ALGORITHMS AND STRUCTURES FOR COVARIANCE ESTIMATES WITH APPLICATION TO FINANCE

A Dissertation Presented

by

Tengjie Jia

to

The Graduate School

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in

Applied Mathematics & Statistics

Quantitative Finance

Stony Brook University

December 2013

Stony Brook University

The Graduate School

Tengjie Jia

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Andrew P. Mullhaupt – Dissertation Advisor
Research Professor
Dept. of Applied Mathematics and Statistics, SUNY Stony Brook

Svetlozar Rachev – Chair person of Defense
Frey Family Foundation Chair of Quantitative Finance
Dept. of Applied Mathematics and Statistics, SUNY Stony Brook

Evangelos Coutsias – Committee Member
Professor
Dept. of Applied Mathematics and Statistics, SUNY Stony Brook

Young Shin Aaron Kim – External Committee Member
Assistant Professor
College of Business, SUNY Stony Brook

This dissertation is accepted by the Graduate School

Charles Taber
Interim Dean of the Graduate School

Abstract of the Dissertation

ALGORITHMS AND STRUCTURES FOR COVARIANCE ESTIMATES WITH APPLICATION TO FINANCE

by

Tengjie Jia

Doctor of Philosophy

in

Applied Mathematics and Statistics

Quantitative Finance

Stony Brook University

2013

Factor analysis is an important statistical tool used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables which are called factors. Maximum likelihood estimation (MLE) has been popular for fitting factor analysis. Among variety of iterative methods that can be used to perform MLE, the EM algorithm is probably one of the most stable in terms of monotonely increasing the likelihood and the easiest to implement. However, in the real world, the rate of convergence of EM could be painfully slow in factor model estimation.

In this dissertation, we study two popular problems in algorithms and structures for covariance estimates. The first problem is factor analysis and mixture of factor analyzers models estimation by using the α-EM algorithm. In the α-EM algorithm we replace the logarithm by α-logarithm. Logarithms have important roles besides the derivation of the log-EM algorithm. The Kullback-Leibler divergence and Fisher

information matrix all bring about the logarithm. For α-logarithm with different values of α we actually have other important information measurements such as the Hellinger distance and weighted square distance besides the Kullback-Leibler divergence. After calculation we get two non-tractable update equations in α-EM. In order to get tractable update equations as we have in log-EM, we need to do two more things. One of them is iteration index shifting and the other one is series expansion. These two steps are necessary for practical reasons. In addition, we apply the α-EM algorithm to actual financial data. The speed of convergence is much faster than traditional log-EM algorithm and you could choose different values of α to achieve the best rate of convergence.

The second problem is covariance estimation by using matrix fraction representations. There is a vast literature that suggests factor models for dealing with covariance estimation. One of the important reason is that we can interpret the statistical factors by actual financial indicators. Here, we consider using matrix fraction representations. One of the many reasons that this would be a better idea than factor model is that the inverse of a factor model no longer have the same factor structure. But fraction representations don't have this problem. Another reason is that factor model is not a convex set. But band fraction representation is a convex set. More importantly we can show that factor model is a special case of band fraction representation. That means if the covariance matrices have factor structure we still use band fraction representation. It had been expected that band fraction representation would be better than factor model. We show the foresight is true.

The main contribution of the thesis:

- We apply the α-EM algorithm to do factor model and mixture of factor model estimation. In the α-EM algorithm we replace the logarithm by α-logarithm. We get non-causal (implicit) update equations through a lot of calculation.

- We use shift of index and series expansion to get causal (explicit) update equations of α-EM algorithm for both factor model and mixture of factor models. We show that log-EM is a special case of the α-EM algorithm. We compare the convergence speed of the log-EM algorithm and the α-EM algorithm in terms of both log-likelihood and the Hellinger distance.

- We compute gradients of the α-log-likelihood function of factor model. We apply conjugate gradient acceleration to the α-EM algorithm. Empirical results are given.

- We consider a new structure which is band fraction representation for covariance estimates. We show that band fraction representation includes factor model as a special case. We show that for factor model with d factors and band fraction with $d + 1$ bandwidth, the Hellinger distance between sample covariance and band fraction representation is much smaller than it between sample covariance and factor model. This is also true in terms of log-likelihood.

Contents

List of Tables

List of Figures

Acknowledgements

In the period of my doctoral study, I have received considerable help and support from numerous people.

I need to first thank Prof. Ann Tucker who was a visiting associate professor at AMS department and the executive director of Stony Brook's Quantitative Finance program in the early stage. As the main contributor in working towards the official startup of the SBQF program, she brought us fresh knowledge about quantitative finance from her industry experience. She also provided us a fabulous platform as in the QF program to learn about the mysterious "Wall Street". I was very lucky to be one of the few students at the start of QF program. Prof. Ann Tucker was very much kind and friendly to every student and we missed her so much when she left the department.

Prof. Andrew P. Mullhaupt, my academic advisor, who had over twenty years experience in high-frequency trading. Undoubtedly provided me tremendous guidance on how to conduct research, how to apply research to the financial world. He brought us not only inspiring lectures and insightful ideas but also an insider vision to the state-of-the-art science and technology in quantitative finance. Back to the summer of 2010, after my first research on Trade and Quote database and prof. Ann Tucker's interim guidance, I became a PhD student of Prof. Mullhaupt. He was always generous and kind in sharing his opinions and providing his guidance not only limited to quantitative research. I am thankful to all his help in answering my questions and I feel very much fortunate that I had the opportunity to learn from such a great professor.

Prof. Svetlozar Rachev, our program director, offered me huge guidance on how to conduct research and how to present research to the academic world. He was always available and patient for any discussion not only limited to academic research and he was a beloved professor by all students at our QF program. He also brought

his wide and strong research nexus and provided us numerous opportunities to study the latest and most challenging topics in quantitative finance. Besides research, Prof. Rachev provided us enormous opportunities to work on patents which helped us put the knowledge of the book into practice. I have also learned a lot from him about academic writing skills and experience, which will benefit me for a life-long time.

I also want to thank Prof. Robert Frey for his enormous effort in starting such a wonderful program which attracts so many prominent professors, and his friendliness and patience in communicating with us on line. Other professors, such as Prof. Haipeng Xing who provided useful comments in my preliminary exam. I'd like to express my appreciation to many colleagues including (but not limited to) Xu Dong, Barret Shao, Xiao Yu, Riyu Yu, with whom I had chance to work together and discuss about problems and ideas during my research. I'd also like to thank Michael Tiano who offered me many useful comments about my thesis and helped me a lot in editing my thesis.

Since the last year of my PhD study, I have been working as a quantitative researcher at Meadowood Capital Management. In addition to the academic research on campus, this industry experience has provided me a much deeper understanding about practical quantitative trading. I am grateful to this great opportunity and I want to express my special thanks to our CEO Paul Cargiulo and COO David Spring. In the meanwhile, I am thankful to the beneficial discussions with my colleagues Lorne Applebaum and Ilya Presman.

Last but most, I would like to thank the support from my family members during all my study life, without whom I would not success in any accomplishments.

Vita, Publications and/or Fields of Study

I was born in Shijiazhuang, Hebei, P.R. China in 1986. I received my B.S in Applied Mathematics at Donghua University in 2009. My research interests include mathematical models in finance.

I was admitted to the master program in Applied Mathematics and Statistics at Stony Brook University in 2009. After one half year of study I jointed to the PhD program. My concentration is quantitative finance and my academic advisor is Professor Andrew P. Mullhaupt. My research topics in PhD include Trade and Quote data analysis, matrix fraction representation, α-EM algorithm, factor model estimation, covariance shrinkage and quadratic programming. The related research work include three working papers:

1. Tengjie Jia, Andrew P. Mullhaupt. Band fraction representation estimation of covariance estimation. Working paper. (2013)

2. Tengjie Jia, Andrew P. Mullhaupt, Lorne Applebaum, Xu Dong. Factor model estimation by using the α-EM algorithm. Working paper. (2013)

3. Tengjie Jia, Andrew P. Mullhaupt. Intraday financial data dependency. Working paper. (2013)

I am currently working as an quantitative researcher at Meadowood Captial Management, New York, NY

1 Introduction

In this section, we do a literature review about previous work in factor model estimation, mixture of factor models estimation, α-EM algorithm, conjugate gradient and rank-structured matrices such as semiseparable matrix. At the end of this section, we give the organization of the following sections of the thesis.

Factor analysis is a statistical method used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors. Factor analysis searches for such joint variations in response to unobserved latent variables. The observed variables are modeled as linear combinations of the potential factors, plus “error” terms. Factor models have been widely used to construct portfolios with certain characteristics, such as risk, because they have many useful properties that sample covariance matrices don’t have. One advantage of a factor model is the reduction of number of variables, by combining two or more variables into a single factor. Another advantage is the identification of groups of inter-related variables, to see how they are related to each other.

In statistics, a mixture model is a probabilistic model for representing the presence of subpopulations within an overall population. Mixture models don’t require that the observed data-set should identify the sub-population to which an individual observation belongs. Formally a mixture model corresponds to the mixture distribution which represents the probability distribution of observations in the overall population. Mixture of factor analysis (MFA) models are widely used in clustering and dimensionality reduction. These are two of the fundamental problems in unsupervised learning. The reason is that MFA models can perform clustering and dimensionality reduction simultaneously. MFA looks for directions that have maximal interesting correlations within each cluster. In model based clustering the data are assumed to come from a finite mixture model. For quantitative data each mixture component is usually modeled as a multivariate Gaussian distribution.

The expecatation-maximization (EM) algorithm is a well known iterative method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. Typically these models involve latent variables in addition to unknown parameters and known data observations. That is, either there are missing values among the data, or the model can be formulated more simply by assuming the existence of additional unobserved data points. The EM algorithm has been suggested for fitting factor and mixture of factor models. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

A. P. Dempster, N. M. Laird and D. B. Rubin presented both the general theory of EM algorithms and a general approach to iterative computation of maximum-likelihood estimates in 1977 [38]. D. B. Rubin and D. T. Thayer applied log-EM algorithm for maximum likelihood factor model analysis in 1982 [107]. After that, a number of methods have been proposed to accelerate the sometimes slow convergence of the EM algorithm, such as those utilizing conjugate gradient and modified Newton–Raphson techniques. Xiaoli Meng and D. B. Rubin introduced a class of generalized EM algorithms which they call the ECM algorithm in 1993 [88]. Expectation conditional maximization (ECM) replaces each M step with a sequence of conditional maximization (CM) steps in which each parameter is maximized individually, conditionally on the other parameters remaining fixed. Chuanhai Liu and Donald B. Rubin introduced a simple extension of EM and ECM with faster covergence in 1994 [72], which they call the ECME algorithm. They applied ECEM algorithm for maximum likelihood estimation of factor analysis in 1998 [74]. This idea is further extended in the generalized expectation maximization (GEM) algorithm, in which one only seeks an increase in the objective function for both the E step and M step under the alternative description. The Q-function used in the EM algorithm is based on the log-likelihood. Thus in this thesis we call it the log-EM algorithm.

The EM algorithm has also been widely used for fitting the MFA models. It is easy to implement and converges stably since its M-step is in closed form. Zoubin Ghahramani and Geoffrey E. Hinton showed how to use the log-EM algorithm for both single factor analysis and mixture of factor analyzers in 1996 [51]. Later, G. J. McLachlan, D. Peel and R.W. Bean used a mixture of factor analysers to model high-dimensional data [85] and fitted the model by using the alternating expectation-conditional maximization (AECM) algorithm. After that, G. J. McLachlan, R.W. Bean and L. Ben-Tovim Jones extended the mixture of factor analysers in order to incorporate the multivariate t-distribution [86]. However, missing data of MFA models contains indicator factors and also latent factors. Because of so much missing data, the convergence of the EM algorithm for MFA can be painfully slow due to the fact that the rate of convergence of EM is determined by the portion of missing information in complete data [38]. In order to deal with the missing data, Jian-Hua Zhao and Philip L. H. Yu proposed a fast expectation conditional maximization (ECM) algorithm for maximum-likelihood (ML) estimation of mixture of factor analysers (MFA) [129]. The convergence of ECM is substantially faster than EM and AECM regardless of whether they are assessed by CPU time or number of iterations. To further reduce the amount of missing data, Jangsun Baek, G. J. McLachlan, and Lloyd K. Flack proposed the use of common component-factor loadings which considerably reduces the number of parameters. They applied this new method to the clustering and visualization of high-dimensional data in 2010 [3].

