{v}_k', \\
\hat{\boldsymbol{\epsilon}}_{x2} &= \mathbf{X} - \hat{\mathbf{X}}_{x2} = \sum_{k=1}^{K-1} [\boldsymbol{\beta}_k \mathbf{f}_k' - d_k \mathbf{u}_k \mathbf{v}_k'] + \boldsymbol{\beta}_K \mathbf{f}_K' + \boldsymbol{\epsilon} \\
&\approx \boldsymbol{\beta}_K \mathbf{f}_K' + \boldsymbol{\epsilon}.
\end{aligned}$$

Again, we sample $\hat{\boldsymbol{\epsilon}}_{x2}$ with replacement to get $\boldsymbol{\epsilon}^*_{x2}$ and $\mathbf{X}^*_{x2} = \hat{\mathbf{X}}_{x2} + \boldsymbol{\epsilon}^*_{x2}$, then compare SVD($\mathbf{X}$) with SVD($\mathbf{X}^*_{x2}$) and SVD($\mathbf{X}^*_{x1}$). An example of this process is depicted in the second panel of Figure 6.10. Again, we see little difference in singular values. Although resample affects the dynamic dependence of the 3rd factor, the SVD of the reconstructed $\mathbf{X}$ matrices is indifferent to this in terms of the number of factors and the proportion of variance explained by them. This is in fact what motivated the proposed approach in Section 6.3.2.

6.4 Order Selection

To select the order r of the auto-regressive processes for the dynamic factors, we employ the Akaike Information Criterion (AIC) and Schwartz's Bayesian Information Criterion (BIC). Because the order selection is only for the determination of the specification for the dynamic factors, we need not employ the entire penalized conditional log-likelihood $l_p(\mathbf{X}, \mathbf{B})$ from Equation 4.7. Rather, we focus solely on the conditional likelihood $l(\mathbf{B})$ for the dynamic factors given in Equation 4.5:

$$\begin{aligned}
\text{AIC}_p &= l^{(p)}(\mathbf{B}_{(p)}) + 2p, \qquad (6.8)\\
\text{BIC}_p &= l^{(p)}(\mathbf{B}_{(p)}) + 2\ln{(n-p)},
\end{aligned}$$

where $l^{(p)}(\mathbf{B}_{(p)})$ is the likelihood expression for the dynamic factors from Equation 4.5 evaluated at the MLEs for dynamic factors given by

$$\beta_{ik} = c_k + \sum_{l=1}^{p} \varphi_{i-l,k}\beta_{i-l,k} + v_{ik},$$

for factors $k = 1, \ldots, K$ and $p = 1, 2, \ldots$. We evaluate the criteria in Equation (6.8) based on E-step values for $E[\boldsymbol{\beta}|\boldsymbol{X}]$ from the final EM iteration of FDFM estimation. The value of p that results in the lowest of either AIC or BIC is chosen as the order of auto-regressive process for the dynamic factors. It is possible that the dynamic factors follow differing AR processes AR(p_k) or even ARMA processes ARMA(p_k, q_k), and it is entirely possible to apply AIC and BIC for reasonable combinations of p_k, q_k. However, for the purposes of the current FDFM, we maintain the convention that all of the dynamic factors follow the same order of AR process.

6.5 Inference

To create confidence intervals for FDFM parameters, factor loading curves, and forecasts, we rely on a bootstrap method to form empirical distributions for each of these based on resampled model residuals. The following method and some notation is based on the method described in Alonso et al. (2011). After FDFM estimation, we have the predicted functional time series

$$\hat{x}_i(t_j) = \sum_{k=1}^{K} \beta_{ik}\hat{f}_k(t_j),$$

where β_{ik} is understood to be $E[\beta_{ik}|\boldsymbol{X}]$ from the E-step of the final EM iteration of estimation.

To formulate the bootstrap procedure, we revert to the traditional DFM notation – used most recently in Section 6.3.1; Equations (6.1):

$$\begin{aligned} \mathbf{x}_i &= \mathbf{F}\boldsymbol{\beta}_i + \boldsymbol{\epsilon}_i, \\ \Phi(L)\boldsymbol{\beta}_i &= \mathbf{c} + \mathbf{v}_i, \end{aligned}$$

with $\mathbf{x}_i = [x_i(t_1), \ldots, x_i(t_1)]'$; $\boldsymbol{\beta}_i = [\beta_{i1}, \ldots, \beta_{iK}]'$ and $\mathbf{c} = [c_1, \ldots, c_K]'$; and $\Phi(L)$ is the lag polynomial from Equation (6.1).

We define the model residuals as:

$$\begin{aligned}\hat{\boldsymbol{\epsilon}}_i &= \mathbf{x}_i - \hat{\mathbf{x}}_i = \mathbf{x}_i - \hat{\mathbf{F}}\boldsymbol{\beta}_i, \\ \hat{\mathbf{v}}_i &= \boldsymbol{\beta}_i - \hat{\mathbf{c}} - \sum_{r=1}^{p} \hat{\Phi}_r \boldsymbol{\beta}_{i-r}.\end{aligned} \tag{6.9}$$

We draw *iid* resamples from $\hat{\boldsymbol{\epsilon}}_i = [\hat{\epsilon}_{i1}, \ldots, \hat{\epsilon}_{im}]'$ to form ϵ^*_{ij} for each $j = 1, \ldots, m$. Similarly, we draw *iid* resamples from $\hat{\mathbf{v}}_i = [\hat{v}_{i1}, \ldots, \hat{v}_{iK}]'$ to form v^*_{ik} for each $k = 1, \ldots, K$.

The next step is to generate bootstrap data based on the resampled residuals:

$$\begin{aligned}\boldsymbol{\beta}^*_i &= \hat{\mathbf{c}} + \sum_{r=1}^{p} \hat{\Phi}_r \boldsymbol{\beta}^*_{i-r} + \mathbf{v}^*_i, \\ \mathbf{x}^*_i &= \hat{\mathbf{F}}\boldsymbol{\beta}^*_i + \boldsymbol{\epsilon}^*_i,\end{aligned}$$

with $\boldsymbol{\epsilon}^*_i = [\epsilon^*_{i1}, \ldots, \epsilon^*_{im}]'$ and $\mathbf{v}^*_i = [v^*_{i1}, \ldots, v^*_{iK}]'$. We resample and replicate the data B times. At this point, we now have three sets of model parameters and factor loading curves – the true values, the FDFM estimates and resampled FDFM estimates:

$$\begin{aligned}\Theta &= \left\{\sigma^2 \cup \bigcup_{k=1}^{K} \{\mathbf{f}_k, \sigma^2_k, \mathbf{c}, \Phi_1, \ldots, \Phi_p\}\right\}, \\ \hat{\Theta} &= \left\{\hat{\sigma}^2 \cup \bigcup_{k=1}^{K} \{\hat{\mathbf{f}}_k, \hat{\sigma}^2_k, \hat{\mathbf{c}}, \hat{\Phi}_1, \ldots, \hat{\Phi}_p\}\right\}, \\ \Theta^* &= \left\{\sigma^2_* \cup \bigcup_{k=1}^{K} \{\mathbf{f}^*_k, \sigma^2_{k;*}, \mathbf{c}^*, \Phi^*_1, \ldots, \Phi^*_p\}\right\}.\end{aligned} \tag{6.10}$$

In the lattermost case we have the number of bootstrap replicates B of Θ^* by which to form a set of empirical distributions we denote as F_{Θ^*}. Based on percentiles of these, we can create confidence intervals or bands/functions for $\hat{\Theta}$, and – in the case of simulated data – compare the confidence bands and estimates with the true values Θ.

In a similar manner, we implement a bootstrap procedure for forecast intervals. For forecast horizon h, using the notation of this section we generate forecasts from FDFM estimates by:

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{n+h} &= \hat{\mathbf{c}} + \sum_{r=1}^{p} \hat{\Phi}_r \hat{\boldsymbol{\beta}}_{n+h-r}, \\ \hat{\mathbf{x}}_{n+h} &= \hat{\mathbf{F}} \hat{\boldsymbol{\beta}}_{n+h}. \end{aligned}$$

Likewise, from our resampled datasets and resulting estimates Θ^*, we create a distribution of forecasts based on

$$\begin{aligned} \boldsymbol{\beta}^*_{n+h} &= \mathbf{c}^* + \sum_{r=1}^{p} \Phi^*_r \boldsymbol{\beta}^*_{n+h-r}, \\ \mathbf{x}^*_{n+h} &= \mathbf{F}^* \boldsymbol{\beta}^*_{n+h}. \end{aligned}$$

for each of the B bootstraps. Thus in addition to the forecasted curves $\hat{\mathbf{x}}_{n+h}$ we can produce forecast intervals based on the bootstrap distribution. Again, with simulated data, we will be able to compare the forecasted curve to the true one, and also examine if the true curve is contained within the forecast interval.

Chapter 7

Yield Curve Application

The yield curve is an instrument for portfolio management and for pricing synthetic or derivative securities (Diebold and Li, 2006). Bond prices are hypothesized to be a function of an underlying continuum of yields as a function of maturity, known as the yield curve. Our contribution to the yield literature is pragmatic: the FDFM reconciles the theory-based desire to model yield data as a curve with the applied need of accurately forecasting that curve over time.

The yield curve is a *theoretical* construct not without its own inherent *practical* difficulties. First and foremost, although yield determines prices, only bond *prices* are observed for a set of discrete maturity horizons; from these a corresponding discrete set of yields are calculated. Thus the yields themselves are not directly observed, nor is an entire curve for every possible maturity. Further, not only is it of interest to know the yield for all maturities at each point in time (cross-sectional), but also for a single maturity as it evolves over time (dynamic). Finally, because a bond at time i of maturity t is essentially the same bond as the one at time $i+1$ of maturity $t-1$, there is also a certain amount of systematic *cross-correlation* in yield data. Therefore, bond data have each of cross-sectional, dynamic and cross-correlational behaviors to consider for predictive modeling.

To this end, yield curve models have traditionally assumed either of two formulations. The first is theoretical in nature: as in Hull and White (1990) and Heath et al. (1992), for a given date the emphasis is on fitting a yield curve to existing yields based on no-arbitrage principles stemming from economic theory. The other approach is the so-called equilibrium or affine-class models where time series techniques are used to model the dynamics of yield on a short term or instantaneous maturity. Yields for longer maturities are then derived using an affine model. This method has been developed in works such as Vasicek (1977), Cox et al. (1985), and Duffie and Kan (1996).

These contrasting methods illustrate the dichotomy of yield forecast models. As a practical matter, goodness of fit is paramount in a model for it to be of any use. Still, a yield model should be consistent with its underlying theory, and maintain a degree of economic interpretation. Cross sectional/no-arbitrage models ignore the dynamics of yields over time (as noted in Diebold and Li, 2006; Koopman et al., 2010, e.g.) and thus threaten the former yet satisfy the latter. Time series/equilibrium models place emphasis on the former at the expense of the latter (as seen in Duffee, 2002).

7.1 Models for Yield Curve Forecasting

What we propose in this chapter is a synthesis of the cross sectional and dynamic considerations mentioned above. We approach yield curves as a *functional* time series; the yields of the observed maturities are a discrete sampling from a true underlying yield *curve*. To this end we conflate concepts from functional data analysis (FDA; Ramsay and Silverman, 2002, 2005) and from dynamic factor analysis/modeling (DFM; Basilevsky, 1994, e.g.). Recall from Chapter 4 that the FDFM's factor loading curves are natural cubic splines (NCS): a significant result which facilitates interpolation of yields both within and out of sample so that forecasts are indeed true yield curves. While the factor loadings account for the cross-sectional/curve dimension of yields, the dynamic factors, in turn, determine the evolution of these functions over time. Thus, they account for the time series and cross-correlational nature of yield data. Our particular specification of the FDFM enables its estimation via the Expectation Maximization algorithm (EM) (Dempster et al., 1977).