The α-EM algorithm was introduced by Yasuo Matsuyama [78], [79], [80], [81], who also proved the covergence speed of the α-EM algorithm is faster than the log-EM algorithm as long as the incomplete data comes from an exponential family. Logarithms have important roles besides simplifying the likelihood maximization.

In information measures, logarithmic is correspond to the Kullback-Leibler divergence which is a key for realizing the maximization transfer in the EM algorithm [38]. The α-EM algorithm is derived by the maximization transfer which uses more general surrogate functions than log-EM. The use of the log-likelihood ratio can be generalized to that of the α-log-likelihood ratio. The log-EM corresponds to the special case of $\alpha = -1$. Yasuo Matsuyama also applied α-EM to clustering. His results showed that it is better than the log-EM algorithm in terms of both the number of iterations and the total computation time. In 2010 and 2011 Yasuo Matsuyama applied α-EM algorithm to hidden Markov model estimation [82], [83]. It had been expected that the α-EM for factor model estimation would exist. On one hand, the complete data of factor model comes from an exponential family, so theoretically α-EM can be applied to factor analysis. On the other hand, the convergence speed of log-EM for factor models can be slow when the problem is not well conditioned. For applications such as high frequency trading, problems may be ill condition and require fast computation. Since the log-EM is a subclass of the α-EM, the α-EM can only do better than the log-EM. In practice there are several hurdles when it cares to implement the α-EM for factor model. In this thesis, we present a way to use α-EM for factor model estimation.

The conjugate gradient method is an algorithm for the numerical solution of particular systems of linear equations, namely those whose matrix is symmetric and positive-definite. Because covariance structure analyses can usually require optimization of functions with a large numbers of parameters, they often lead to expensive computer runs. Algorithms which are computational efficient and don't require a large amount of memory are welcomed for such analyses. The conjugate gradient method is a commonly used method for EM acceleration. Mortaza Jamshidian and Robert I. Jennrich showed that the conjugate gradient method can fulfill both of those requirements for factor analysis [60]. Later, Mortaza Jamshidian and Robert I. Jennrich showed that the EM step can be viewed as a generalized gradient, making it natural to apply generalized conjugate gradient methods in an attempt to accelerate the EM algorithm. They considered its application to several problems, such as estimation of a covariance matrix from incomplete multivariate normal data, confirmatory factor analysis and repeated measures analysis [59]. Ruslan Salakhutdinov, Sam Roweis and Zoubin Ghahramani presented a close relationship between EM algorithm and direct optimization approaches such as gradient-based methods for parameter learning [109]. After that, Ruslan Salakhutdinov, Sam Roweis and Zoubin Ghahramani presented an Expectation-Conjugate-Gradient (ECG) algorithm for maximum likelihood estimation in latent variable models, and showed that it can outperform EM in terms of convergence in certain cases [110].

Many rank-structured matrices have been widely used in developing new fast matrix algorithms. Andrew P. Mullhaupt and Kurt Riedel introduced low grade matrices, their fraction representations, and consecutive sub-block product representations in 2002 [92]. They applied their results to signal processing using banded matrix fraction representations of triangular input normal pairs in 2001 [93]. Other various efficient representations for rank structured matrices have been proposed, and efficient and accurate algorithms have been developed using these representations. In particular, semiseparable matrices are very useful. S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A. J. van der Veen, and D. White generalized the hierarchically semiseparable (HSS) representations and propose some fast algorithms for HSS matrices in 2005 [18]. Those algorithms are useful for problems where off-diagonal blocks have small numerical ranks. S. Delvaux and M. Van Barel they investigated some matrix structures that are preserved by Schur complementation in 2006 [34]. After that, they introduced a given-weight representation for rank-structured matrices where the rank structure is defined by certain submatrices starting from the bottom left or upper right corner of the matrix [35]. There are several efficient algorithms that have been developed for approximating a symmetric matrix A by a symmetric semiseparable matrix, accurate to a constant multiple of any given tolerance $\tau > 0$ [12], [53]. Fast backward stable algorithms have also been constructed to approximate A with an SPD semiseparable matrix [119]. Ming Gu, Xiaoye S. Li and Panayot S. Vassilevski solved the problem of constructing semiseparable SPD matrices to approximate a given dense SPD matrix A for a given tolerance $\tau > 0$ which is very large [55].

The rest of this dissertation is structured as follows. Section 2 reviews factor model, mixture of factor models and usual model estimation methods, such as EM and its conjugate gradient acceleration method. The α-EM algorithm is also presented. Section 3 shows how to use the α-EM algorithm in factor model estimation and this is new in the literature. Comparison between the log-EM and the α-EM are presented through real financial data. Section 4 focuses on mixture of factor analysers. Section 5 introduces conjugate gradient acceleration method to the α-EM algorithm. Section 6 presents a new structure for covariance estimates which is better than factor model under the Hellinger distance.

2 Models Description and Estimation Methods Review

2.1 Factor Model

Given a set of p observable random variables, $x_1, x_2, \ldots, x_p$ with means, $\mu_1, \mu_2, \ldots, \mu_p$, a factor model consists of an (unobserved) factor loading matrix $\Lambda_{p \times d}$ and d (unobserved) factor-scores $z_1, z_2, \ldots, z_d$ for each observation. We require $d < p$.We have:

$$x_i - \mu_i = \Lambda_{i,1} z_1 + \Lambda_{i,2} z_2 + \cdots + \Lambda_{i,d} z_d + u_i \ , \ 1 \leq i \leq p$$

where u_i are independently distributed error terms with zero mean and finite variance, which may not be the same for all i. Let X be the $p \times n$ data matrix with zero mean and Z be the $d \times n$ unobserved factor-score matrix corresponding to n observations.

In matrix terms the generative model is given by:

$$X = \Lambda Z + U$$

We also will impose the following assumptions on Z and U.

1. Z and U are independent.

2. $E[Z] = 0$ and $Cov[Z] = I$ (to make sure that the factors are uncorrelated)

3. $E[U] = 0$ and $Cov[U] = Diag(\phi_1, \phi_2, \ldots, \phi_n) \overset{def}{=} \Phi$

Suppose $Cov[X] = \Sigma$ and we have $Cov[X] = Cov[\Lambda Z + U]$, so we can get $\Sigma = \Phi + \Lambda\Lambda'$. Estimation of the factor model means that given observed data X we need to estimate Φ and Λ.

Note that for any orthogonal matrix Q if we set $\Lambda^* = \Lambda Q$ and $Z^* = Q'Z$, the criteria for being factors and factor loadings still holds. Hence a set of factors and factor loadings is identical only up to orthogonal transformations.

2.2 Mixture of Factor Models

Assume now that the data are from a mixture of k factor models indexed by $w_j, j = 1, \ldots, k$. The distribution of the observation x_i can be modeled as:

$$x_i - \mu_i^{(j)} = \Lambda_{i,1}^{(j)} z_1^{(j)} + \Lambda_{i,2}^{(j)} z_2^{(j)} + \cdots + \Lambda_{i,d}^{(j)} z_d^{(j)} + u_i^{(j)} \text{ with prob. } \pi_j (j = 1, \ldots, k)$$

where $u_i^{(j)}$ are independently distributed error terms with zero mean and finite variance, which may not be the same for all i and j.

The generative model now obeys the following mixture distribution:

$$P(x) = \sum_{j=1}^{m} \int P(x|z, w_j) P(z|w_j) P(w_j) dz$$

We have the same assumptions as in regular factor analysis, the factors z are all assumed to be $N(0, 1)$. We have:

$$P(x|z, w_j) = N(\Lambda_j z, \Phi_j)$$

and the jth component-covariance matrix Σ_j has the form

$$\Sigma_j = \Phi_j + \Lambda_j \Lambda_j' \ (j = 1, \ldots, k)$$

where Λ_j is $p \times d$ matrix of factor loadings and Φ_j is $p \times p$ diagonal matrix along with the mixing proportion $\pi_j (j = 1, \ldots, k)$.

The parameters of this model are $\{\Lambda_j, \Phi_j, \pi_j\}_{j=1}^{k}$ where $\pi_j = P(w_j = 1)$. The latent variables in this model are the factors Z and the mixture indicator variable w_j, where $w_j = 1$ when the data point was generated by w_j. So for mixture of factor models we need to estimate $\{\Lambda_j, \Phi_j, \pi_j\}_{j=1}^{k}$.

2.3 Maximum Likelihood estimators

Maximum-likelihood estimation (MLE) is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model's parameters. Intuitively, it finds the parameter point for which the observed sample is most likely to appear. Suppose $X = (X_1, \ldots, X_n)$ is an i.i.d. sample from a distribution with pdf or pmf $f(x; \theta)$, the likelihood function is defined as:

$$L(\theta|X) = \prod_{i=1}^{n} f(x_i; \theta)$$

In practice, it is often more convenient to work on the log-likelihood:

$$l(\theta|X) = \log L(\theta|X) = \sum_{i=1}^{n} \log f(x_i; \theta)$$

Then, the maximum likelihood estimate is obtained as:

$$\widehat{\theta} = \arg\max_{\theta \in \Theta} l(\theta|X)$$

As the sample size increases to infinity, sequences of maximum-likelihood estimators have these properties:

- Consistency: the sequence of MLEs converges in probability to the value being estimated.

$$\widehat{\theta}_{MLE} \xrightarrow{p} \theta_0$$

- Asymptotic normality: as the sample size increases, the distribution of the MLE tends to the Gaussian distribution with mean θ and covariance matrix equal to the inverse of the Fisher information matrix.

$$\sqrt{n}(\theta'_{MLE} - \theta_0) \xrightarrow{d} N(0, I(\theta)^{-1})$$

- Functional invariance: if $\widehat{\theta}$ is the MLE of θ, then for any function $g(\theta)$, the MLE of $g(\theta)$ is $g(\widehat{\theta})$. For example, the MLE parameters of the log-normal distribution are the same as those of the normal distribution fitted to the logarithm of the data.

- Efficiency: it achieves the Cramér–Rao lower bound when the sample size tends to infinity. This means that no consistent estimator has lower asymptotic mean squared error than the MLE (or other estimators attaining this bound).

2.4 Log-EM Algorithm

The EM algorithm ([38], [107]) is a very popular and widely applicable algorithm for the computation of maximum likelihood estimation. Given a statistical model consisting of observed data X, unobserved latent data Z and unknown parameters Φ, Λ, along with log-likelihood function, the maximum likelihood estimate of the unknown parameters is determined by the marginal likelihood of the observed data. Then the incomplete data log-likelihood is $L(X, \Phi, \Lambda) = \log \prod_i^n p(x_i|\Phi, \Lambda)$. On the other hand, the complete data log-likelihood is $L_C(X, Z, \Phi, \Lambda) = \log \prod_i^n p_C(x_i, z_i|\Phi, \Lambda)$. The log-EM algorithm seeks to find the maximization of the marginal likelihood by iteratively applying the following two steps: (The subscripts of Φ_0, Λ_0 mean the current estimates and the subscripts of Φ_1, Λ_1 mean the next estimates)

- Expectation step (E step): Calculate the expected value of the log likelihood function, with respect to the conditional distribution of z given x under the current estimate of the parameters Φ_0, Λ_0

$$Q(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0) = E_{Z|X,\Phi_0,\Lambda_0}[L_C(X, Z, \Phi, \Lambda)]$$

- Maximization step (M step): Find the parameter that maximizes this quantity:

$$\Phi_1, \Lambda_1 = \arg \max_{\Phi_1,\Lambda_1} Q(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0)$$

Given Φ_0, Λ_0 and x_i, i means the ith observation, the expected value of the factors z_i can be computed and this computation is in fact necessary for log-EM algorithm. For the distribution of the observed variable $p(x_i)$ we have $E[x_i] = 0$ and $Cov[X] = \Sigma = \Phi_0 + \Lambda_0\Lambda_0'$. For the distribution of the complete data $p(x_i, z_i)$, let $y_i = \begin{bmatrix} x_i \\ z_i \end{bmatrix}$ and $Y = \begin{bmatrix} X \\ Z \end{bmatrix}$, we have $E[y_i] = 0$ and $Cov[Y] = \begin{bmatrix} \Phi_0 + \Lambda_0\Lambda_0' & \Lambda_0 \\ \Lambda_0 & I \end{bmatrix}$. For the distribution $p(z_i|x_i)$ we have $p(z_i|x_i) = \frac{p(x_i,z_i)}{p(x_i)}$. Since we know the distribution of $p(x_i)$ and $p(x_i, z_i)$, we will have $E[z_i|x_i] = \beta x_i$ and $Var[z_i|x_i] = C$ where (see appendix A for proof):

$$\begin{aligned} \beta_0 &= \Lambda_0'(\Phi_0 + \Lambda_0\Lambda_0')^{-1} \\ C_0 &= I - \Lambda_0'(\Phi_0 + \Lambda_0\Lambda_0')^{-1}\Lambda_0 \end{aligned}$$

2.5 α-EM Algorithm

Yasuo Matsuyama devised the α-EM algorithm ([78],[79],[80],[81]), which generalizes the EM-algorithm, with application to some model estimation. In this thesis, we adapt the α-EM algorithm to estimation of factor models.
The α-logarithm function is defined as follows [81]:

$$L^{(\alpha)}(r) \overset{def}{=} \frac{2}{1+\alpha}\left(r^{\frac{1+\alpha}{2}} - 1\right) \tag{1}$$

where $r \in (0, \infty)$. $L^{(\alpha)}(r)$ is strictly concave for $\alpha < 1$, a straight line $r - 1$ for $\alpha = 1$ and strictly convex for $\alpha > 1$. Especially when $\alpha = -1$ we have $L^{(-1)} = \log(r)$. The α-EM algorithm maximizes the α-logarithm of the likelihood ratio, which in the special case $\alpha = -1$ corresponds to ordinary maximum likelihood.

Let $P_I(X|\Phi, \Lambda)$ be the probability density for the observed(incomplete) data X parameterized by Φ and Λ. Let $P_C(X, Z|\Phi, \Lambda)$ be the probability density for the complete data. Then the incomplete data α-log-likelihood ratio is:

$$L_X^{(\alpha)}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0) = \frac{2}{1+\alpha}\left[\left(\frac{P_I(X|\Phi_1, \Lambda_1)}{P_I(X|\Phi_0, \Lambda_0)}\right)^{\frac{1+\alpha}{2}} - 1\right]$$

On the other hand, the complete data α-log-likelihood ratio is :

$$L_{X,Z}^{(\alpha)}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0) = \frac{2}{1+\alpha}\left[\left(\frac{P_C(X, Z|\Phi_1, \Lambda_1)}{P_C(X, Z|\Phi_0, \Lambda_0)}\right)^{\frac{1+\alpha}{2}} - 1\right]$$

by taking the conditional expectation in terms of $P_{Z|X,\Phi_0,\Lambda_0}$ we can get

$$Q_{X,Z|X}^{(\alpha)}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0) = E\left[L_{X,Z}^{(\alpha)}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0)\right]$$

by computing the α-divergence between $P_{Z|X,\Phi_0,\Lambda_0}(Z|X,\Phi_0,\Lambda_0)$ and $P_{Z|X,\Phi_1,\Lambda_1}(Z|X,\Phi_1,\Lambda_1)$ we have the following basic equality for the α-EM algorithm [78], [79], [80], [81].

$$L_X^{(\alpha)}(\Phi_1,\Lambda_1|\Phi_0,\Lambda_0) = Q_{X,Z|X}^{(\alpha)}(\Phi_1,\Lambda_1|\Phi_0,\Lambda_0) + \frac{1-\alpha}{2}\left\{\frac{P(X|\Phi_1,\Lambda_1)}{P(X|\Phi_0,\Lambda_0)}\right\}^{\frac{1+\alpha}{2}} D^{(\alpha)}(\Phi_1,\Lambda_1||\Phi_0,\Lambda_0) \tag{2}$$

Therefore, the α-log likelihood ratio of the observed data can be expressed by using the Q-function of the α-log likelihood ratio and the α-divergence. The α-divergence is an information measure. When $\alpha = \pm 1$, it is the Kullback-Leibler divergence. When $\alpha = 0$, it is the well known the Hellinger distance. Equation (2) is the core of the α-EM algorithm. The second term on right-hand side is nonnegative for $\alpha < 1$, this also ensures positivity of the α-information matrix. So the algorithm to increase $L_X^{(\alpha)}(\Phi_1,\Lambda_1|\Phi_0,\Lambda_0)$ is obtained by increasing the $Q_{X,Z|X}^{(\alpha)}(\Phi_1,\Lambda_1|\Phi_0,\Lambda_0)$ function with respect to the argument Φ_1 and Λ_1.

Obtaining the $Q_{X,Z|X}^{(\alpha)}(\Phi_1,\Lambda_1|\Phi_0,\Lambda_0)$ function is a generalized E step. Its maximization is a generalized M step. This pair of steps is called the α-EM algorithm which contains the log-EM algorithm as its subclass. Thus, the α-EM algorithm by Yasuo Matsuyama is an exact generalization of the log-EM algorithm. The α-EM shows faster convergence than the log-EM algorithm by choosing an appropriate α. For possible choices of α, we already have $\alpha < 1$, and on the other hand the α-EM requires $\alpha > -1$ for the exponential family.

2.6 Conjugate Gradient Method

The conjugate gradient method is an iterative method. The EM algorithm is designed to find a parameter vector θ that maximizes a likelihood function $L(\theta), \theta \in \Theta$. For a specific application, a function $Q(\theta', \theta)$ is identified. It may be viewed as a local approximation to $\log L(\theta')$ in a neighborhood of θ. Let θ' be the value that maximizes $Q(\theta', \theta)$ then the step $\theta' - \theta$ is called an EM step. If θ' is an interior point of Θ, then

$$\theta' - \theta = -(Q^{(2)}(\theta', \theta'))^{-1} g(\theta) + o(\theta - \theta') \tag{3}$$

where $g(\theta)$ is the gradient of $\log L(\theta)$ at θ and $Q^{(2)}(\theta', \theta')$ is the Hessian of $Q(\theta', \theta)$ viewed as a function of θ' and evaluated at $(\theta', \theta) = (\theta', \theta')$. Typically $Q^{(2)}(\theta', \theta')$ is negative definite. Thus using (3) the EM step $\theta' - \theta$ is a generalized gradient of $\log L(\theta)$. Each step begins with an EM step. First, its direction is modified and then its length is optimized. In addition to the EM steps, we must compute gradients $g(\theta)$ of the $\log L(\theta)$. In order to find the optimal length, we need to do a line search [59]. This is the most complicated part.

Given θ_0, let $g_0 = \nabla f(\theta_0)$, and $d_0 = -g_0$, set $k = 0$, the generalized conjugate gradient algorithm proceeds as follows [59]:

1 α_k, the value of α that maximizes $f(\theta_k + \alpha_k d_k)$

2 $\theta_{k+1} = \theta_k + \alpha_k d_k$ and $g_{k+1} = \nabla f(\theta_{k+1})$

3 $\beta_k = \langle g_{k+1}, g_{k+1} - g_k \rangle / \langle d_k, g_{k+1} - g_k \rangle$

4 $d_{k+1} = -g_{k+1} + \beta_k d_k$

Here, $\langle \cdot, \cdot \rangle$ is the inner product. The best know formulas for β_k are called the Fletcher-Reeves (FR), Polak-Ribiere (PR) and Hestense-Stiefel (HS) formulas and are given by

$$\begin{aligned}
\beta_k^{FR} &= \|g_{k+1}\|^2 / \|g_k\|^2 \\
\beta_k^{PR} &= \langle g_{k+1}, g_{k+1} - g_k \rangle / \|g_k\|^2 \\
\beta_k^{HS} &= \langle g_{k+1}, g_{k+1} - g_k \rangle / \langle d_k, g_{k+1} - g_k \rangle
\end{aligned}$$

The numerical performance of the FR method is somewhat erratic and it is sometime as efficient as the PR and HS methods, but it is often much slower. The PR and HS methods appear to perform very similarly in practice, and are to be preferred over the FR method.

2.7 Band Fraction Representation (Model)

Factor models are popular models for structured covariance estimates, but they are not the only ones. In the literature we have Toeplitz, band-Toeplitz, Circulant, Semiseparable matrices ([20], [29], [18], [17], [55]) which are also structured covariances. In this thesis, we study semiseparable matrices which contain factor models as a special case (proof is in the following section). We then apply band fraction representations ([93], [18]) to covariance matrices estimation.

Theorem 1 (Mullhaupt, Riedel 2002)**.** *Suppose L is a lower triangular matrix with lowgrade $\leq d-1$. For any $\varepsilon > 0$, there exists M and N which are also lower triangular matrix with bandwidth $\leq d$. s.t. $\left\|L - M^{-1}N\right\| < \varepsilon$.*

Let's assume Σ is the covariance matrix that we get from some given data X. Since Σ is a positive definite matrix, we can apply Cholesky decomposition which is a decomposition of positive definite matrix into the product of a lower triangular matrix and its transpose:

$$\Sigma = LL^T \tag{4}$$

where L is a lower triangular matrix with real and positive diagonal entries. Every positive definite matrix has a unique Cholesky decomposition.

Then we get $\Sigma = LL^T \approx (M^{-1}N)(M^{-1}N)^T$, so the covariance matrix Σ can be approximate by M and N. This is the band fraction representation of covariance matrix. In order to get this representation we need to estimate M and N. Note that for any nonsingular matrix X we will have the equivalent transform:

$$\begin{aligned} ML &= N \Leftrightarrow (XM)L = XN \\ NL^{-1} &= M \Leftrightarrow (XN)L^{-1} = XM \end{aligned}$$

There are some reasons that we think band fraction is better than factor model. For covariance matrix Σ, there are two important properties that factor models don't preserve but band fraction representations do. First, Σ^{-1}, the inverse of covariance matrix Σ, is still symmetric, positive definite and could be a covariance matrix. But the inverse of a factor model is no longer a factor model. When we use the covariance matrix we are always concerned with its inverse. From the Woodbury formula we have:

$$\begin{aligned}(\Phi + \Lambda\Lambda')^{-1} &= \Phi^{-1} - \Phi^{-1}\Lambda(I_d + \Lambda'\Phi^{-1}\Lambda)^{-1}\Lambda'\Phi^{-1} \\ &= \Phi_1 - \Lambda_1\Lambda_1'\end{aligned}$$

we will notice that there is a sign change in the inverse of the factor model. Meanwhile band fraction representation doesn't have this problem, $(M^{-1}N)^{-1} = N^{-1}M$, it is still a band fraction representation. Second, Σ is in a convex set. Suppose $0 < t < 1$, let's consider a convex combination of Σ_1 and Σ_2. We have $t\Sigma_1 + (1-t)\Sigma_2 = \Sigma_3$ and we can always find Σ_3 from the same set contains Σ_1 and Σ_2. Now, let's consider a convex combination of two factor models Φ_1, Λ_1 and Φ_2, Λ_2. Even though $t\Phi_1 + (1-t)\Phi_2$ preserves positivity and diagonality, but $t\Lambda_1 + (1-t)\Lambda_2$ does not preserve rank. If Λ_1 and Λ_2 don't have the same span then you can't find Λ_3 from the set that Λ_1 and Λ_2 are in. However, band fraction representation doesn't have this problem either. Let's consider a convex combination of two band fraction representations M_1, N_1 and M_2, N_2. Then we get $t(M_1, N_1) + (1-t)(M_2, N_2) = (M_3, N_3)$ which preserves triangularity, bandwidth and nonsingularity. We can find M_3, N_3 from the same set contains M_1, N_1 and M_2, N_2. Therefore the set of parameters (M, N) with d-bandwidth for Cholesky factors is a convex set, and that means optimizing a convex function over this set is a 'nice' problem.