Why the need for *both* a functional and a dynamic factor framework? Recall that the unifying goal is to develop a model consistent with the concept of the yield *curve* posited by economic theory and one which is of use for practical forecasting. A naive attempt to merge the latter need with the former is to model yields for all observed maturities over time as a multivariate time series. However, as the number of observed maturities increases to even moderate size, vector autoregressive models (VARs) – for example – become intractable in dimension.

Abstracting from the yield setting for a moment, in a more general sense large multivariate time series have been successfully modeled (Engle and Watson, 1981; Geweke and Singleton, 1981; Molenaar, 1985; Pena and Box, 1987; Pena and Poncela, 2004, to name just a few) using a dynamic factor approach. In DFMs the multivariate data are assumed to be dependent on a small set of unobserved dynamic factors. This solves the dimensionality problem, yet DFMs *per se* leave to question the interpretability of the unobserved factors. Further, in our present context, DFMs fall short of producing a functional yield *curve*.

To incorporate the functional aspect, we propose to combine the DFM framework with ideas from functional data analysis (FDA) (Ramsay and Silverman (2002, 2005) provide a thorough treatment of FDA in both theory and through application). However, FDA in general is an area still nascent in development, and these applications deal primarily with collections of *independent* curves. Earlier work by Besse et al. (2000) applied functional autoregressive models (FAR) to univariate climatological data: the seasonal cycle is hypothesized to be functional. In a similar hypothesis, Shen (2009) forecasted periodic call volume data using a method akin to functional principle component analysis (FPCA). In an applied setting more similar to ours, Hyndman and Shang (2009) developed a weighted FPCA method to forecast time series of curves and applied it to multivariate time series of fertility or mortality data indexed by different ages. Yet, unlike these models where FPCA and time series modeling are performed in separate steps, ours is a method that estimates both functional and time series components *simultaneously*, and does so in a quite natural manner.

Within the context of yield curve forecasting, other recent developments have begun to reconcile the statistical viability of DFMs and functional data analysis with the underlying theory in regard to yield dynamics; a constraint which all but requires the usually absent interpretation for the dynamic factors. Diebold and Li (2006) introduced the Dynamic Nelson–Siegel model (DNS): a three factor DFM with functional coefficients estimated in two steps. The functional coefficients are pre-specified as fixed parametric curves and the authors further provide an economic interpretation of each. Koopman et al. (2010) extended the DNS specification to allow (G)ARCH volatility and a fourth dynamic factor which allows time dependence to the otherwise fixed parametric factor loading curves. Another DFM-type approach is provided by Bowsher and Meeks (2008) which present a cointegrated DFM using natural cubic splines (NCS). Spline knots serve as dynamic factors following an error correction model process; the knot locations are determined via an initial selection procedure.

It is worth noting our FDFM is in a similar vein as those of the aforementioned yield models: a dynamic factor model with functional coefficients; one which– quite coincidentally– even exploits the properties of NCS for the cross-sectional/curve dimension of yields. However, unlike Diebold and Li (2006), the FDFM functional coefficients are estimated; thus, they are free to vary with the particular application to explain the functional nature of the data. Further, as opposed to either class of models, estimation of the FDFM is achieved in a single step. Within the yield context it will be seen that the FDFM satisfies our two aforementioned criteria: goodness of it and economic interpretability. That the factor loading curves are estimated facilitates application of the FDFM to contexts outside of yield curve forecasting as well. We will show through simulation that our specification even permits the inclusion of observed non-latent variables in the dynamic factors similar to Diebold et al. (2006).

Presented in this chapter is our functional dynamic factor model (FDFM) which we show to perform very well in regard to yield curve forecasting. Further, we do so in multiple assessments which highlight the model's capability of accurately forecasting the entire function as well as the potential profit generated from employing these forecasts in trading strategies. In any sense the FDFM outperforms existing models which require either multiple-step estimation or lack a functional component.

7.2 Application to Yield Curve Data

7.2.1 Yield Curve Data

In this section we consider the application of our functional dynamic factor model to actual yield data. We use the same data as Diebold and Li (2006)[1] which is a sample of monthly yields on zero coupon bonds of eighteen different maturities (in months):

$$t_j = \{1.5, 3, 6, 9, 12, 15, 18, 21, 24, 30, 36, 48, 60, 72, 84, 96, 108, 120\},$$

$j = 1, \ldots, m = 18$. The yields are from the period January 1985 through December 2000 (192 months) originally obtained from forward rates provided by the Center for Research in Securities Prices (CRSP), then converted to unsmoothed Fama-Bliss yield rates; see Fama and Bliss (1987) for further details on the method.

7.2.2 Candidate Models

As noted by Koopman et al. (2010), in yield curve forecasting there is often a tradeoff between a statistically rigorous model which accounts for both the dynamic and cross sectional behavior of the data, and a theoretical model which is consistent with the tenets of modern Economics in regard to yield curve formulation.

To this end, Diebold and Li (2006) built on the Nelson-Siegel framework (Nelson and Siegel, 1987) by formulating a dynamic version thereof known as the Dynamic Nelson-Siegel model (DNS). It is a three factor model that accounts for the short, mid and long term behavior of the yield curve over time. Extensions of the DNS model such as Diebold et al. (2006) or Koopman et al. (2010) allow state-space specification, inclusion of additional latent or non-latent factors and GARCH errors. In any case, the DNS specification has proven successful in outperforming yield curve forecasts produced from traditional VAR methods. However, it is also somewhat restricted by the fixed parametric form of the functional coefficients and using a two-step estimation procedure.

Another recent approach is that considered by Bowsher and Meeks (2008) known as a func-

[1] See http://www.ssc.upenn.edu/$\tilde{\text{f}}$diebold/papers/paper49/FBFITTED.txt for the actual data.

tional signal plus noise error correction model (FSN-ECM). Here the cross sectional behavior of the yield curve is modeled as an NCS with significantly fewer knot locations than the number of observed maturities. The latent dynamic factors at these knot locations then follow a cointegrated multivariate time series. Despite the model's fit, selection of the knot locations is exhaustive, and is an additional step in the estimation. Further, as noted in Koopman et al. (2010), cointegrated factors present a difficulty in terms of retaining economic interpretation.

In the following sections we compare the FDFM with the DNS model presented in Diebold and Li (2006). Their model is composed of three factors with corresponding factor loading curves. The factor loadings are pre-specified parametric curves (see the dashed curves in Figure 7.1) based on financial economic theory. Let $x_i(t)$ denote the yield at date i on a zero coupon bond of maturity t, then the DNS model is represented as

$$\begin{cases} x_i(t) = \sum_{k=1}^{3} \beta_{i,k} f_k(t) + \epsilon_i(t) \text{ , for } i = 1, \dots, n, \\ f_1(t) \equiv 1 \text{ , } f_2(t) \equiv \frac{1-\exp(-\alpha_i t)}{\alpha_i t} \text{ , } f_3(t) \equiv f_2(2) - \exp(-\alpha_i t), \\ \beta_{i,k} = c_k + \varphi_k \beta_{i-1,k} + \zeta_{i,k} \text{ , for } k = 1, 2, 3, \end{cases} \tag{7.1}$$

evaluated at maturities t_j, $j = 1, \dots, m$. The first loading curve $f_1(t)$ is constant and intended to represent the long term component of yields (level); the second $f_2(t)$ represents a short term component, or slope. Finally, the third loading $f_3(t)$ represents a mid-term component, or curvature. The parameter α_i determines the point $t^*(\alpha_i)$ at which $f_3(t)$ achieves its maximum. While this can be estimated as a fourth factor (see, e.g., Koopman et al., 2010), Diebold and Li (2006) set α_i to a fixed value for all $i = 1, \dots, n$. This results in entirely predetermined, parametric curves. The specific value $\alpha = 0.0609$ is determined by their definition of "mid-term" as $t = 30$ months.

Estimation of the DNS model is a two step procedure. First, time series of factor scores of $\hat{\beta}_{i,k}$ are estimated by ordinary least squares (OLS) of $x_i(t_j)$ on $[1, f_2(t_j), f_3(t_j)]$ for $j = 1, \dots, m$ at each time point $i = 1, \dots, n$. Second, an AR(1) model is fit on each series $\hat{\beta}_{i,k}$ for the purpose of forecasting $\hat{\beta}_{n+1,k}$ and ultimately $\hat{x}_{n+1}(t_j)$ via Equation (4.32) from Section 4.3.3.

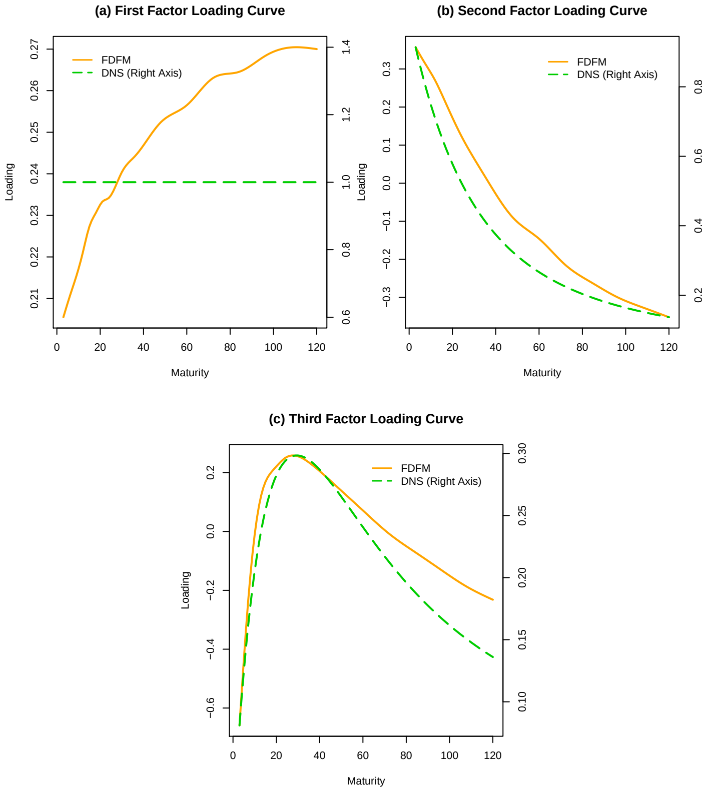


Figure 7.1: Example of factor loading curves: FDFM curves (solid, left axis) estimated from the period May 1985 to April 1994; pre-specified DNS curves (dashed, right axis). FDFM estimates closely resemble the shape of the DNS curves for the second and third factors, while f_1^{FDFM} resembles a typical yield curve shape. Dual axes have been utilized to account for difference in scale.

7.2.3 Assessment

We assess the performance of the FDFM in three distinct exercises. The first two are traditional error based assessments of forecasts or within-sample predictions of yield curves or sections thereof. The final application is a combination of both forecasting and curve synthesis. Through an adaptation of the trading algorithms introduced in Bowsher and Meeks (2008), we develop trading strategies based on the forecasts of the FDFM and DNS models and assess the resulting profit generated by each.

For each of these, as a comparison, we use the DNS specification aforementioned above in Section 7.2.2. For the purpose of making an unbiased comparison, we use a similar formulation of our FDFM model with 3 factors following independent AR(1) processes. The key distinction between this FDFM model and the DNS model is that the FDFM estimates the model simultaneously: the smooth factor loading curves *and* the AR(1) parameters are estimated in a single step. In contrast, the estimation for the DNS model requires two steps given the *prespecified* factor loading curves: first the factor time series are estimated; from these the AR(1) parameters are determined.

The key distinction between the two models raises an interesting question: How do the factor loading curves between the two models compare? Figure 7.1, Panels (a)-(c), show an example of the factor loading curves estimated by the FDFM (solid line) for the period May 1985 through April 1994. Pictured alongside, the dashed line plots the DNS model curves. Recall the DNS motivation for the form of f_1, f_2, and f_3 was an economic argument, while the formulation of the FDFM described in Chapter 2 is based entirely on statistical considerations. Despite this, we see that the FDFM model is flexible enough to adapt to a specific application. Factor loading curves $f_2(t)$ and $f_3(t)$ from the FDFM assume the behavior of those from the DNS model without imposing any constraints that would force this. Thus, the FDFM inherits the economic interpretation of $f_2(t)$ and $f_3(t)$ set forth in Diebold and Li (2006). In the case of $f_1(t)$, the FDFM version resembles a typical yield curve shape as opposed to a constant value for DNS; however, inspecting the magnitude suggests that departure of the FDFM version from a constant value is small. Less typical yield curve shapes are usually characterized by deviations in the short and mid term yields from the norm. This is exactly what $f_2(t)$ and $f_3(t)$ capture.

Thus we consider the first factor as the mean yield, while the second and third account for short and mid term deviations from this norm.

Forecast Error Assessment

In this section, we compare the FDFM and DNS models using a rolling window of 108 months to forecast the yield curve 1, 6, or 12 months ahead[2]. Yields on bonds of maturity less than three months are omitted in order to match the methodology used in Diebold and Li (2006). To compare the models we use the mean forecast error (MFE) and root mean squared forecast error (RMSFE):

$$\text{MFE}_j = \sum_{i=1}^{r} \frac{[x_{n+h}(t_j) - \hat{x}_{n+h}(t_j)]}{r}\ , \ \text{RMSFE}_j = \sqrt{\sum_{i=1}^{r} \frac{[x_{n+h}(t_j) - \hat{x}_{n+h}(t_j)]^2}{r}}.$$

where $r = 84, 79, 73$ is the number of rolling forecasts for forecast horizon $h = 1, 6, 12$, respectively.

A summary of the forecasting performance is shown in Table 7.1. For month ahead forecasts, the MFE is lower (in magnitude) with the FDFM for four out of the five displayed maturities (highlighted in **bold**), while RMSFE is lower for all five. For six months ahead, DNS outperforms FDFM just 2 out of five times in both MFE and RMSFE. For twelve month ahead forecasts, DNS outperforms FDFM in MFE for 3 of 5 displayed maturities. However FDFM has lower RMSE for all 5 maturities.

Curve Synthesis

Because each factor loading curve $\hat{f}_k(\cdot)$ is an NCS, between any two observed maturities t_j and t_{j+1}, we can calculate the value for $\hat{f}_k(t)$: see Equation (4.32) of Section 4.3.3. It follows, then, that between any two time series of yields $\{x_i(t_j)\}_{i=1}^n$ and $\{x_i(t_{j+1})\}_{i=1}^n$, we are able to replicate an entire time series for the intermediate maturity t: $\{\hat{x}_i(t)\}_{i=1}^n$.

To illustrate this point, we use the entire data set (see introduction of Section 7.2), that is, use $i = 1, \ldots, n = 192$ months of yield data for maturities t_j, $m = 18$. For both the

[2] For example, for the one month ahead forecast we fit the models on the first 108 months of data and forecast the 109th month. Then fit the models on the 2nd through 109th month and forecast the 110th month, etc.

Table 7.1: MFE and RMSFE: 1, 6, and 12 month ahead Yield Curve Forecast Results. The better result between the two models is highlighted in **bold**. For 1 month ahead forecasts, the FDFM results in lower (magnitude) MFE for most maturities but results are mixed for 6 and 12 months ahead. RMSFE is typically lower with the FDFM for 1, 6 and 12 months ahead.

	MFE					
	1 Month Ahead		6 Months		12 Months	
Maturity	DNS	FDFM	DNS	FDFM	DNS	FDFM
3 Months	-0.045	**0.026**	**0.123**	0.172	**0.203**	0.257
1 Year	**0.023**	0.035	0.177	**0.168**	0.229	**0.215**
3 Years	-0.056	**0.015**	**0.022**	0.060	**0.003**	0.013
5 Years	-0.091	**-0.004**	-0.079	**-0.021**	-0.166	**-0.133**
10 Years	-0.062	**-0.023**	-0.139	**-0.121**	**-0.316**	-0.318
	RMSFE					
3 Months	0.176	**0.164**	**0.526**	0.535	0.897	**0.867**
1 Year	0.236	**0.233**	**0.703**	0.727	0.998	**0.967**
3 Years	0.279	**0.274**	0.784	**0.775**	1.041	**0.947**
5 Years	0.292	**0.277**	0.799	**0.772**	1.078	**0.953**
10 Years	0.260	**0.250**	0.714	**0.697**	1.018	**0.921**

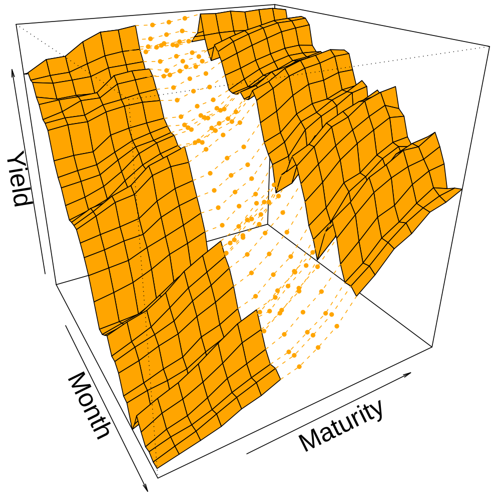


Figure 7.2: Example of Curve Synthesis: Entire time series of yields are omitted from estimation, then "filled in" using the imputation described in Section 4.3.3. Here, 3 consecutive maturities have been omitted, resulting in 3 missing time series corresponding to these maturities.

DNS and FDFM models, we delete a set of adjacent time series from the data, estimate the model, then assess the prediction error of the predicted series in reference to the actual deleted series. Specifically, for our data matrix $\mathbf{X}_{n\times m}$ with columns $\mathbf{x}_1, \ldots, \mathbf{x}_m$, we omit $l = 1, \ldots, L$ consecutive columns from $\mathbf{X}$, then estimate the model on the remaining $Q \equiv m - L$ maturities. From this we compute the L time series of missing data: $\hat{\mathbf{x}}_j, \ldots, \hat{\mathbf{x}}_{j+L}$. For each choice of L, we delete a "horizontally" rolling window of width L maturities and estimate the model on the remaining Q maturities, $R \equiv m - L + 1$ times. As an example, for $L = 3$, we can estimate the models on $\mathbf{x}_4, \ldots, \mathbf{x}_m$ and predict $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}_3}$; then estimate the models on $\mathbf{x}_1, \mathbf{x}_5, \ldots, \mathbf{x}_m$ and predict $\hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}_4}$, etc.

We examine the RMSFE for the lth omitted maturity of the rth sequence; $r = 1, \ldots, R$; $l = 1, \ldots, L$. Because the models are estimated based on a rolling window of maturities, for each choice of L a time series $\mathbf{x}_j$ for yield t_j will be estimated multiple times. Therefore, for each choice of L we take the mean of the RMSFE of the predicted series for each maturity. We further average over our definitions of short ($t \in [1.5, 21)$), mid ($t \in [21, 36]$), and long term ($t \in (36, 120]$) horizons. Finally, we average over all maturities as a one-number summary. These results are presented in Table 7.2 with FDFM as a fraction of DNS. Because prediction for the FDFM model outside the range of the data is linear extrapolation[3], we expect these to become increasingly inaccurate as L grows large. Thus, results are also presented excluding extrapolated predictions in order to better illustrate the truly functional predictions of the FDFM.

In general, as L increases from 1 to 8 we see the expected decline in the performance of the FDFM model relative to DNS. In Panel (a) of Table 7.2 the average RMSFE on short term bonds for the FDFM remains surprisingly robust as we delete more and more maturities. On mid term bonds, DNS results in lower prediction error when the number of deleted series reaches 5 or more. For long term, DNS more or less outperforms FDFM across the board (this trend will be echoed in Section 7.2.3). These results are similar whether or not the extrapolated results are included. Perhaps the best summary is the last column in each of Panel (a) and (b) of Table 7.2, where beyond 3 or 4 omitted maturities, the parametric based DNS model begins

[3] This is due to the NCS framework; see Section 4.3.1 for details.

to outperform the FDFM.

Table 7.2: Average RMSFE; FDFM as a fraction of DNS: (a) with Extrapolation (b) Without Extrapolation

	(a) With Extrapolation				(b) Without Extrapolation			
	Short	Mid	Long	All	Short	Mid	Long	All
Omitted								
1	**0.88**	**0.97**	1.05	**0.95**	**0.84**	**0.97**	1.04	**0.94**
2	**0.95**	**0.90**	1.13	**1.00**	**0.90**	**0.90**	1.01	**0.94**
3	**0.94**	**0.98**	1.06	**0.98**	**1.00**	**0.98**	**1.00**	**0.99**
4	**0.87**	**0.93**	1.64	1.14	**0.99**	**0.93**	1.07	1.01
5	**0.99**	1.01	**0.88**	**0.95**	1.07	1.01	**0.99**	1.03
6	**0.99**	**1.00**	1.67	1.26	1.05	**1.00**	1.20	1.09
7	1.34	1.11	**0.93**	1.13	1.24	1.11	1.16	1.19
8	**0.92**	1.17	1.82	1.39	1.30	1.18	1.48	1.33

Portfolio-Based Assessment

RMSFE-type assessment provides a good diagnostic measure of forecast performance from a statistical perspective. However, as Bowsher and Meeks (2008) argued in their paper, in applied economic settings, a pure error-based assessment measure may fail to fully explain the financial implications of having used a particular model. Therefore, in this section we consider an adaptation of the profit based assessment introduced therein. By using modified versions of their three trading strategies, we create portfolios based on the model forecasts, then measure the cumulative profit of the strategy. This also serves as a good capstone exercise for our presentation of the FDFM as it simultaneously involves both forecasting *and* curve synthesis: the primary uses for our model.

In each strategy we use the same rolling window of 108 months as described in Section 7.2.3 so that the trading algorithm is employed every month over the course of 84 months. Each period i we create a portfolio consisting of a \$1M purchase of one bond or set of bonds and a corresponding sale of another bond or set of bonds for the same amount. Therefore, the net investment per period is \$0. The decision of which bond to sell and which to buy is made based on the sign of the predicted spread in their one period returns.