2.8 Semiseparable Factorization (Method)

A matrix S is called a lower-(upper-) semiseparable matrix of semiseparability rank d if all submatrices which can be taken out of the lower(upper) triangular part of the matrix S have rank $\leq d$. Semiseparable matrix has a classical representation.

Classical The symmetric semiseparable matrix is represented with two vectors $u = [u_1, u_2, \ldots, u_n]^T$ and $v = [v_1, v_2, \ldots, v_n]^T$. For example a 5×5 matrix has the following from:

$$\text{lower triangular}(u * v^T) = \begin{bmatrix} u_1v_1 & & & & \\ u_2v_1 & u_2v_2 & & & \\ u_3v_1 & u_3v_2 & u_3v_3 & & \\ u_4v_1 & u_4v_2 & u_4v_3 & u_4v_4 & \\ u_5v_1 & u_5v_2 & u_5v_3 & u_5v_4 & u_5v_5 \end{bmatrix}$$

Given a factor model $\Sigma = \Phi + \Lambda\Lambda^T$ where Φ is $p \times p$ and Λ is $p \times d$. $\Lambda\Lambda^T$ is actually a classical representation of a semiseparable matrix with semiseparability rank d. One important property of semiseparable matrix is that any diagonal matrix plus a semiseparable matrix is still a semiseparability, but the semiseparability rank will increase by 1. So factor model is actually a semiseparable matrix with semiseparability rank $d+1$, where d is the number of factors. But not all the semiseparable matrices can be represented by the classical representation which means that factor model can't represent all the semiseparable matrices. That's part of the reason why we use band fraction representation instead of factor model. We will prove in the following section that any semiseparable matrices can be represented by band fraction representation.

To begin the semiseparable factorization procedure, we first recall the following standard block Cholesky factorization procedure. For $k = 1, 2, \ldots, n$

1 Cholesky factorize $R_{k,k}^T R_{k,k} = A_{k,k}$

2 Compute $R_{k,k+1:n} = R_{k,k}^{-T} A_{k,k+1:n}$

3 Schur complement $A_{k+1:n,k+1:n} = A_{k+1:n,k+1:n} - R_{k,k+1:n}^T R_{k,k+1:n}$

The output of this procedure is the upper triangular matrix

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,n} \\ & R_{2,2} & \cdots & R_{2,n} \\ & & \ddots & \vdots \\ & & & R_{n,n} \end{bmatrix} \text{ such that } A = R^T R$$

Ming Gu et al. [55] modified the above procedure in order to find an approximate Cholesky factorization satisfying

$$S^T S = A + o(\sqrt{\|A\|_2} \tau) \text{ and } S^T S Z = AZ$$

where

$$Z = \begin{pmatrix} Z_1 \\ \vdots \\ Z_n \end{pmatrix}$$

and S is an upper triangular semiseparable matrix with semiseparability rank 1 of the form:

$$S = \begin{bmatrix} D_1 & S_{1,2} & \cdots & S_{1,n} \\ & D_2 & \cdots & S_{2,n} \\ & & \ddots & \vdots \\ & & & D_n \end{bmatrix}$$

with the D_k's being upper triangular and $S_{k,t} = U_k W_{k+1} \cdots W_{t-1} V_t^T$.

The method above is an efficient and backward stable algorithm for constructing SPD semiseparable matrices that approximate a given dense SPD matrix A with a guaranteed a priori given tolerance $\tau > 0$. We generalized that procedure to be able to get S with different semiseparability rank from 1 to d. With S_d we can find M and N with bandwidth d. We prove it in section 6.

3 Factor Model Estimation By Using The α-EM Algorithm

We apply the α-EM algorithm to factor model estimation. The α-EM includes the traditional log-EM as a special case. For estimation of other models, it has been shown that the convergence speed of the α-EM algorithm is much faster than log-EM algorithm, we investigate this for factor models and mixture of factor models. The α-EM algorithm also allows us to choose different αs to achieve the fastest convergence speed and more accurate factor model estimation for different problems. In practice the update equations from the α-EM algorithm are not tractable so we apply causal approximation and series expansion to those update equations to get practical update equations. With these update equations we can show that the α-EM algorithm can save us in total computation time. Empirical results from real financial data are given.

3.1 Non-Causal Update Equations

Here, by non-causal we mean that given the current estimations we can not use the update equations to get the next estimations directly. The non-causal update equations are the most accurate equations you can get after applying the α-EM algorithm for factor model estimation. In order to be able to use the α-EM we use a causal approximation of non-causal update equations.

For factor model, we have

$$P_C(X, Z|\Phi_0, \Lambda_0) = \prod_{i=1}^{N} P_c(x_i, z_{i,}|\Phi_0, \Lambda_0) = \prod_{i=1}^{N} P(x_i|z_i, \Phi_0, \Lambda_0) * P(z_i)$$

so the $Q^{(\alpha)}_{X,Z|X}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0)$ function is:

$$\begin{aligned}
Q^{(\alpha)}_{X,Z|X}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0) &= E_{P(Z|X,\Phi_0,\Lambda_0)}[L^{(\alpha)}_{X,Z}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0)] \qquad (5) \\
&= E_{P(Z|X,\Phi_0,\Lambda_0)}\left[\frac{2}{1+\alpha}\left(\frac{P_C(X,Z|\Phi_1,\Lambda_1)}{P_C(X,Z|\Phi_0,\Lambda_0)}\right)^{\frac{1+\alpha}{2}} - 1\right] \\
&= \frac{2}{1+\alpha}\left(\prod_{i=1}^{N} E_{P(z_i|x_i,\Lambda_0,\Phi_0)}\left[\left(\frac{Pc(x_i,z_i|\Lambda_1,\Phi_1)}{Pc(x_i,z_i|\Lambda_0,\Phi_0)}\right)^{\frac{1+\alpha}{2}}\right] - 1\right) \\
&= \frac{2}{1+\alpha}\left(S^{(\alpha)}_{Z|X,\Phi_0,\Lambda_0} - 1\right)
\end{aligned}$$

where

$$\begin{aligned}
S^{(\alpha)}_{Z|X,\Phi_0,\Lambda_0} &= \prod_{i=1}^{N} W_i^{(\alpha)} \\
W_i^{(\alpha)} &= E\left[P_i^{\frac{1+\alpha}{2}}\right] \\
P_i &= \frac{Pc(x_i,z_i|\Lambda_1,\Phi_1)}{Pc(x_i,z_i|\Lambda_0,\Phi_0)} \\
E[\cdot] &= E_{P(z|x,\Lambda_0,\Phi_0)}[\cdot]
\end{aligned}$$

After the E-step we need to do the M-step. The update equations can be obtained by differentiating $Q^{(\alpha)}_{X,Z|X}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0)$ with respect to the update parameters Φ_1 and Λ_1 and setting differentiation to zero solve for maximization. For Λ_1 we have

$$\begin{aligned}
\frac{\partial Q^{(\alpha)}}{\partial \Lambda_1} &= 0 \Rightarrow \frac{\partial S^{(\alpha)}}{\partial \Lambda_1} = 0 \qquad (6) \\
\frac{\partial S^{(\alpha)}}{\partial \Lambda_1} &= \sum_{j=1}^{N} \frac{\partial W_j^{(\alpha)}}{\partial \Lambda_1} \prod_{i=1, i\neq j}^{N} W_i^{(\alpha)} = \sum_{j=1}^{N} \frac{\partial W_j^{(\alpha)}}{\partial \Lambda_1}\frac{S^{(\alpha)}}{W_j^{(\alpha)}} = \sum_{j=1}^{N} \frac{\frac{\partial W_j^{(\alpha)}}{\partial \Lambda_1}}{W_j^{(\alpha)}} S^{(\alpha)} \\
S^{(\alpha)} &\neq 0 \Rightarrow \sum_{i=1}^{N} \frac{\frac{\partial W_i^{(\alpha)}}{\partial \Lambda_1}}{W_i^{(\alpha)}} = 0 \Rightarrow \sum_{j=1}^{N} \frac{\frac{\partial E\left[P^{\frac{1+\alpha}{2}}\right]}{\partial \Lambda_1}}{E\left[P^{\frac{1+\alpha}{2}}\right]} = \sum_{j=1}^{N} \frac{E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Lambda_1}\right]}{E\left[P^{\frac{1+\alpha}{2}}\right]} = 0
\end{aligned}$$

and likewise for Φ_1 we have

$$\frac{\partial Q^{(\alpha)}}{\partial \Phi_1^{-1}} = 0 \Rightarrow \frac{\partial S^{(\alpha)}}{\partial \Phi_1^{-1}} = 0 \Rightarrow \sum_{j=1}^{N} \frac{\frac{\partial E\left[P^{\frac{1+\alpha}{2}}\right]}{\partial \Phi_1^{-1}}}{E\left[P^{\frac{1+\alpha}{2}}\right]} = \sum_{j=1}^{N} \frac{E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Phi_1^{-1}}\right]}{E\left[P^{\frac{1+\alpha}{2}}\right]} = 0 \qquad (7)$$

In order to solve equations (6) and (7) we need to calculate $E\left[P^{\frac{1+\alpha}{2}}\right]$, $E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial\Lambda_1}\right]$ and $E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial\Phi_1^{-1}}\right]$. By the definition of expectation:

$$
\begin{aligned}
E\left[P^{\frac{1+\alpha}{2}}\right] &= \int P(z_i|x_{i,}\Lambda_0,\Phi_0)\left(\frac{Pc(x_i,z_i|\Lambda_1,\Phi_1)}{Pc(x_i,z_i|\Lambda_0,\Phi_0)}\right)^{\frac{1+\alpha}{2}} dz_i \qquad (8)\\
E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial\Lambda_1}\right] &= \frac{1+\alpha}{2}E\left[P^{\frac{1+\alpha}{2}} * \left(\Phi_1^{-1}x_iz_i' - \Phi_1^{-1}\Lambda_1 z_iz_i'\right)\right] \qquad (9)
\end{aligned}
$$

$$
E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial\Phi_1^{-1}}\right] = \frac{1+\alpha}{2}E\left[P^{\frac{1+\alpha}{2}} * \frac{1}{2}\left(\Phi_1 - x_ix_i' + x_iz_i'\Lambda_1' + \Lambda_1 z_ix_i' - \Lambda_1 z_iz_i'\Lambda_1'\right)\right] \tag{10}
$$

After calculating the expectations, we have the update equations (see Appendix B for details):

$$
\begin{aligned}
\Lambda_1 &= \sum_{i=1}^{N} x_iE[z_i']\left(\sum_{i=1}^{N} E[z_iz_i']\right)^{-1} \qquad (11)\\
\Phi_1 &= diag\left(\frac{1}{n}\left(\sum_{i=1}^{N} x_ix_i' - \sum_{i=1}^{N} x_iE[z_i']\Lambda_1'\right)\right) \qquad (12)
\end{aligned}
$$

However, the expectation here is w.r.t. a new distribution:

$$
\begin{aligned}
E[z_i] &= \Sigma W'x_i \Rightarrow E[z_i'] = x_i'W\Sigma\\
E[z_iz_i'] &= Var[z_i] + E[z_i]E[z_i]' = \Sigma + \Sigma W'x_ix_i'W\Sigma\\
\Sigma^{-1} &= \frac{1+\alpha}{2}\Lambda_1'\Phi_1^{-1}\Lambda_1 - \frac{1+\alpha}{2}\Lambda_0'\Phi_0^{-1}\Lambda_0 + C^{-1}\\
W &= \frac{1+\alpha}{2}\Phi_1^{-1}\Lambda_1 - \frac{1+\alpha}{2}\Phi_0^{-1}\Lambda_0 + \beta'C^{-1}
\end{aligned}
$$

if we assume the sample covariance is $C_{xx} = \sum_{j=1}^{N}\frac{x_ix_i'}{N}$, we get:

$$
\begin{aligned}
\Lambda_1 &= C_{xx}W\Sigma\left(\Sigma + \Sigma W'C_{xx}W\Sigma\right)^{-1} \qquad (13)\\
\Phi_1 &= diag(C_{xx-}C_{xx}W\Sigma\Lambda_1') \qquad (14)
\end{aligned}
$$

and we notice that we have Φ_1 and Λ_1 on both right and left hand sides of these update equations. It is hard to put either Φ_1 or Λ_1 on one side of the equations and this is what we mean by non-causal. So we can't use these update equations directly.