At time $i + 1$ we cash out our portfolio and record the cumulative profit over the 84 month trading period. Denoting the yield at time i of a zero coupon bond of maturity t months as $x_i(t)$, the price of the bond at time i is

$$P_i(t) = \exp[-tx_i(t)]. \tag{7.2}$$

Correspondingly, the price the next period (month) is then $P_{i+1}(t-1) = \exp[-(t-1)x_{i+1}(t-1)]$ since in the month that has elapsed the maturity is reduced by, not surprisingly, one month. We denote the one period return as

$$R_{i+1}(t) = \left[\frac{P_{i+1}(t-1)}{P_i(t)}\right] - 1, \tag{7.3}$$

and the log one period return as $r_{i+1}(t) \equiv \ln[1 + R_{i+1}(t)]$. Equations (7.2) and (7.3) imply

$$r_{i+1}(t) = tx_i(t) - (t-1)x_{i+1}(t-1). \tag{7.4}$$

Thus for a forecasted yield $\hat{x}_{i+1|i}(t)$ we have $\hat{r}_{i+1|i}(t) = tx_i(t) - (t-1)\hat{x}_{i+1|i}(t-1)$, which is a combination of both actual and forecasted yields. We use the data presented in Section 7.2.1 and thus are limited to a set of non-consecutive observed maturities. Akin to Bowsher and Meeks (2008), we rely on linear interpolation of $x_i(t-1)$ to provide the yield for $x_i(t)$ and use the same random walk forecast (RW) as a benchmark by which to compare models:

$$x_{i+1}(t) = x_i(t) + \eta_{i+1}(t)\ ,\ \eta_{i+1}(t) \overset{iid}{\sim} WN(0, \nu^2), \tag{7.5}$$

with forecast $\hat{x}_{i+1|i}(t) = x_i(t)$.

Algorithm 1. For this strategy, we use a method very similar to the second algorithm presented in Bowsher and Meeks (2008). Let $t \in T = \{4, 5, \ldots, 13, 16, \ldots, 85\}^4$, $t_1 = 4$ and $t_{2,j} \in T\backslash\{4\}$; $j = 1, \ldots, 33$. Every period i we form a portfolio of sub-portfolios with two bonds $\{t_1, t_{2,j}\}$. Define weights w_j as the proportion of the historical absolute excess return on

[4] The set T is slightly different from Bowsher and Meeks (2008) because the shortest maturity we use from the data presented in Section 7.2.1 is 3 months.

portfolio $\{t_1, t_{2,j}\}$ to the sum over all j of the same:

$$w_j = \frac{\sum_i |R_i(t_{2,j}) - R_i(t_1)|}{\sum_j \sum_i |R_i(t_{2,j}) - R_i(t_1)|},$$

where i spans the period January 1985 to December 1993.

To borrow some notation from Bowsher and Meeks (2008), let d_{ij} represent the investment rule for the amount at time i invested in each jth sub-portfolio. To determine the amount invested in each sub-portfolio, let

$$d_{ij} \quad = \quad \$1M \times w_j \times \text{sgn}[\hat{r}_{i+1|i}(t_{2j}) - \hat{r}_{i+1|i}(t_1)].$$

We set $d_{ij} = 0$ in the off chance of $\hat{r}_{i+1|i}(t_{2j}) = \hat{r}_{i+1|i}(t_1)$. Let π_{i+1} denote the time $i+1$ profit resulting from from these rules. Then

$$\pi_{i+1} \quad = \quad \sum_j d_{ij}[R_{i+1}(t_{2j}) - R_{i+1}(t_1)] \approx \sum_j d_{ij}[r_{i+1}(t_{2j}) - r_{i+1}(t_1)].$$

The results of this trading strategy are summarized in Table 7.3. Use of the FDFM model results in nearly twice the cumulative profit produced from the DNS model. Also shown is the capability of each model in successfully predicting the positive (1,520) and negative (1,252) actual spreads of the sub-portfolios in each period. Surprisingly, the random walk model has the greatest accuracy in predicting positive spreads (84%), as compared to the FDFM (73%) and DNS (61%) models. All three models are less accurate in the prediction of a negative spread, though RW is the worst by far (8%).

Table 7.3: Algorithm 1: Weighted Pairs. Use of the FDFM model results in nearly twice the cumulative profit produced from the DNS model.

	Profit ($000)				Directional Accuracy	
			Percentile		of Sub-Portfolios	
Model	Cumulative	Median	10th	90th	+	-
FDFM	**1,089**	5.06	-101.92	149.53	1,102/1,520 (72.5%)	392/1,252 (31.3%)
DNS	519	5.07	-110.77	116.02	926/1,520 (60.9%)	538/1,252 (43%)
RW	-94	-10.52	-190.5	163.64	1,274/1,520 (83.8%)	97/1,252 (7.7%)

Algorithm 2. The strategy in Algorithm 1 is a fairly basic one: to use *every* available

bond at our disposal to predict the spread between its return and a short term bond. Our second strategy[5] is more sophisticated by creating portfolios of an optimal pair of bonds each period i. Given a fixed value of t_1, we choose t_{2i} to optimize the absolute spread in predicted return:

$$t_{2i} = \arg\max_{t \neq t_1} |\hat{r}_{i+1|i}(t) - \hat{r}_{i+1|i}(t_1)|. \tag{7.6}$$

Because we examine multiple portfolios, we use a sparser set of maturities in this exercise than previous, though of the same range. This set is defined by the observed maturities of Section 7.2.1:

$$t_1, t_{2i} \in T = \{4, 7, 10, 13, 16, 19, 22, 25, 31, 37, 49, 61, 73, 85\}.$$

We perform this exercise for all choices of t_1 as long as $t_1 < t_{2i}$ and compare the results. Our investment rule d_i at time i and resulting profit π_{i+1} the next period is of a similar form to Algorithm 1:

$$\begin{aligned} d_i &= \$1M \times \ \text{sgn}[\hat{r}_{i+1|i}(t_{2i}) - \hat{r}_{i+1|i}(t_1)], \\ \pi_{i+1} &= d_i[R_{i+1}(t_{2i}) - R_{i+1}(t_1)] \approx d_i[r_{i+1}(t_{2i}) - r_{i+1}(t_1)]. \end{aligned}$$

Again, we set $d_i = 0$ whenever $\hat{r}_{i+1|i}(t_{2i}) = \hat{r}_{i+1|i}(t_1)$.

Table 7.4: Algorithm 2: Optimal Pairs Portfolio.

		Profit ($000)					Profit ($000)		
	t_1	FDFM	DNS	RW		t_1	FDFM	DNS	RW
Short	3	1,013	**3,574**	-228	Mid	21	**1,246**	202	680
	6	1,381	**2,828**	-133		24	**1,592**	242	70
	9	**1,061**	1,013	-297		30	**2,284**	203	-951
	12	**1,873**	-367	-307		36	**1,466**	919	-173
	15	**1,519**	-582	-432	Long	48	-361	**589**	236
	18	**1,081**	-481	-263		60	**740**	339	-284
						72	-131	-1	**72**

The results of the strategy are shown in Table 7.4. When the choice of t_1 is six months or less, the DNS model generates greater cumulative profit than either of the other models.

[5] The method is an adaption of the third algorithm presented in Bowsher and Meeks (2008).

However, when the choice of t_1 is within 9 and 36 months, the FDFM consistently generates significantly greater profit than the DNS and RW models. Thus, when we are free to pick the bond that optimizes the predicted spread each period, the FDFM performs rather well, provided the maturity of the first bond is within a certain range. Our final strategy expands upon this idea.

Algorithm 3. Because the choice of the optimal second bond can vary from one period to the next in Algorithm 2, it is not clear what a *consistently* good combination is. Thus, for our third strategy we consider an exploratory and exhaustive approach as a diagnostic assessment of with which combination of bonds our model excels. As such, we expand our set of bonds to include those of longer maturity:

$$t_1, t_2 \in T = \{4, 7, 10, 13, 16, 19, 22, 25, 31, 37, 49, 61, 73, 85, 97, 109\}.$$

In this modification of strategy 1 from Bowsher and Meeks (2008), the portfolio is a simple one consisting of two bonds with maturities t_1 and t_2. For the duration of the strategy, these maturities remain fixed over all periods $i = 1, \ldots, 84$. As before, the decision at time i of which bond to sell and which to buy is made based on the predicted direction of the spread in log one period returns: $d_i = \$1M \times \operatorname{sgn}[\hat{r}_{i+1|i}(t_2) - \hat{r}_{i+1|i}(t_1)]$ (we set $d_i = 0$ whenever $\hat{r}_{i+1|i}(t_2) = \hat{r}_{i+1|i}(t_1)$). This yields the time $i+1$ profit

$$\pi_{i+1} = d_i[R_{i+1}(t_2) - R_{i+1}(t_1)] \approx d_i[r_{i+1}(t_2) - r_{i+1}(t_1)].$$

We examine the cumulative profit of all combinations of this type or portfolio such that $t_2 > t_1$.

Figure 7.3 depicts the results of our final trading strategy. For each combination of $t_2 > t_1$, the name of model with the largest cumulative profit is displayed in that cell by the first initial of its acronym ("F" for FDFM, e.g.). A "+" or "-" suffix indicates the largest profit was positive or negative, respectively.

The FDFM model typically has the greatest profit when $t_2 \in \{30, \ldots, 72\}$. These results are consistent with Sections 7.2.3 and 7.2.3: the FDFM was either comparable or better on RMSFE for forecasting and for imputation on maturities in this range. We also see a certain similarity

	Short Term					Mid Term				Long Term					
Mat.	t_2=6	9	12	15	18	21	24	30	36	48	60	72	84	96	108
t_1=3	D+	D+	D+	D+	D+	R-	F-	D+	D+	F+	F-	F+	D+	D+	D+
6		F+	D+	D+	R+	D-	R-	R+	D+	F+	R+	F+	D+	D+	D+
9			R+	R+	D+	R+	R-	F+	F+	F+	F+	F+	D+	F+	D+
12				R+	D+	D+	R+	R+	F+	F+	F+	F+	R+	F+	D+
15					R+	D+	R+	F+	F+	F+	F-	F+	D+	F+	D+
18						R+	F+	F+	F+	F+	F+	F+	D+	D+	D+
21							R+	F+	F+	F+	F+	F+	D+	D+	D+
24								F+	F+	F+	F+	F+	D+	D+	D+
30									F+	F+	F+	F+	D+	D+	D+
36										D+	F+	F+	R+	D+	D+
48											R+	R+	R+	R+	D+
60												F+	F+	D+	D+
72													R+	R+	D+
84														F+	D+
96															F+

Figure 7.3: Algorithm 3: All combinations of portfolios for $t_2 > t_1$. The model with the largest cumulative profit is displayed by the first initial of its acronym with "+" or "-" indicating positive or negative profit.

in these results to those of Algorithm 2. Namely, that the FDFM typically outperformed the other two models when t_1 was exactly in this range.

For the longest maturities (> 72), the DNS model results in greater profit when $t_1 < 48$. Results for other regions are mixed. Recall from Section 7.2.1 that in our data short and mid term yields are typically spaced either 3 or 6 months apart, whereas long term maturities are spaced 12 months apart. As we saw in Section 7.2.3, as the spacing between maturities increased, the FDFM model eventually broke down; it is, after all, very much a data driven model. DNS, on the other hand, maintains the same factor loading curves regardless of the data, which could explain its greater profits at long maturities.

7.3 Conclusion

In this chapter we reviewed our method for modeling and forecasting functional time series. This novel approach synthesizes concepts from functional data analysis and dynamic factor modeling culminating in a functional dynamic factor model. By specifying error assumptions and smoothness conditions for functional coefficients, estimation by the Expectation Maximization algorithm results in non-parametric factor loading curves that are natural cubic splines.

Thus for a given time series of curves we can forecast entire curves as opposed to a discrete multivariate time series.

Chapter 8

Additional Applications

In this chapter, forecast performance for the FDFM is explored two additional applied settings. In the first application, the FDFM is fit on the call center data first introduced in Chapter 1 and used to forecast out-of-sample call data compared to the SVD method. In the second application, the FDFM is used on climatological data involving sea surface temperatures and air pressure from a region in the South Pacific. Similarly, out-of-sample forecasts are produced and are compared with several competing models including functional auto-regressive models (FAR).

8.1 Call Center Data

Revisiting the call center example introduced in the initial chapter, the FDFM is used to model and forecast the data compared with the SVD method. The data consist of daily call volumes from a call center for an Eastern U.S. bank. The volumes are recorded at fifteen minute intervals throughout each workday. Due to the high frequency of the intra day call volume measurements, it is assumed that the volumes are represented by underlying smooth curves. In this context, the daily time series of smooth curves lends itself exactly to the FDFM framework. This first section presents a description of the data followed by the models used to estimate it. Forecast results between FDFM models and an SVD model are then compared.

8.1.1 Data

The call data is collected for each weekday beginning January 6, 2003 through October 24, 2003 (210 days). Call volumes are recorded every fifteen minutes throughout the hours 7:00 am through 12:00 am the following day (68 intervals). Call volumes vary significantly depending on the day of the week, as evidenced by the plot of the mean call volumes by day in Figure 8.1. Therefore, missing values in the data set are imputed using the mean value of the non-missing values for that interval for the corresponding day. For example, if the volume for the 43rd day on the 16th interval is missing, and day 43 is a Monday, then the missing value is imputed with the mean value of all non missing 16th interval values on Mondays only.

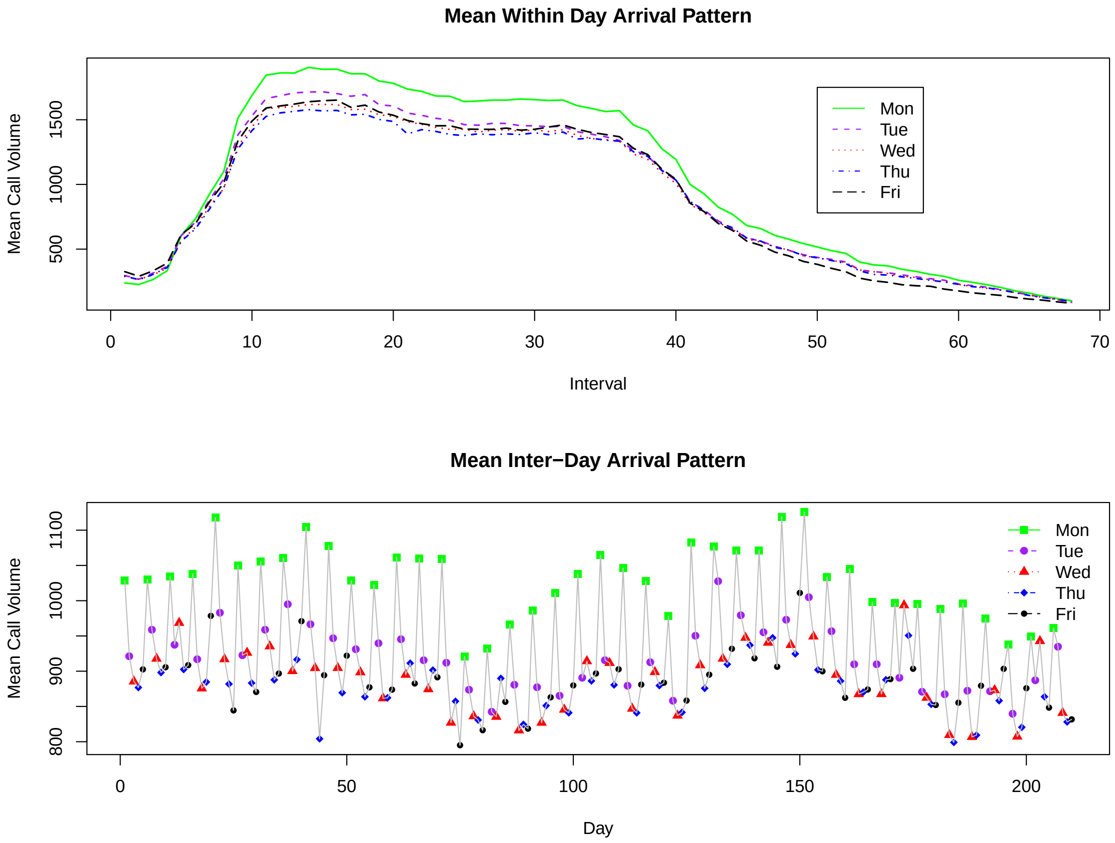


Figure 8.1: Day of Week Effect: intra-day and inter-day mean call volumes. In the top panel, mean call volume throughout the day shows a clear day-of-week effect. In the lower panel, the mean daily call volumes show a marked periodic effect.

8.1.2 Model

Before the FDFM framework an be applied to the data, it is important to note that even with the high frequency nature of the data, that it is still discrete count data. Whereas the FDFM model is based on the assumption of normally distributed data. Therefore, denoting the raw call center data for day i, interval j as N_{ij} it is assumed that N_{ij} is a Poisson process with dynamic rate $\lambda_i(t)$. See Shen (2009), Brown et al (2005) and Weinberg et al (2007). Then $x_{ij} \equiv \sqrt{N_{ij} + 1/4}$ has an approximate mean and variance of $\sqrt{\lambda_i(t_j)}$ and $1/4$, respectively. Further as $\lambda_i(t_j) \to \infty$, x_{ij} is approximately normal (Shen 2009).

This convention then permits the use of the FDFM. A five factor model is proposed, with an independent AR(1) process for each factor. Further, it is assumed that each factor has five intercepts in order to account for a day of the week effect.

$$\begin{aligned} x_{ij} &= \sum_{k=1}^{5} \beta_{ik} f_{kj} + \epsilon_{ij}, \\ \beta_{ik} &= c_{d_{i-1},k} + \varphi_k \beta_{i-1,k} + v_{ik}, \end{aligned} \tag{8.1}$$

with $\epsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma^2)$, $v_{ik} \overset{i.i.d.}{\sim} N(0, \sigma_k^2)$ and $E[\epsilon_{ij} v_{sk}] = 0$ for $i, s = 1, \ldots, n; j = 1, \ldots, m; d_i = 1, 2, 3, 4, 5$ and $k = 1, \ldots, 5$.

8.1.3 Forecast Assessment

To assess model performance, the SVD method is compared with both the sequential FDFM and the simultaneous FDFM. n is set to 160, and rolling one step forecasts are produced for days 161 through 210. For each forecast, and each model, Root Mean Squared Forecast Error (RMSE) and Average Percent Error (APE) are calculated based on call volumes $N_{ij} = x_{ij}^2 - 1/4$. Thus, forecasts are calculated as

$$\begin{aligned} \hat{x}_{i,j} &= \sum_{k=1}^{5} \hat{\beta}_{i,k} \hat{f}_{kj}, \\ \hat{\beta}_{i,k} &= \hat{c}_{d_n,k} + \hat{\varphi}_k \hat{\beta}_{i-1,k}, \\ \hat{N}_{i,j} &= \hat{x}_{i,j}^2 - 1/4. \end{aligned}$$

With APE and RMSE given by

$$RMSE_i = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(N_{i,j} - \hat{N}_{i,j})^2},$$

and

$$APE_i = \frac{1}{m}\sum_{j=1}^{m}\frac{|N_{i,j} - \hat{N}_{i,j}|}{N_{i,j}}.$$

Summary statistics for these measures are shown in Table 8.1, based on the 50 rolling forecasts. The simultaneous method has the lowest mean and median RMSE and APE. In terms of variability of forecasts, results are mixed and comparable between the three methods.

Table 8.1: RMSE and APE For Call Data From SVD Model, FDFM Sequential (EM-Q), and FDFM Simultaneous (EM-S). The simulataneous method has the lowest mean and median RMSE and APE. Variability of forecasts are comparable between the three methods.

	RMSE			APE (%)		
	EM-Q	EM-S	SVD	EM-Q	EM-S	SVD
First Quartile	37.75	38.00	38.08	4.41	4.38	4.37
Median	46.88	45.79	46.38	5.11	5.07	5.08
Mean	56.41	56.37	56.62	5.78	5.75	5.80
Third Quartile	63.12	63.06	63.34	6.47	6.45	6.51
Std. Dev.	30.59	30.75	30.47	2.30	2.29	2.29
IQR	25.37	25.06	25.26	2.06	2.07	2.14

Figure 8.2 illustrates the estimated factor loadings for each of the five factors for each of the three models. Not surprisingly the SVD method has the least smooth estimates since there is nothing implicit in that model that should guarantee any amount of smoothness. The first factor of course dominates as illustrated by the similarity and variability (or lack thereof) of the loading estimates. Examining the factor loading curves two through five however, considerably less variability is evidenced for the sequential EM model compared with the SVD method. Estimates for the sequential FDFM are generally less smooth than with the simultaneous method because smoothing parameters for the former are typically smaller than those selected for the latter. It is worth noting the greater variability of the simultaneous FDFM estimates, and to date it is unclear what the cause for this is. One possibility is that for both this data and for the

simulated data in the previous chapter, the EM for the simultaneous method fails to converge before 100 iterations are reached. In general, the EM may be used with a Newton–Raphson algorithm for faster convergence. Other causes may include a theoretical issue or even one due to numerical precision; regardless further investigation is warranted and will be performed.

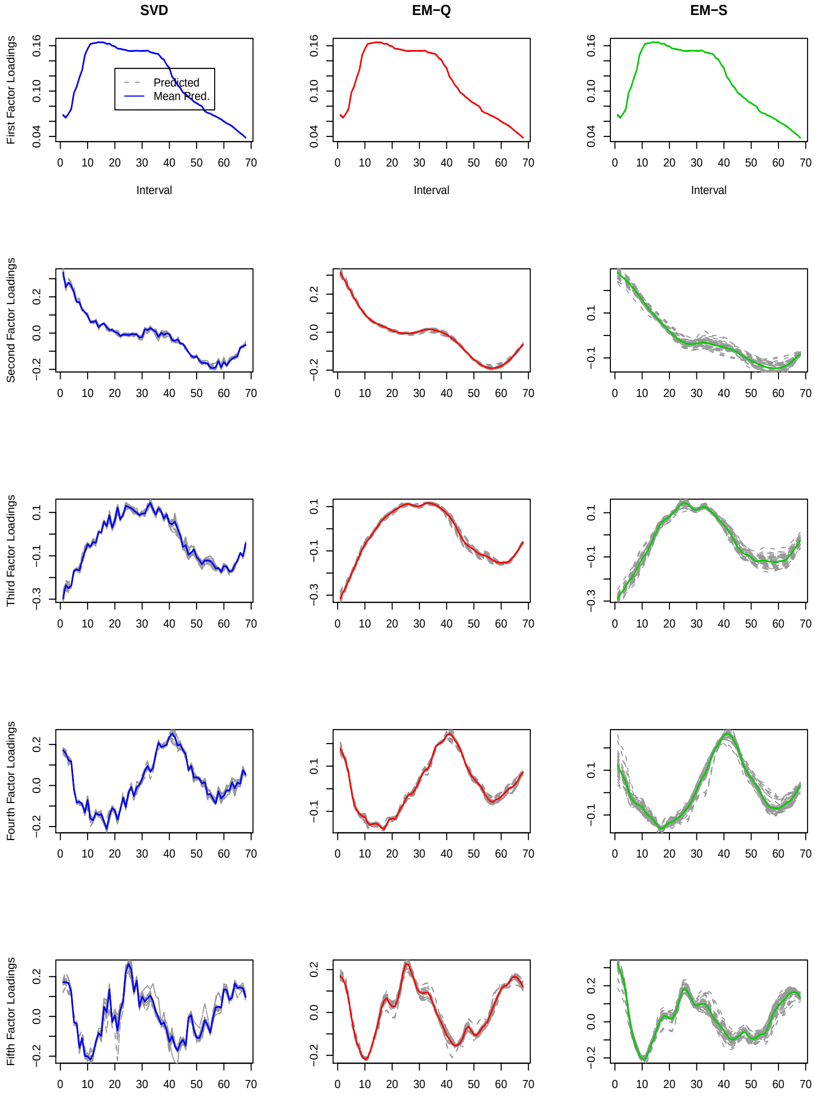


Figure 8.2: Estimated Factor Loadings. EM methods result in smoother estimates due to the presence of a smoothing parameter. The first factor dominates. The greatest variability in estimates for the loading curves is exhibited in the simultaneous method, for which the reason may be slow convergence.