For general case $-1 < \alpha < 1$, the update equations:

$$\begin{aligned} \Lambda_1 &= F(\Lambda_1, \Phi_1, \Phi_0, \Lambda_0) && (15) \\ \Phi_1 &= G(\Lambda_1, \Phi_1, \Phi_0, \Lambda_0) && (16) \end{aligned}$$

are non-causal but they illustrate two important things. First, we can iteratively update Λ_1 and Φ_1 through (15) and (16) until Λ_1 and Φ_1 converge and we call it one major iteration. This major iteration is the iterations we count in the log-EM. Then replace Φ_0, Λ_0 with Λ_1, Φ_1, do the same thing for the next major iteration Λ_2, Φ_2. In practice, the convergence speed is much faster than the log-EM for the same major iteration. Second, each major iteration here contains many minor iterations which will take a large amount of time. Using this updating method, in practice, the α-EM algorithm can't save us in total computation time. Therefore, on one hand we know that α-EM is better than log-EM in convergence speed, on the other hand we need effective update equations otherwise we can't use the α-EM in practice.

Let's consider two special cases first:

Case 1. $\alpha = -1$:

$$\begin{aligned} \Sigma^{-1} &= C^{-1} \\ W &= \beta' C^{-1} \end{aligned}$$

then we have $W\Sigma = \beta' = (\Phi_0 + \Lambda_0\Lambda_0')^{-1}\Lambda_0$. Assume that

$$\begin{aligned} \delta_0 &= (\Phi_0 + \Lambda_0\Lambda_0')^{-1}\Lambda_0 \\ \Delta_0 &= I_d - \Lambda_0'(\Phi_0 + \Lambda_0\Lambda_0')^{-1}\Lambda_0 \end{aligned}$$

we get

$$\begin{aligned} \Lambda_1 &= C_{xx}\delta_0(\Delta_0 + \delta_0' C_{xx}\delta_0)^{-1} = f(\Lambda_0, \Phi_0) && (17) \\ \Phi_1 &= diag(C_{xx-}C_{xx}\delta_0\Lambda_1') = g(\Lambda_1, \Lambda_0, \Phi_0) && (18) \end{aligned}$$

these update equations are the same as Rubin and Thayer [107]. This shows that the log-EM algorithm is a special case of the α-EM algorithm.

Case 2. $\alpha = 1$:

$$\Lambda_1 = C_{xx}\delta_1(\Delta_1 + \delta_1' C_{xx}\delta_1)^{-1} = f(\Lambda_1, \Phi_1) \tag{19}$$

$$\Phi_1 = diag(C_{xx-}C_{xx}\delta_1\Lambda_1') = g(\Lambda_1, \Phi_1) \tag{20}$$

where

$$\delta_1 = (\Phi_1 + \Lambda_1\Lambda_1')^{-1}\Lambda_1$$

$$\Delta_1 = I_d - \Lambda_1'(\Phi_1 + \Lambda_1\Lambda_1')^{-1}\Lambda_1$$

here we have two equations and two unknown parameters Λ_1 and Φ_1 but it is impossible to solve for Λ_1 and Φ_1 directly. Because if we assume Λ_1 and Φ_1 are the optimal solutions, we have

$$C_{xx} = \Phi_1 + \Lambda_1\Lambda_1' \tag{21}$$

If we substitute (21) back to (19) and (20), we get:

$$\Lambda_1 = \Lambda_1$$

$$\Phi_1 = diag(\Phi_1) = \Phi_1$$

Since we can not solve (19) and (20), we can iteratively update Λ_1 and Φ_1 through (19) and (20) until Λ_1 and Φ_1 don't change. In practice this method takes exactly the same computation time as when $\alpha = -1$ because they have identical f and g. In order to have a practical solution we need to solve the non-causality.

3.2 Causal update equations

In order to solve the non-causality, we need to know why we have non-causality in the first place. The reason is the expectations, equation (8), (9) and (10). We need to calculate the three expectations $E\left[P^{\frac{1+\alpha}{2}}\right]$, $E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Lambda_1}\right]$ and $E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Phi_1^{-1}}\right]$ in a causal way. These three expectations have integral of the form:

$$\int P(z_i|x_{i,}\Lambda_0,\Phi_0)\left(\frac{Pc(x_i,z_i|\Lambda_1,\Phi_1)}{Pc(x_i,z_i|\Lambda_0,\Phi_0)}\right)^{\frac{1+\alpha}{2}} dz_i \tag{22}$$

in common and we need to calculate (22) without using Λ_1 and Φ_1.

3.2.1 Causal approximation

We have

$$\begin{aligned} P(z_i|x_{i,}\Lambda_0,\Phi_0)\left(\frac{Pc(x_i,z_i|\Lambda_1,\Phi_1)}{Pc(x_i,z_i|\Lambda_0,\Phi_0)}\right)^{\frac{1+\alpha}{2}} &\approx P(z_i|x_i,\Lambda_1,\Phi_1)\left(\frac{Pc(x_i,z_i|\Lambda_1,\Phi_1)}{Pc(x_i,z_i|\Lambda_0,\Phi_0)}\right)^{-\frac{1-\alpha}{2}} \\ &\approx P(z_i|x_i,\Lambda_0,\Phi_0)\left(\frac{Pc(x_i,z_i|\Lambda_0,\Phi_0)}{Pc(x_i,z_i|\Lambda_{-1},\Phi_{-1})}\right)^{-\frac{1-\beta}{2}} \end{aligned} \tag{23}$$

around the region of $P(x_i|\Lambda_1,\Phi_1) = P(x_i|\Lambda_0,\Phi_0) + o(1)$ and the last term is the causal approximation w.r.t. the iteration index shift or shift of time [82], [83]. We can use the relationship:

$$\frac{1+\alpha}{2} = -\frac{1-\beta}{2} \Rightarrow \beta = \alpha + 2 \tag{24}$$

and because we have $\alpha \in (-1,1)$, then we also have $\beta \in (1,3)$.

Within the first few iterations $\beta = \alpha + 2$ is not always a good approximation. For example, when $\alpha = 1$, $P_1/P_0 \approx P_0/P_{-1}$ is not a good approximation in the first a few iterations. This bad approximation will cause us numerical problems in practice. Another example is when $\alpha = 0$, $(P_1/P_0)^{1/2} \approx (P_0/P_{-1})^{1/2}$ is a better approximation during the first a few iterations since P_1/P_0 and P_0/P_{-1} are greater than 1. So for choice of α close to 1, we can choose β starting at 2 and approaching $\alpha + 2$ as iteration increases. We will see this is sometimes necessary in practice.

Now, we can approximately calculate (22) without knowing Λ_1, Φ_1. However, this requires a power computation of a likelihood ratio. This is computational expensive and becomes intractability as time increases. So another approximation is necessary in view of computational complexity.

3.2.2 Series Expansion

A Taylor expansion can simplify this without discarding merit of the α-log likelihood ratio.

$$P(z_i|x_i,\Lambda_0,\Phi_0)\left(\frac{Pc(x_i,z_i|\Lambda_0,\Phi_0)}{Pc(x_i,z_i|\Lambda_{-1},\Phi_{-1})}\right)^{-\frac{1-\beta}{2}} = P(z_i|x_i,\Lambda_{-1},\Phi_{-1})\left(\frac{Pc(x_i,z_i|\Lambda_0,\Phi_0)}{Pc(x_i,z_i|\Lambda_{-1},\Phi_{-1})}\right)^{\frac{1+\beta}{2}} \tag{25}$$

Let's assume that $f(x) = x^{\frac{1+\beta}{2}}$, according to Taylor expansion we have $f(x) = f(r) + \frac{f^{'}(r)}{1!}(x-r) + o(1)$ For our case $x = \frac{Pc(x_i,z_i|\Lambda_0,\Phi_0)}{Pc(x_i,z_i|\Lambda_{-1},\Phi_{-1})}$ and assume $r = 1$, so we get [82], [83]:

$$\left(\frac{Pc(x_i,z_i|\Lambda_0,\Phi_0)}{Pc(x_i,z_i|\Lambda_{-1},\Phi_{-1})}\right)^{\frac{1+\beta}{2}} \approx \frac{1-\beta}{2} + \frac{1+\beta}{2}\frac{Pc(x_i,z_i|\Lambda_0,\Phi_0)}{Pc(x_i,z_i|\Lambda_{-1},\Phi_{-1})} \tag{26}$$

now we substitute the right hand side of equation (23) with equation (25) and (26) then we get:

$$P(z_i|x_{i,}\Lambda_0,\Phi_0)\left(\frac{Pc(x_i,z_i|\Lambda_1,\Phi_1)}{Pc(x_i,z_i|\Lambda_0,\Phi_0)}\right)^{\frac{1+\alpha}{2}} \approx \frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1},\Phi_{-1}) + \frac{1+\beta}{2}P(z_i|x_i,\Lambda_0,\Phi_0)$$

So, now we can calculate the expectations in a causal way without using Λ_1 and Φ_1. We get the update equations:

$$\begin{aligned}
\Lambda_1 &= \frac{\left(\frac{1-\beta}{2}\sum_{j=1}^{N} x_i E_{-1}[z_i'] + \frac{1+\beta}{2}\sum_{j=1}^{N} x_i E_0[z_i']\right)}{\left(\frac{1-\beta}{2}\sum_{j=1}^{N} E_{-1}[z_i z_i'] + \frac{1+\beta}{2}\sum_{j=1}^{N} E_0[z_i z_i']\right)} && (27)\\
\Phi_1 &= diag\left(C_{xx} - C_{xx}\left(\frac{1-\beta}{2}\sum_{j=1}^{N} x_i E_{-1}[z_i'] + \frac{1+\beta}{2}\sum_{j=1}^{N} x_i E_0[z_i']\right)\Lambda_1'\right) && (28)
\end{aligned}$$

where

$$\begin{aligned}
E_{-1}[z_i] &= \beta_{-1}x_i \text{ and } E_{-1}[z_i z_i'] = C_{-1}^{-1} + \beta_{-1}x_i x_i'\beta_{-1}' \\
E_0[z_i] &= \beta_0 x_i \text{ and } E_0[z_i z_i'] = C_0^{-1} + \beta_0 x_i x_i'\beta_0'
\end{aligned}$$

with

$$\begin{aligned}
\beta_{-1} &= \Lambda'_{-1}(\Phi_{-1} + \Lambda_{-1}\Lambda'_{-1})^{-1} \\
C_{-1} &= I - \Lambda'_{-1}(\Phi_{-1} + \Lambda_{-1}\Lambda'_{-1})^{-1}\Lambda_{-1} \\
\beta_0 &= \Lambda'_0(\Phi_0 + \Lambda_0\Lambda'_0)^{-1} \\
C_0 &= I - \Lambda'_0(\Phi_0 + \Lambda_0\Lambda'_0)^{-1}\Lambda_0
\end{aligned}$$

For the first a few iterations $r = 1$ is not a very accurate estimation. We should choose r close to 1, so that:

$$\left(\frac{Pc(x_i, z_i|\Lambda_0, \Phi_0)}{Pc(x_i, z_i|\Lambda_{-1}, \Phi_{-1})}\right)^{\frac{1+\beta}{2}} \approx \frac{1-\beta}{2} r^{\frac{1+\beta}{2}} + \frac{1+\beta}{2} r^{\frac{1+\beta}{2}-1} \frac{Pc(x_i, z_i|\Lambda_0, \Phi_0)}{Pc(x_i, z_i|\Lambda_{-1}, \Phi_{-1})}$$

and

$$P(z_i|x_i,\Lambda_0, \Phi_0)\left(\frac{Pc(x_i, z_i|\Lambda_1, \Phi_1)}{Pc(x_i, z_i|\Lambda_0, \Phi_0)}\right)^{\frac{1+\alpha}{2}} \approx \frac{1-\beta}{2} r^{\frac{1+\beta}{2}} P(z_i|x_i, \Lambda_{-1}, \Phi_{-1}) + \frac{1+\beta}{2} r^{\frac{1+\beta}{2}-1} P(z_i|x_i, \Lambda_0, \Phi_0)$$