Finally, Figure 8.3 depicts the mean actual and mean predicted call volumes for each method, broken down by day of the week. All methods appear to correctly separate the day of the week effect. However it is also the case that while the estimates for Mondays are correctly the largest estimated volumes, all methods on average under predict the effect. A possible solution may be to include a multiplicative parameter either in addition to or as a replacement for the additive intercept.

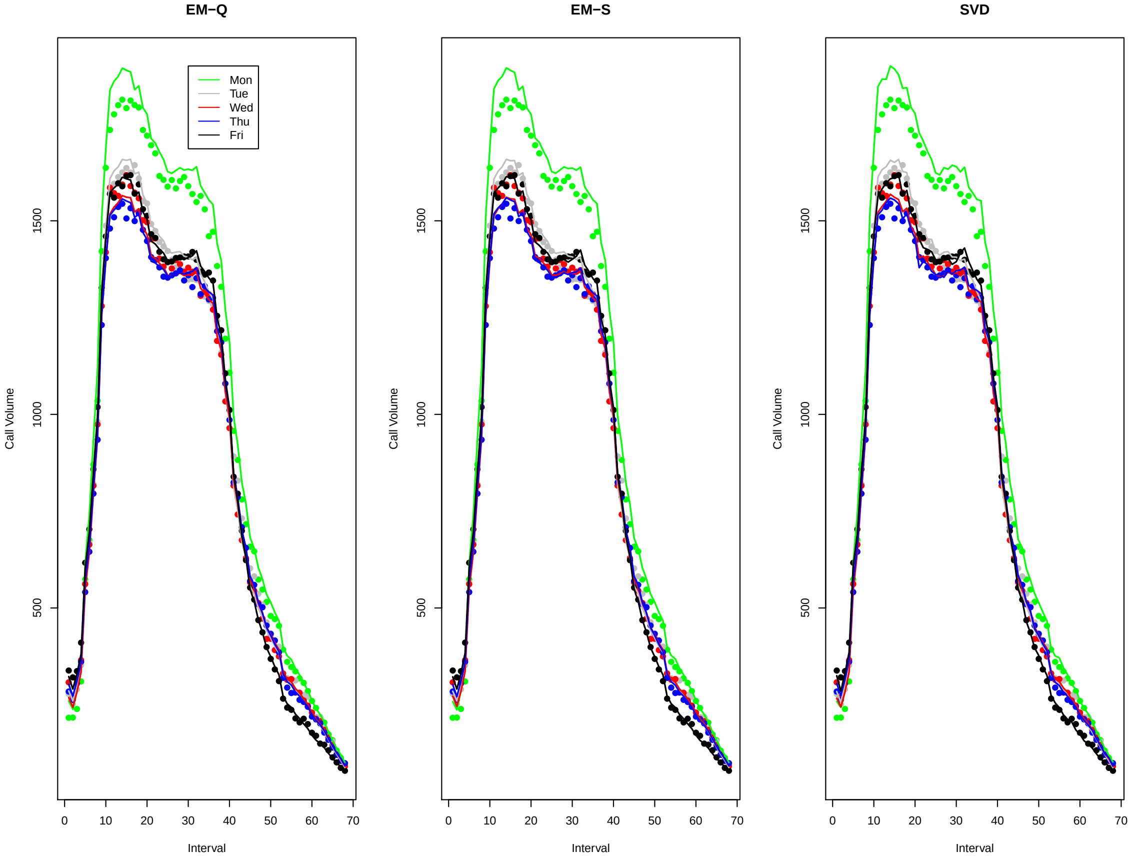


Figure 8.3: Forecasts by Day of Week. All methods are able to capture the day of the week effect. However all models also underpredict call volumes on Mondays, on average.

8.2 Climatological Data

Similar to the the call center application, the FDFM can be used to model univariate periodic time series by specifying the periodic component as a smooth curve. Then the univariate time series can also be viewed as a time series of seasonal curves. This particular example compares the FDFM with several other methods used for time series of curves forecasting on strongly seasonal data representing the well known El Nino climatological phenomenon.

8.2.1 Data

The following analysis draws heavily from the analysis performed by Besse et al. (2000). Analysis on two data sets is performed. The first is monthly mean sea surface temperature (SST) for the El Nino 3 domain defined as 5S to 5N, 150W to 90W for the period January 1950 through December 1996; hereafter referred to as the EN data. The second is the corresponding sea level air pressure at Tahiti during the same time period which is used as a proxy for the Southern Oscillation behavior observed in the El Nino phenomenon. In a similar manner, this data shall be referred to as the SO data.

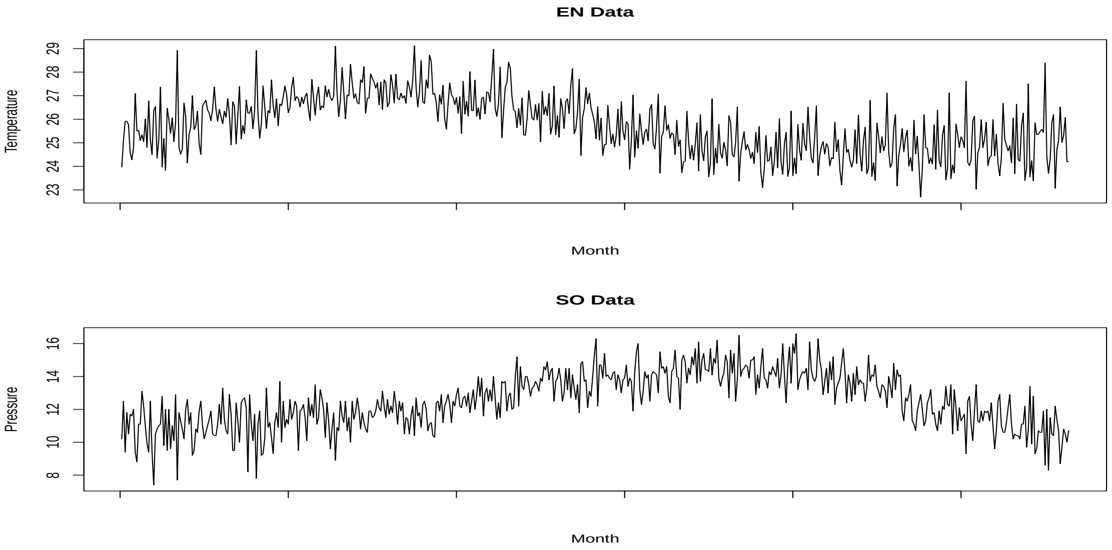


Figure 8.4: Temperature and Pressure Data for El Nino Region

8.2.2 Models

For the FDFM, a model with five independent AR(1) factors is fit on the 37 year period 1950 through 1986. One year ahead rolling forecasts are produced for the the period 1987 through 1996. Comparisons are made to the mean squared error (MSE) and mean relative absolute error (MRAE) performance measures for other functional time series models presented in Besse et al. (2000). It is important to note that for the other models presented, the models are fit based on the 37 year period, and that the 10 one year ahead forecasts are based on those fixed parameters. That is, the fitting is not updated each year, only the updated data is fed through the model.

In order to apply the FDFM, first the univariate time series data must be restated as a time series of curves. In either of the cases for the SO or EN data, consider the monthly data as a univariate time series y_s, $s = 1, \dots, T$. In the data there is periodicity m of 12 months over n years; $i = 1, \dots, n$ and $j = 1, \dots, m$ so that $T = n \times m$. Instead of a univariate time series, the seasonal component can be modeled as a smooth curve, giving rise to an annual time series of curves. Using the previous notation from the functional dynamic factor model then, $x_i(t_j) = x_{ij}$ is the observation for the jth month of the ith year.

$$\begin{aligned} y_s &= x_{ij} && (8.2) \\ j &= (s-1) \mod m + 1 \\ s &= (i-1)m + j. \end{aligned}$$

The FDFM fit in this setting is

$$\begin{aligned} x_{ij} &= \sum_{k=1}^{5} \beta_{ik} f_{kj} + \varepsilon_{ij} && (8.3) \\ \beta_{ik} &= c_k + \varphi_k \beta_{i-1,k} + v_{ik}. \end{aligned}$$

The one step ahead forecasts are then calculated by

$$\begin{aligned} \hat{x}_{i,j} &= \sum_{k=1}^{5} \hat{\beta}_{i,k} \hat{f}_{kj} \\ \hat{\beta}_{i,k} &= \hat{c}_k + \hat{\varphi}_k \hat{\beta}_{i-1,k}. \end{aligned} \tag{8.4}$$

Besse et al. (2000) use seven other models to assess forecast performance for both data sets. Briefly these are:

Traditional ARMA Models: These are methods based on traditional time series methods.

1. Climatology: The climatology model is the most straightforward in that it is simply the previous year's mean monthly temperature or atmospheric pressure.
2. Seasonal ARIMA: These models treat the data as univariate seasonal data. For the EN data a ARIMA$(0,1,1)(1,0,1)_{12}$ model is fit; for the SO data a ARIMA$(1,1,1)(0,1,1)_{12}$ is fit.

Non-Parametric Models: A popular approach with functional data.

1. Kernel.
2. Functional Kernel.

Functional Auto-Regressive models (FAR): Besse et al. (2000) use various specifications of the FAR model.

1. Smooth FAR(1) models.
2. Local FAR(1).

8.2.3 Forecast Assessment

The models can be compared on the basis of their forecasting performance of the ten year period 1987 through 1996. One step ahead forecasts are equivalent to one year ahead forecasts for each of the twelve months with the exception of the ARIMA models which are essentially forecasting 1 through 12 steps ahead. Mean squared forecast error (MSE) and mean relative absolute forecast error (MRAE) are calculated for each model. These results are presented in Table 8.2. One step MRAE at each year i calculated as

$$\mathrm{MRAE}_i = \frac{1}{m}\sum_{j=1}^{m}\frac{|x_{i,j} - \hat{x}_{i,j}|}{|x_{i,j}|}. \tag{8.5}$$

Where $\hat{x}_{i,j}$ represents the forecasted temperature or pressure given the information available at year $i-1$. Then the average over the 10 years is taken. Similarly for MSE is calculated as

$$\mathrm{MSE}_i = \frac{1}{m}\sum_{j=1}^{m}\left[x_{i,j} - \hat{x}_{i,j}\right]^2. \tag{8.6}$$

and again the average is taken over the 10 years. For the EN data, the FDFM outperforms only the climatology and ARIMA models. However in terms of methodology, the FDFM is more easily implemented than the other functional approaches. The SO data is in general noisier and so MSE and MRAE measures for the data are higher than with the EN data. Also the distinction between each model's performance is less pronounced. Here the FDFM only outperforms the ARIMA model in terms of MSE but is still close to the MSE for the other models. On MRAE, the FDFM performs as well as the climatology model, but the "worst" model to the "best" in terms of MRAE is only separated by 0.5%.

Table 8.2: Summary of one year ahead rolling forecasts of the EN and SO data. The FDFM outperforms the climatology and ARIMA models for the EN data. For the SO data, the FDFM only outperforms the ARIMA model but is much closer in performance to the other functional methods.

	El Nino Index		S. Osc. Index	
Model	MSE	MRAE	MSE	MRAE
FDFM	0.72	2.5 %	0.93	6.3 %
Climatology	0.73	2.5%	0.91	6.3 %
ARIMA	1.45	3.7%	0.95	6.2 %
Kernel	0.60	2.3%	0.87	6.1%
Functional Kernel	0.58	2.2%	0.82	6.0 %
Smooth FAR(1)	0.55	2.3 %	0.78	5.8 %
Smooth FAR(1) with $q = p = 12$	0.60	2.4 %	0.91	6.5 %
Local FAR(1)	0.53	2.2 %	0.82	5.8 %

Figure 8.5 displays a sample of the forecasts produced by all of the models for the EN and SO data respectively. In the top panel for the EN data, the FDFM most closely resembles the

seasonal ARIMA model but is slightly closer to the actual data. The other functional methods are closer to actual. The lower panel shows the forecasting results on the SO data for the year 1987. In this case, all models have the propensity to overestimate the period April through October. The FDFM performs at least as well as the other functional models besides February, and the ARIMA model produces the lowest, and in this case closest forecast.

Finally, Figures 8.6 and 8.7 show the FDFM performance for all ten years of the forecast period. Panel (a) of Figure 8.6 shows the mean of the actual EN data and mean forecast by month which appears to fit the data quite well. However, on panel (b), there is little variation in the forecasts in comparison with the actual data which could explain why MSE and MRAE is higher with the FDFM than with some of the other models. Panel (a) of Figure 8.7 shows mean forecast and actual for the SO data. Again, the fit is very good, perhaps better than with the EN data. Upon examination of panel (b), the FDFM forecasts are again less variable than the corresponding actual seasonal curves. However they are still closer to the actual data than in the EN case which is in accordance with the results in Figure 8.5 and Table 8.2.

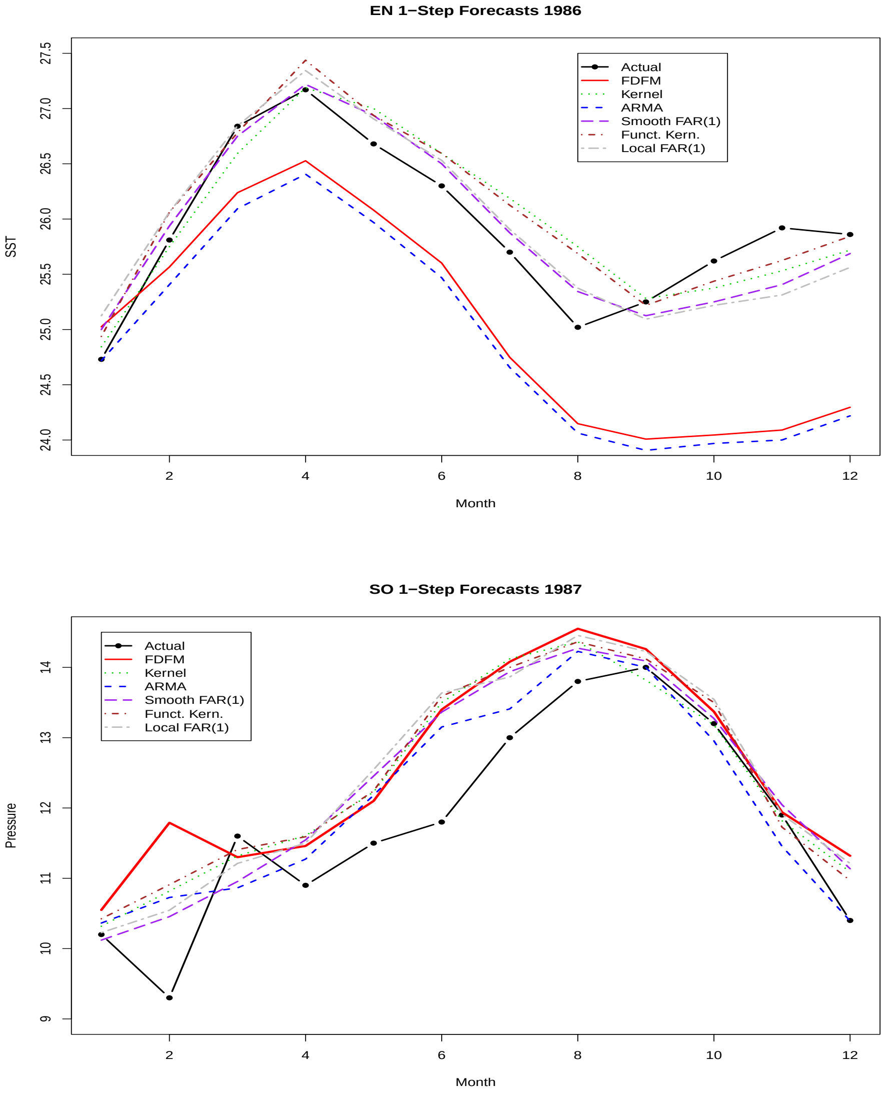


Figure 8.5: Forecasts for the years 1986 (EN) and 1987 (SO) . In the top panel, the FDFM on the EN data closely resembles the ARIMA fit though is slightly closer to the actual data. The other functional models closely follow the pattern of the true data. In the lower panel are SO forecasts for the year 1987. All models produce similar forecasts and overestimate atmospheric pressure for the period April through October. Besides February, the FDFM performs at least as well as the other functional methods.

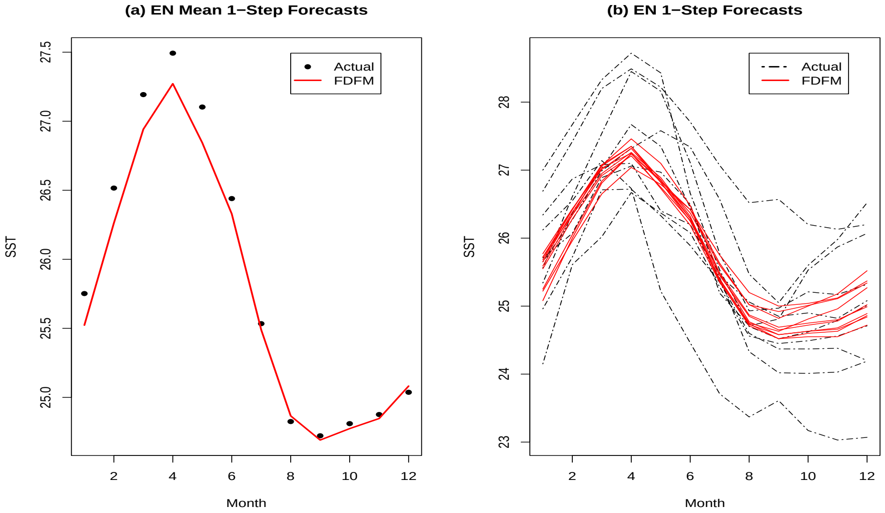


Figure 8.6: FDFM Forecast Performance on EN data. Panel (a) shows mean actual and mean forecast for the EN data 1987 through 1996. The model appears to fit quite well, however panel (b) shows the actual data is much more variable over the period than the FDFM forecasts.

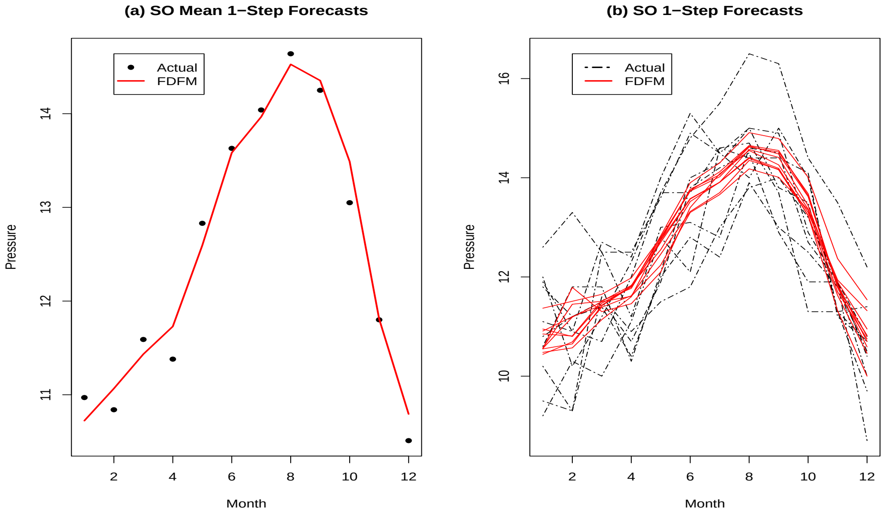


Figure 8.7: FDFM Forecast Performance on SO data. Panel (a) shows mean actual and mean forecast for the SO data 1987 through 1996. Again the model appears to fit quite well. Panel (b) shows the actual data is again more variable than what the FDFM forecasts but less dramatically than with the EN data.

Chapter 9

Conclusion and Future Work

This dissertation has introduced a novel method by which to forecast time series of curves which is applicable in numerous settings. The functional dynamic factor model (FDFM) integrates ideas stemming from functional data analysis and dynamic factor models. The factor approach facilitates dimension reduction in the sense paring down a large set of observed time series into a smaller set of unobserved factor time series. By specifying that the factor loadings for each factor are part of a smooth factor loading curve, this facilitates a functional perspective.

With distributional assumptions, maximum likelihood is a natural choice for estimation. With the aid of roughness penalties as in Green and Silverman (1994), conditions are placed on the factor loadings so that they may be interpreted as smooth factor loading curves. Considering the unobserved factors as a problem of missing data, the Expectation Maximization algorithm in conjunction with a penalized likelihood expression is used to estimate the model parameters and factor loading curves; pending some initial estimates from singular value decomposition of the observed data matrix.

EM estimation is, of course, an iterative process. Each iteration is rife with computational intensity, from large matrix inversions in the E-step to costly GCV selection of smoothing parameters in the M-step. However, several efficiencies to this end have been illustrated and derived, including even some practical commentary as to the computing packages that may be utilized.

It has been shown that the estimated factor loading curves for the FDFM form natural cubic splines under a specific roughness penalty. This result facilitates straightforward interpolation and permits the forecast or imputation of entire curves rather than a sequence of discrete points. Another fortunate result that has been shown is that cross validation and generalized cross validation for the optimal choice of the smoothing parameter does not require re-estimation of the FDFM for each leave-out. We have shown convenient expressions exist for the CV and GCV criteria based only on FDFM estimation of the full data set.

Finally, actual FDFM performance has been displayed on both simulated and real data; and in the latter case, among some diverse applications. A compelling application is yield curve forecasting, where existing approaches typically exhibit a tradeoff of consistency-with-economic-theory and goodness-of-fit. However, through multiple forecasting exercises we have shown that the model satisfies both of these criteria. Further applications showcase the model's viability to settings well outside of economics and yield curve forecasting and where a prior theory does not exist. We have shown applicability to call volume forecasting as well as the capability to model and forecast seasonal time series in a climatological setting.