So the update equations are:

$$\Lambda_1 = \frac{\left(\frac{1-\beta}{2} r^{\frac{1+\beta}{2}} \sum_{j=1}^{N} x_i E_{-1}[z'_i] + \frac{1+\beta}{2} r^{\frac{1+\beta}{2}-1} \sum_{j=1}^{N} x_i E_0[z'_i]\right)}{\left(\frac{1-\beta}{2} r^{\frac{1+\beta}{2}} \sum_{j=1}^{N} E_{-1}[z_i z'_i] + \frac{1+\beta}{2} r^{\frac{1+\beta}{2}-1} \sum_{j=1}^{N} E_0[z_i z'_i]\right)} \tag{29}$$

$$\Phi_1 = diag\left(C_{xx} - C_{xx}\left(\frac{1-\beta}{2} r^{\frac{1+\beta}{2}} \sum_{j=1}^{N} x_i E_{-1}[z'_i] + \frac{1+\beta}{2} r^{\frac{1+\beta}{2}-1} \sum_{j=1}^{N} x_i E_0[z'_i]\right)\Lambda'_1\right) \tag{30}$$

Now we get two causal update equations. We are able to use these to do factor model estimation. At the kth iteration to obtain Λ_k and Φ_k, we use $\Lambda_{k-1}, \Phi_{k-1}$ and $\Lambda_{k-2}, \Phi_{k-2}$ on the right-hand sides of (27) and (28). We need to calculate $E[z]$ and $E[zz']$ of the previous state and the state prior to that. It seems like α-EM needs to do more calculation in each update but in practice we can save $E[z]$ and $E[zz']$ of the previous state for the next iteration, so we don't need to recalculate it. For example, for Λ_1 and Φ_1 we need to calculate $E_{-1}[z'_i]$ and $E_0[z'_i]$, when we calculate Λ_2 and Φ_2 we can reuse $E_0[z_i']$ and we only need to calculate $E_1[z_i']$. We only compute $E[z]$ and $E[zz']$ once for each state. This is exactly the same with log-EM. Thus, the total computation time per iteration of α-EM and log-EM are almost the same. We will see the numerical results in the following section.

3.3 Empirical Results

3.3.1 Factor analysis from complete observations

Here, we applied both the log-EM and the α-EM to the same data used in Rubin and Thayer (1982) [107] with $p = 9$ and $d = 4$.

$$C_{xx} = \begin{bmatrix} 1.0 & 0.554 & 0.227 & 0.189 & 0.461 & 0.506 & 0.408 & 0.280 & 0.241 \\ 0.554 & 1.0 & 0.296 & 0.219 & 0.479 & 0.530 & 0.425 & 0.311 & 0.311 \\ 0.227 & 0.296 & 1.0 & 0.769 & 0.237 & 0.243 & 0.304 & 0.718 & 0.730 \\ 0.189 & 0.219 & 0.769 & 1.0 & 0.212 & 0.226 & 0.291 & 0.681 & 0.661 \\ 0.461 & 0.479 & 0.237 & 0.212 & 1.0 & 0.520 & 0.514 & 0.313 & 0.245 \\ 0.506 & 0.530 & 0.243 & 0.226 & 0.520 & 1.0 & 0.473 & 0.348 & 0.290 \\ 0.408 & 0.425 & 0.304 & 0.291 & 0.514 & 0.473 & 1.0 & 0.374 & 0.306 \\ 0.280 & 0.311 & 0.718 & 0.681 & 0.313 & 0.348 & 0.374 & 1.0 & 0.672 \\ 0.241 & 0.311 & 0.730 & 0.661 & 0.245 & 0.290 & 0.306 & 0.672 & 1.0 \end{bmatrix}$$

To do the 1st iteration to obtain Λ_2 and Φ_2, we require previous two estimates which are Λ_0, Φ_0 and Λ_1, Φ_1. For Λ_{-1} and Φ_{-1}, we can use a random guess as:

$$\begin{aligned} \Phi_{-1} &= diag(C_{xx}) \\ Vr &= \mathrm{rand}(p, d) \\ V_{-1} &= Vr * \sqrt{\|C_{xx}\|_F \,/\, \|V\|_F} \end{aligned}$$

where $p = 9$, $d = 4$ and then for Λ_0, Φ_0 and Λ_1, Φ_1 we can do the log-EM by using (17) and (18) with Λ_{-1} and Φ_{-1}. With Λ_0, Φ_0 and Λ_1, Φ_1 we can apply the α-EM now. Figure 1 illustrates the different convergence curves of α-EM with different values of α. Remember that when $\alpha = -1$, it is log-EM.

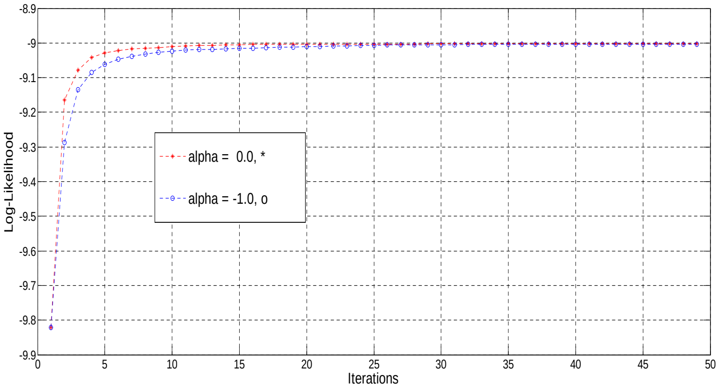

Figure 1: Convergence speed for various alpha

The log-likelihood on the y-axis is calculated by

$$LL = \log\det((\Phi + \Lambda\Lambda')^{-1}C_{xx}) - trace((\Phi + \Lambda\Lambda')^{-1}C_{xx}) \tag{31}$$

so for the optimal results we have $C_{xx} = \Phi + \Lambda\Lambda'$ then $LL = -p$ where p is the dimensionality of the problem. Whether you can reach the optimal results depends on the condition of the problem. Factor analysis can be only as good as the data allows.

Table 1 shows a speedup comparison. The second column shows that α-EM (the case of $\alpha = 0$) is 30/15=2.00 times faster than log-EM (the case of $\alpha = -1$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The α-EM algorithm didn't require more CPU time per iteration. So we can see that α-EM is much faster than log-EM by a total CPU-time speedup ratio of 30t/15t=2.00.

Table 1: Speedup Ratio For Factor Model Estimation(p=9, d=4)

α	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
-1.00	30	1t	30t	1.00
0	15	1t	15t	2.00

Besides the log-likelihood, we can compare the Hellinger distance for different values of α. The definition of the Hellinger distance is $H^2 = 1/2\int\left(\sqrt{P_0} - \sqrt{P_1}\right)^2 dx$ where P_0 and P_1 are probability density functions. In our case $P_0 \sim N(0, C_{xx})$ and $P_1 \sim N(0, \Phi + \Lambda\Lambda')$ then we have (see Appendix C for details) :

$$
\begin{aligned}
H^2 &= 1/2\int\left(\sqrt{P_0} - \sqrt{P_1}\right)^2 dx = 1 - \int\sqrt{P_0}\sqrt{P_1}dx \\
&= 1 - \det\left(\left(C_{xx}^{-1} + \left(\Phi + \Lambda\Lambda'\right)^{-1}\right)/2\right)^{-\frac{1}{2}} \det(C_{xx})^{-\frac{1}{4}} \det(\Phi + \Lambda\Lambda')^{-\frac{1}{4}} \\
&= 1 - \det\left(\left(C_{xx} + \left(\Phi + \Lambda\Lambda'\right)\right)/2\right)^{-\frac{1}{2}} \det(C_{xx})^{\frac{1}{4}} \det(\Phi + \Lambda\Lambda')^{\frac{1}{4}}
\end{aligned}
$$

Figure 2 illustrates the decreasing speed in the Hellinger distance as the number of iteration increases for various α.

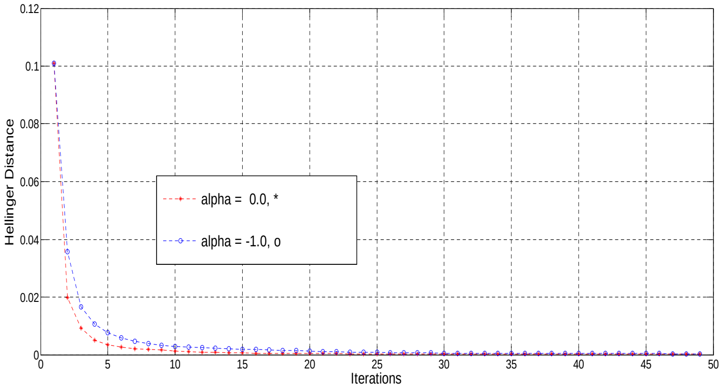

Figure 2: Hellinger distance for various alpha

From [94] we know that the Hellinger distance is actually bounded by the Information distance.

$$H\left(\Theta_0, \Theta_1\right) \leq \frac{1}{\sqrt{8}} I\left(\Theta_0, \Theta_1\right)$$

Especially for sufficiently small $I\left(\Theta_0, \Theta_1\right)$, we have:

$$\frac{K}{\sqrt{8}} I\left(\Theta_0, \Theta_1\right) \leq H\left(\Theta_0, \Theta_1\right)$$

where $0 < K < 1$. This shows that the upper bound of the Hellinger distance is the best possible. So if we can get the the sample covariance and factor model close in the Hellinger distance, it also implies that they are close in Information distance.

3.3.2 Factor analysis on financial data

Here, we download daily close prices of each member of S&P500 from Jan-03-2007 to May-31-2013 from Yahoo. Then we select members which have prices since Jan-03-2007 and got 471 members. We calculate the sample covariance C first, then with the sample covariance we can calculate the factor model using both log-EM and α-EM. In order to do the 1st iteration to obtain Λ_1 and Φ_1, it requires previous two states which are Λ_0, Φ_0 and Λ_{-1}, Φ_{-1}. For Λ_{-1} and Φ_{-1}, we use a random guess method with $d = 20$ and then we use log-EM for Λ_0 and Φ_0. after we got Λ_0, Φ_0 and Λ_{-1}, Φ_{-1} we can apply α-EM. Figure 3 illustrates the different convergence curves of α-EM for different values of α. Remember that when $\alpha = -1$, this is log-EM.

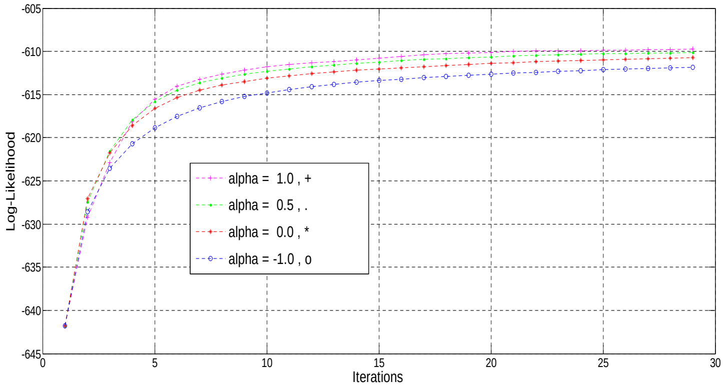

Figure 3: Convergence speed for various alpha

Table 2 also shows a speedup comparison. The second column shows that the α-EM (the case of $\alpha = 1$) is 30/10=3.00 times faster than the log-EM (the case of $\alpha = -1$). The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. Again, the α-EM takes the same CPU time as the log-EM per iteration. We can see that the α-EM is still faster than the log-EM by a CPU-time ratio of 30t/10t=3.00.