For the simulated data based on call volumes, the FDFM results in accurate parameter and factor loading curve estimation, with low MSE, bias, and variance in comparison to a benchmark models. For the simulated data inspired from true yield curve data, we have illustrated viable methods for model selection and assessment. This includes identification of the number of factors for setting where it is unclear what that number should be. Further, AIC and BIC can be used to select to order of the time series process that the dynamic factors follow. Finally, we have derived a bootstrap approach to construct confidence intervals for model estimates, and forecast intervals for the resulting forecasts.

Indeed, this exciting new class of models is fertile for further development and application. Below is a listing of some possible directions of future research. Among these, a few are further elaborated upon as probable later model extensions.

Other Correlated Processes In the present model, the factors were assumed to follow independent, stationary, auto-regressive processes. A possible enhancement to the FDFM is to consider other types of correlated processes. Those of most interest are:

1. Moving Average and Integrated Processes: Clearly there is more to the acronym

"ARIMA" than just "AR." While still maintaining the assumption of independent factors, integrated or non-stationary univariate processes may be considered for the factors. Another obvious addition is to include moving average components to the independent factors. Further investigation is warranted as to why or why not these might be useful in application.

2. Periodic Auto-regressive Models (PAR): Independent AR(1) factors may be too strong an assumption to properly account for the time series component of the observed data. A more general alternative may be to consider PAR(1) factors. As was seen with the call center data, there is a clear day of week effect, and mostly likely there are other sources of periodicity in the data (Taylor, 2008). A PAR specification then may better account for periodic correlation than an AR(1) specification, without the need to escalate to a more complex model like a VAR.

3. Vector Auto-Regression and Cointegrated Processes: Removing the assumption of independent factors gives rise to Vector Auto-Regressions or even more general multivariate time series procedures. The model presented by Bowsher and Meeks (2008) has already illustrated an application where a cointegrated VAR specification is desirable. Further, Pena and Poncela (2006) expanded the dynamic factor model to include non-stationary factors, and used the model to forecast interest rates. A popular method for non-stationary or (co)integrated time series using maximum likelihood estimation is a state space approach using an algorithm like the Kalman Filter.

   The state space formulation of the FDFM is:

$$\begin{aligned} X_i &= \mathbf{F}'\boldsymbol{\beta}_i + \epsilon_i, \qquad (9.1) \\ \boldsymbol{\beta}_i &= \Phi\boldsymbol{\beta}_{i-1} + v_i, \\ \epsilon_i &\sim N(0, \sigma^2\mathbf{I}_m), \\ v_i &\sim N(0, V). \end{aligned}$$

   A future area of research then to explore to possibility of vector tome series in the context of the FDFM. The question is as to whether the functional aspect of the model can be incorporated into a method like the Kalman Filter.

4. Spatial and Spatio-temporal Processes: Discussion of correlated processes would not be complete without discussion of the spatial domain. An interesting application of the FDFM might be to have the factors represent one axis of coordinates (latitude), while the loading curves represent the other (longitude). Perhaps even more compelling would be to consider the evolution of smooth *surfaces* over time.

Further Opportunities. This dissertation included several mathematical results, primarily in regard to the computation involved in estimation. Although the aspects of practical implementation are fairly complete, some of the more inference related properties of the FDFM have yet to be derived, such as:

1. Asymptotic Properties: The asymptotic properties of any of the estimators in the model have yet to be shown. Meng and Rubin (1993) were able to derive some results in the case of a general EM model; most notably that the algorithm always converges. In the current setting, the likelihood expression is augmented by a series of roughness penalty terms, and so their results may not necessarily hold. Clearly then it is desirable to see if these beneficial results hold in the case of this FDFM.

2. Slow EM Convergence: In the Real Applications chapter (Chapter 8) it was noted that the EM was slow to converge and that this may account for some of the less-than-stellar forecast performance of the FDFM as compared with other models. The theoretical question of convergence has already been asked above; here the question is, can the EM be made to converge faster? Methods exist that can be used in conjunction with the EM so that it converges in fewer iterations; methods such as Newton-Raphson, conditional methods as in Zhao et al. (2008) or other augmented approaches like the so-called Supplemented EM Algorithm of Meng and Rubin (1991). Faster convergence obviates an arbitrary limit on the number of iterations until the EM is deemed converged. Therefore, parameter estimates and the forecasts generated from them can truly be considered final; this permits more

valid comparisons with other models.

Thus, the FDFM has been shown to be a viable model for working with functional data and is rife with opportunity for further development and research.

Bibliography

Alonso, A. M., Garcia-Martos, C., Rodriguez, J., and Sanchez, M. J. (2011), "Seasonal Dynamic Factor Analysis and Bootstrap Inference: Application to Electricity Market Forecasting," *Technometrics*, 53, 137–151.

Basilevsky, A. (1994), *Statistical Factor Analysis and Related Methods*, Wiley: New York.

Bathia, N., Yao, Q., and Ziegelmann, F. (2010), "Identifying the Finite Dimensionality of Curve Time Series," *The Annals of Statistics*, 38, 3352–3386.

Besse, P. C., Cardot, H., and Stephenson, D. B. (2000), "Autoregressive Forecasting of Some Functional Climatic Variations," *Scandinavian Journal of Statistics*, 27, 673–687.

Bowsher, C. G. and Meeks, R. (2008), "The Dynamics of Economic Functions: Modeling and Forecasting the Yield Curve," *Journal of the American Statistical Association*, 103, 1419–37.

Brown, L. D., Cai, T., Zhang, R., and Zhao, L. (2010), "The Root-Unroot Algorithm for Density Estimation as Implemented Via Wavelet Block Thresholding," *Probability Theory and Related Fields*, 146, 401–433.

Cox, J. C., Ingersoll, J. E., and Ross, S. A. (1985), "A Theory of the Term Structure of Interest Rates," *Econometrica*, 53, 385–407.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via EM Algorithm," *Journal of the Royal Statistical Society, Series B (Methodological)*, 39, 1–38.

Diebold, F. X. and Li, C. (2006), "Forecasting the Term Structure of Government Bond Yields," *Journal of Econometrics*, 130, 337–64.

Diebold, F. X., Rudebusch, S., and Aruoba, S. (2006), "The Macroeconomy and the Yield Curve," *Journal of Econometrics*, 131, 309–338.

Duffee, G. (2002), "Term Premia and Interest Rate Forecasts in Affine Models," *Journal of Finance*, 57, 405–443.

Duffie, D. and Kan, R. (1996), "A Yield Factor Model of Interest Rates," *Mathematical Finance*, 6, 379–406.

Engle, R. and Watson, M. (1981), "A One-Factor Multivariate Time Series Model of Metropolitan Wage Rates," *Journal of the American Statistical Association*, 78, 774–781.

Fama, E. and Bliss, R. (1987), "The Information in Long-Maturity Forward Rates," *American Economic Review*, 77, 680–92.

Gans, N., Koole, G. M., and Mandelbaum, A. (2003), "Telephone Call Centers: Tutorial, Review and Research Prospects," *Manufacturing Service Operations Management*, 5, 79–141.

Geweke, J. F. and Singleton, K. J. (1981), “Maximum Likelihood Confirmatory Factor Analysis of Economic Time Series,” *International Economic Review*, 22, 37–54.

Green, P. J. and Silverman, B. W. (1994), *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Chapman and Hall: New York.

Hamilton, J. D. (1994), *Time Series Analysis*, Princeton University Press: Princeton, NJ.

Heath, D., Jarrow, R., and Morton, A. (1992), “Bond Pricing and the Term Structure of Interest Rates: a New Methodology for Contingent Claims Valuation,” *Econometrica*, 60, 77–105.

Huang, J. Z., Shen, H., and Baja, A. (2008), “Functional Principal Components Analysis via Penalized Rank One Approximation,” *Electronic Journal of Statistics*, 2, 678–695.

Hull, J. and White, A. (1990), “Pricing Interest–Rate–Derivative Securities,” *Review of Financial Studies*, 3, 573–592.

Hurd, H. L. and Miamee, A. (2007), *Periodically Correlated Random Sequences: Spectral Theory and Practice*, Wiley: Hoboken, NJ.

Hyndman, R. J. and Shang, H. L. (2009), “Forecasting Functional Time Series,” *Journal of the Korean Statistical Society*, 38, 199–211.

Judge, G. G. (1985), *The Theory and Practice of Econometrics*, Wiley: New York.

Koopman, S. J., Mallee, M. I. P., and der Wel, M. V. (2010), “Analyzing the Term Structure of Interest Rates Using the Dynamic Nelson-Siegel Model with Time-Varying Parameters,” *Journal of Business and Economic Statistics*, 28, 329–343.

Lay, D. C. (2003), *Linear Algebra and its Applications*, Pearson Education, Inc., 2nd ed.

Magnus, J. R. and Neudecker, H. (1999), *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Wiley: New York.

Meng, X.-L. and Rubin, D. B. (1991), “Using EM to Obtain Asymptotic Variance-Covariance Matrices: The SEM Algorithm,” *Journal of the American Statistical Association*, 186, 899–909.

Meng, X. L. and Rubin, D. B. (1993), “Maximum Likelihood Estimation via the ECM Algorithm: A General Framework,” *Biometrika*, 80, 267–278.

Molenaar, P. C. M. (1985), “A Dynamic Factor Model for the Analysis of Multivariate Time Series,” *Psychometrika*, 50, 181–202.

Nelson, C. R. and Siegel, A. F. (1987), “Parsimonious Modeling of Yield Curves,” *Journal of Business*, 60, 473–489.

Pena, D. and Box, G. E. P. (1987), “Identifying a Simplifying Structure in Time Series,” *Journal of the American Statistical Association*, 82, 836–843.

Pena, D. and Poncela, P. (2004), “Forecasting with Nonstationary Dynamic Factor Models,” *Journal of Econometrics*, 119, 291–321.

— (2006), “Nonstationary Dynamic Factor Analysis,” *Journal of Statistical Planning and Inference*, 136, 1237–1257.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992), *Numerical Recipes in Fortran: The Art of Scientific Computing*, Cambridge University Press: New York.

Ramsay, J. O. and Silverman, B. W. (2002), *Applied Functional Data Analysis: Methods and Case Studies*, Springer-Verlag: New York.

— (2005), *Functional Data Analysis*, Springer-Verlag: New York, 2nd ed.

Shen, H. (2009), “On Modeling and Forecasting Time Series of Curves,” *Technometrics*, 51, 227–38.

Shen, H. and Huang, J. Z. (2005), “Analysis of Call Centre Arrival Using Singular Value Decomposition,” *Applied Stochastic Models in Business and Industry*, 21, 251–263.

— (2008), “Interday Forecasting and Intraday Updating of Call Center Arrivals,” *Manufacturing and Service Operations Management*, 10, 391–410.

Taylor, J. W. (2008), “A Comparison of Univariate Time Series Models for Forecasting Intraday Arrivals at a Call Center,” *Management Science*, 54, 253–265.

Vasicek, O. (1977), “An Equilibrium Characterization of the Term Structure,” *Journal Financial Economics*, 5, 177–188.

Zhao, J. H., Yu, P. L. H., and Jiang, Q. (2008), “ML Estimation for Factor Analysis: EM or Non-EM?” *Statistical Computing*, 18, 109–123.