Table 2: Speedup Ratio For Factor Model Estimation(p=471, d=20)

α	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
-1.00	30	1t	30t	1.00
0	16	1t	16t	1.875
+0.5	12	1t	12t	2.50
+1.00	10	1t	10t	3.00

We can see that during the first few iterations $\alpha = 1$ is not the fastest. It was mentioned in Section 3.2.1 that $\beta = \alpha + 2$ is not always a good approximation. We should let α increases to 1 as the iterations increase. For example we can choose

$\alpha = 0, 0.25, 0.5$ for the first three iterations and $\alpha = 1$ from the fourth iteration. Figure 4 illustrates the different convergence curves of α-EM for different values of α.

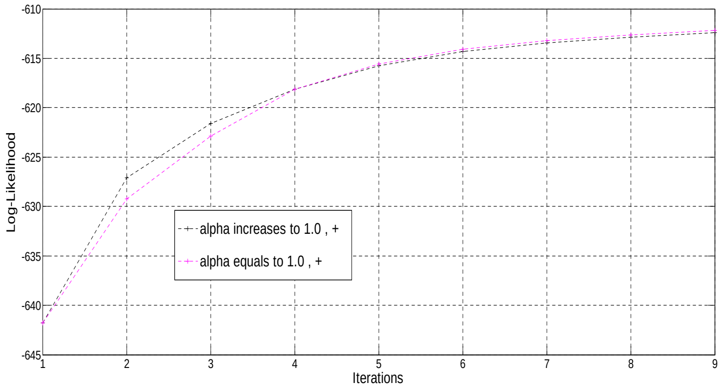

Figure 4: Convergence speed for various alpha

In addition, it was mentioned in Section 3.2.2 that $r = 1$ is not always a good approximation. For $r \neq 1$ in series expansion, we can choose $r = 1 - 0.1\hat{\ }k$ at the kth iteration. It is more appropriate than using $r = 1$ at all iterations. So as iterations increase the value of r gets closer to 1. Figure 5 illustrates the different convergence curves of α-EM with the same α but different values of r in series expansion.

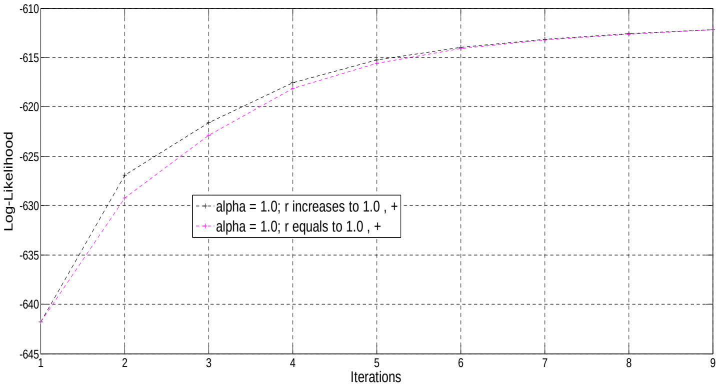

Figure 5: Convergence speed for various r

We can see that they are almost the same, but if you amplify the first few iterations we notice that $r = 1 - 0.1\hat{}k$ is an improvement compared to $r = 1$. The improvement is not large.

Here we consider a smaller p. We just pick the first 100 members out of 471 members of S&P500. For the number of factors we choose $d = 10$, all other parts remain the same. Figure 6 also illustrates the different convergence curves of the α-EM in previous section with different α values. Remember that when $\alpha = -1$, it is the log-EM.

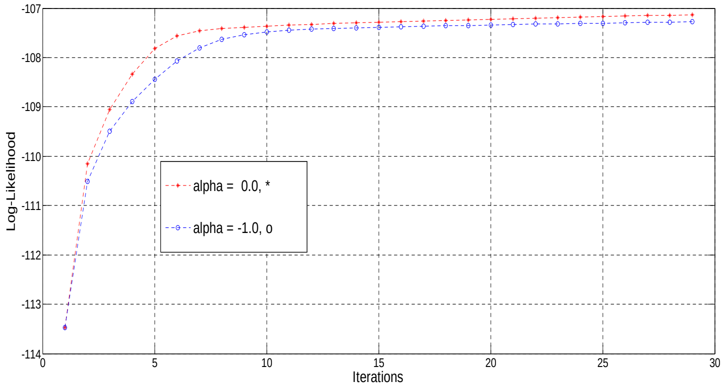

Figure 6: Convergence speed for various alpha

Table 3 also shows a speedup comparison. The second column shows that the α-EM (the case of $\alpha = 0$) is 30/16=1.875 times faster than the log-EM (the case of $\alpha = -1$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. CPU time per iteration is the same for α-EM and log-EM. So we can see that α-EM is much faster than log-EM by a total CPU-time ratio of 30t/16t=1.875.

Table 3: Speedup Ratio For Factor Model Estimation(p=100, d=10)

α	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
-1.00	30	1t	30t	1.00
0	16	1t	16t	1.875

Therefore, for small dimension problems or large dimension problems the α-EM algorithm will not require more CPU time per iteration than the log-EM algorithm. So as long as the number iterations of the α-EM algorithm is smaller than the log-EM algorithm for the same accuracy, the α-EM algorithm will save us the total computation time. That's we should choose the α-EM algorithm over the log-EM algorithm.

3.4 Concluding Remarks

In this section, we applied the α-EM algorithm to factor model estimation. Through calculation we found it is hard to get causal update equations directly. Instead we get two non-causal update equations. From those non-causal update equations we learned that α-EM has faster convergence speed than log-EM but each major iteration for α-EM takes large amounts of time. We can't use the α-EM algorithm in practice without solving the non-causality. This is why we did causal approximation and series expansion in order to get the approximate causal update equations. By choosing the proper values of α, we showed that the α-EM algorithm converges much faster than the log-EM algorithm in factor model estimation and also gives more accurate estimates. In CPU-time, as long as we save some results from the previous updates for the next updates, the α-EM algorithm doesn't require more CPU time than the log-EM algorithm. However more importantly, the speedup in convergence is significant, so the α-EM can save us the total computation time for the same accuracy.

In order to make the α-EM algorithm work in practice, causal approximation and series expansion played very important roles. In causal approximation, there are actually many choices about how α increases to 1. Which choice is better usually related to what problem you have. In series expansion, there are actually many other choices of a, such as $a = 1 - 0.1\hat{}(0.9 + k/10)$ for kth iteration, which works better than $a = 1 - 0.1\hat{}k$ in the first few iterations. But the improvement is not significant in factor model estimation. Also, there must be other methods to solve the non-causality, such as moving all the future states on one side of the original update equations (13) and (14). That would be the most accurate method but it is also harder than the approximation method. Thus, further exploration of practical issues pertaining to the α-EM family is needed.

For the α-EM algorithm we used, we focus our attention on the convex divergence (1) because of its general capacity on convex optimization. We would like to consider other types of surrogate functions.

4 The α-EM Algorithm for Mixture of Factor Models

We apply the α-EM algorithm to mixture of factor Models. This method utilizes the α-logarithm as a surrogate function for the traditional logarithm to process the likelihood ratio. Since existing or traditional mixture of factor models are the outcome of log-EM, it had been excepted that α-EM for mixture of factor models would exist. In this section, we show that this foresight is true by using methods of the iteration index shift and likelihood ratio expansion. The new method is theoretically based on the α-EM algorithm, all of its properties are inherited. Empirical results from artificial data show that the α-EM algorithm can save us in total computation time.

4.1 Non-Causal Update Equations

For mixture of factor models, we have :

$$P_c = \prod_{i=1}^{N}\prod_{j=1}^{K}\{\pi_j P_c(x_i, z_i|\Lambda_j, \Phi_j)\}^{w_j}$$

so the surrogate function $Q^{(\alpha)}_{X,Z,W|X}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0)$ is :

$$\begin{aligned}
Q^{(\alpha)}_{X,Z,W|X}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0) &= E_{P(Z,W|X,\Phi_0,\Lambda_0)}[L^{(\alpha)}_{X,Z,W}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0)] \\
&= E_{P(Z,W|X,\Phi_0,\Lambda_0)}\left[\frac{2}{1+\alpha}\left(\frac{P_C(X,Z,W|\Phi_1,\Lambda_1)}{P_C(X,Z,W|\Phi_0,\Lambda_0)}\right)^{\frac{1+\alpha}{2}} - 1\right] \\
&= \frac{2}{1+\alpha}\left(\prod_{i=1}^{N} E_{P(z_i|x_i,\Lambda_0,\Phi_0)}\left[\prod_{j=1}^{K}\left\{\left(\frac{\pi_{1j}Pc(x_i, z_i|\Lambda_{1j}, \Phi_{1j})}{\pi_{0j}Pc(x_i, z_i|\Lambda_{0j}, \Phi_{0j})}\right)^{\frac{1+\alpha}{2}}\right\}^{w_j}\right] - 1\right) \\
&= \frac{2}{1+\alpha}\left(S^{(\alpha)}_{Z|X,\Phi_0,\Lambda_0} - 1\right) \qquad (32)
\end{aligned}$$

where

$$\begin{aligned}
S^{(\alpha)}_{Z,W|X,\Phi_0,\Lambda_0} &= \prod_{i=1}^{N} W_i^{(\alpha)} \\
W_i^{(\alpha)} &= \sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right] \\
P_i &= \frac{Pc(x_i, z_i|\Lambda_1, \Phi_1)}{Pc(x_i, z_i|\Lambda_0, \Phi_0)} \\
E[\cdot] &= E_{P(z|x,w,\Lambda_0,\Phi_0)}[\cdot]
\end{aligned}$$

and

$$\begin{aligned}
\pi_{0j} &= P(w_j) = \int P(w_j|x_i)P(x_i)dx \\
h_{ij} &= E[w_j|x_i] = P(w_j|x_i) = \frac{P(w_j, x_i)}{P(x_i)} \\
&= \frac{\pi_{0j} N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_{0j})}{\sum\limits_{j=1}^{K} \pi_{0j} N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_{0j})}
\end{aligned}$$

After the E-step we need to do the M-step. The update equations for Φ_j and Λ_j can be obtained by differentiating $Q^{(\alpha)}_{X,Z,W|X}(\Phi_1, \Lambda_1|\Phi_0, \Lambda_0)$ with respect to the update parameters Φ_{1j} and Λ_{1j} and setting differentiation to zero solve for maximization. For Λ_{1j} we have:

$$\begin{aligned}
\frac{\partial Q^{(\alpha)}}{\partial \Lambda_{1j}} &= 0 \Rightarrow \frac{\partial S^{(\alpha)}}{\partial \Lambda_{1j}} = 0 \\
\frac{\partial S^{(\alpha)}}{\partial \Lambda_{1j}} &= \sum_{i=1}^{N} \frac{\partial W_i^{(\alpha)}}{\partial \Lambda_{1j}} \prod_{t=1,t\neq i}^{N} W_i^{(\alpha)} = \sum_{i=1}^{N} \frac{\partial W_i^{(\alpha)}}{\partial \Lambda_{1j}} \frac{S^{(\alpha)}}{W_i^{(\alpha)}} = \sum_{i=1}^{N} \frac{\frac{\partial W_i^{(\alpha)}}{\partial \Lambda_{1j}}}{W_i^{(\alpha)}} S^{(\alpha)}
\end{aligned}$$

$$S^{(\alpha)} \neq 0 \Rightarrow \sum_{i=1}^{N} \frac{\frac{\partial W_i^{(\alpha)}}{\partial \Lambda_{1j}}}{W_i^{(\alpha)}} = 0 \Rightarrow \sum_{i=1}^{N} \frac{\frac{\partial \sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]}{\partial \Lambda 1j}}{\sum\limits_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]} = \sum_{i=1}^{N} \frac{h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Lambda_{1j}}\right]}{\sum\limits_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]} = 0 \tag{33}$$

and for Φ_{1j} we have

$$\frac{\partial Q^{(\alpha)}}{\partial \Phi_{1j}^{-1}} = 0 \Rightarrow \frac{\partial S^{(\alpha)}}{\partial \Phi_{1j}^{-1}} = 0 \Rightarrow \sum_{i=1}^{N} \frac{\frac{\partial \sum\limits_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]}{\partial \Phi_{1j}^{-1}}}{\sum\limits_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]} = \sum_{i=1}^{N} \frac{h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Phi_{1j}^{-1}}\right]}{\sum\limits_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]} = 0 \tag{34}$$

In order to solve equations (33) and (34) we need to calculate $E\left[P^{\frac{1+\alpha}{2}}\right]$, $E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Lambda_{1j}}\right]$ and $E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Phi_{1j}^{-1}}\right]$. By the definition of expectation:

$$E\left[P^{\frac{1+\alpha}{2}}\right] = \int P(z_i|x_{i,}\Lambda_0, \Phi_0) \left(\frac{Pc(x_i, z_i|\Lambda_1, \Phi_1)}{Pc(x_i, z_i|\Lambda_0, \Phi_0)}\right)^{\frac{1+\alpha}{2}} dz_i \tag{35}$$

$$E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Lambda_{1j}}\right] = \frac{1+\alpha}{2} E\left[P^{\frac{1+\alpha}{2}} * \left(\Phi_{1j}^{-1} x_i z_i' - \Phi_{1j}^{-1} \Lambda_{1j} z_i z_i'\right)\right] \tag{36}$$

$$E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Phi_{1j}^{-1}}\right] = \frac{1+\alpha}{2} E\left[P^{\frac{1+\alpha}{2}} * \frac{1}{2}\left(\Phi_{1j} - x_i x_i' + x_i z_i' \Lambda_{1j}' + \Lambda_{1j} z_i x_i' - \Lambda_{1j} z_i z_i' \Lambda_{1j}'\right)\right] \tag{37}$$

The update equations for π_j can be obtained by differentiating the following equation w.r.t. π_{1j} and setting the derivative to zero.

$$L = Q^{(\alpha)} + \lambda \left(\sum_{j=1}^{K} \pi_{1j} - 1\right)$$

We get

$$\frac{\partial L}{\partial \pi_{1j}} = \frac{\partial Q^{(\alpha)}}{\partial \pi_{1j}} + \lambda = 0 \tag{38}$$

$$\frac{\partial L}{\partial \lambda} = \sum_{j=1}^{K} \pi_{1j} - 1 = 0 \tag{39}$$

After computing those expectations (35), (36), (37) and solving equations (38), (39), we have the update equations (see Appendix D for details) :

$$\begin{aligned}
\pi_{1j} &= \frac{1}{N}\sum_{i=1}^{N} h'_{ij} && (40)\\
\Lambda_{1j} &= \sum_{j=1}^{N} h'_{ij} x_i E[z'_i] \left(\sum_{j=1}^{N} h'_{ij} E[z_i z'_i]\right)^{-1} && (41)\\
\Phi_{1j} &= \left(\sum_{i=1}^{N} h'_{ij} x_i x'_i - \sum_{i=1}^{N} h'_{ij} x_i E[z'_i]\Lambda'_1\right)\left(\sum_{i=1}^{N} h'_{ij}\right)^{-1} && (42)
\end{aligned}$$

and the expectation here is w.r.t. a new distribution:

$$\begin{aligned}
E[z_i] &= \Sigma_j W'_j x_i \Rightarrow E[z'_i] = x'_i W_j \Sigma_j\\
E[z_i z'_i] &= Var[z_i] + E[z_i]E[z_i]' = \Sigma_j + \Sigma_j W'_j x_i x'_i W_j \Sigma_j\\
\Sigma_j^{-1} &= \frac{1+\alpha}{2}\Lambda'_{1j}\Phi_{1j}^{-1}\Lambda_{1j} - \frac{1+\alpha}{2}\Lambda'_{0j}\Phi_{0j}^{-1}\Lambda_{0j} + C_j^{-1}\\
W_j &= \frac{1+\alpha}{2}\Phi_{1j}^{-1}\Lambda_{1j} - \frac{1+\alpha}{2}\Phi_{0j}^{-1}\Lambda_{0j} + \beta'_j C_j^{-1}\\
h'_{ij} &= \frac{h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j}{\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j}\\
b_j &= \left(\frac{|\Phi_{1j}|^{-\frac{1}{2}}}{|\Phi_{0j}|^{-\frac{1}{2}}}\right)^{\frac{1+\alpha}{2}} |C_j|^{-\frac{1}{2}}\,|\Sigma_j|^{\frac{1}{2}} e^{-\frac{1}{2}x'_i\beta'_j C_j^{-1}\beta_j x_i} e^{\frac{1}{2}x'_i W_j \Sigma_j W'_j x_i} e^{-\frac{1}{2}\frac{1+\alpha}{2}x'_i\left(\Phi_{1j}^{-1}-\Phi_{0j}^{-1}\right)x_i}
\end{aligned}$$

We notice that in the update equation for π_{1j}, we have π_{1j} on both sides. We have the same problem for Λ_{1j} and Φ_{1j}. It is hard to put either π_{1j}, Λ_{1j} and Φ_{1j} on one side of the equations. This is what we mean by non-causal. So we can't use these update equations directly.

For general case $-1 < \alpha < 1$, let's consider the update equations in a general way:

$$\begin{aligned}
\pi_{1j} &= P(\Lambda_{1j}, \Phi_{1j}, \Phi_{0j}, \Lambda_{0j}, \pi_{0j}) && (43)\\
\Lambda_{1j} &= F(\Lambda_{1j}, \Phi_{1j}, \Phi_{0j}, \Lambda_{0j}, \pi_{0j}) && (44)\\
\Phi_{1j} &= G(\Lambda_{1j}, \Phi_{1j}, \Phi_{0j}, \Lambda_{0j}, \pi_{0j}) && (45)
\end{aligned}$$

They are non-causal but they illustrate two important things. First, we can iteratively update π_{1j}, Λ_{1j} and Φ_{1j} through (43), (44) and (45) until π_{1j}, Λ_1 and Φ_1

converge. We call this one major iteration which is the iterations we count in the log-EM. Then replace π_{1j}, Φ_{0j} and Λ_{0j} with π_{1j}, Λ_1 and Φ_1 do the same thing for the next major iteration π_{2j}, Λ_{2j} and Φ_{2j}. In practice, the convergence speed is much faster log-EM for the same major iteration. Second, each major iteration here contains many minor iterations which will take large amounts of time. Using this updating method the α-EM can't save us in total computation time in practice. Therefore, on one hand we know that α-EM is better than log-EM in convergence speed, on the other hand we need effective update equations otherwise we can't use the α-EM algorithm in practice.

Let's consider two special cases first:

Case 1. $\alpha = -1$:

$$\begin{aligned} \Sigma_j^{-1} &= C_j^{-1} \\ W_j &= \beta_j' C_j^{-1} \end{aligned}$$

so $b_j = 1$ and $h'_{ij} = h_{ij}$ then the update equations are:

$$\pi_{1j} = \frac{1}{N}\sum_{i=1}^{N} h_{ij} = p(\pi_{0j}, \Lambda_{0j}, \Phi_{0j}) \quad (46)$$

$$\Lambda_{1j} = \sum_{j=1}^{N} h_{ij} x_i E[z_i'] \left(\sum_{j=1}^{N} h_{ij} E[z_i z_i'] \right)^{-1} = f(\pi_{0j}, \Lambda_{0j}, \Phi_{0j}) \quad (47)$$

$$\Phi_{1j} = \left(\sum_{i=1}^{N} h_{ij} x_i x_i' - \sum_{i=1}^{N} h_{ij} x_i E[z_i'] \Lambda_1' \right) \left(\sum_{i=1}^{N} h_{ij} \right)^{-1} = g(\pi_{0j}, \Lambda_{1j}, \Lambda_{0j}, \Phi_{0j}) \quad (48)$$

these update equations are the same as Ghahramani and Hinton [51]. We also proved that the log-EM algorithm is a special case of the α-EM algorithm.

Case 2. $\alpha = 1$:

$$h'_{ij} = \frac{\pi_{1j} N(x_i, \Lambda_{0j}\Lambda_{0j}' + \Phi_{0j}) b_j}{\sum_{j=1}^{K} \pi_{1j} N(x_i, \Lambda_{0j}\Lambda_{0j}' + \Phi_{0j}) b_j}$$

According to Sylvester's determinant theorem we have

$$\left|I_d + \Lambda'_{0j}\Phi_0^{-1}\Lambda_{0j}\right| = \left|(\Phi_{0j}^{-1}\Lambda_{0j}\Lambda'_{0j} + I_p)\right|$$

then

$$h'_{ij} = \frac{\pi_{1j}N(x_i, \Lambda_{1j}\Lambda'_{1j} + \Phi_{1j})}{\sum\limits_{j=1}^{K} \pi_{1j}N(x_i, \Lambda_{1j}\Lambda'_{1j} + \Phi_{1j})}$$

so the update equations are:

$$\begin{aligned}
\pi_{1j} &= \frac{1}{N}\sum_{i=1}^{N} h'_{ij} = p(\pi_{1j}, \Lambda_{1j}, \Phi_{1j}) && (49)\\
\Lambda_{1j} &= \sum_{j=1}^{N} h'_{ij}x_iE[z'_i]\left(\sum_{j=1}^{N} h'_{ij}E[z_iz'_i]\right)^{-1} = f(\pi_{1j}, \Lambda_{1j}, \Phi_{1j}) && (50)\\
\Phi_{1j} &= \left(\sum_{i=1}^{N} h_{ij}x_ix'_i - \sum_{i=1}^{N} h_{ij}x_iE[z'_i]\Lambda'_1\right)\left(\sum_{i=1}^{N} h_{ij}\right)^{-1} = g(\pi_{1j}, \Lambda_{1j}, \Phi_{1j}) && (51)
\end{aligned}$$

Since we can't solve these update equations, we can iteratively update (49), (50) and (51) until π_{1j}, Λ_{1j} and Φ_{1j} don't change. In practice this method takes exactly the same computation time as when $\alpha = -1$ because they have identical p, f and g. In order to have a practical solution we need to solve the non-causality.

4.2 Causal update equations

As in the non-mixture case, the reason for non-causality is the expectations, equation (35), (36), (37) and the update equation for π_{1j}. We need to calculate the three expectations $E\left[P^{\frac{1+\alpha}{2}}\right]$, $E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Lambda_{1j}}\right]$ and $E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Phi_{1j}^{-1}}\right]$ in a causal way and also π_{1j}. These three expectations have the form:

$$\int P(z_i|x_{i,}\Lambda_{0j}, \Phi_{0j})\left(\frac{Pc(x_i, z_i|\Lambda_{1j}, \Phi_{1j})}{Pc(x_i, z_i|\Lambda_{0j}, \Phi_{0j})}\right)^{\frac{1+\alpha}{2}} dz_i \qquad (52)$$

in common and we need to calculate (52) without using Λ_{1j} and Φ_{1j}. We also need to calculate π_{1j} without using π_{1j}.

4.2.1 Causal approximation

We have

$$\begin{aligned} P(z_i|x_{i,}\Lambda_{0j},\Phi_{0j})\left(\frac{Pc(x_i,z_i|\Lambda_{1j},\Phi_{1j})}{Pc(x_i,z_i|\Lambda_{0j},\Phi_{0j})}\right)^{\frac{1+\alpha}{2}} &\approx P(z_i|x_i,\Lambda_{1j},\Phi_{1j})\left(\frac{Pc(x_i,z_i|\Lambda_{1j},\Phi_{1j})}{Pc(x_i,z_i|\Lambda_{0j},\Phi_{0j})}\right)^{-\frac{1-\alpha}{2}} \\ &\approx P(z_i|x_i,\Lambda_{0j},\Phi_{0j})\left(\frac{Pc(x_i,z_i|\Lambda_{0j},\Phi_{0j})}{Pc(x_i,z_i|\Lambda_{-1j},\Phi_{-1j})}\right)^{-\frac{1-\beta}{2}} \quad (53) \end{aligned}$$

around the region of $P(x_i|\Lambda_{1j},\Phi_{1j}) = P(x_i|\Lambda_{0j},\Phi_{0j}) + o(1)$. The last term is the causal approximation w.r.t. the iteration index shift or shift of time [82], [83]. We can use the relationship:

$$\frac{1+\alpha}{2} = -\frac{1-\beta}{2} \Rightarrow \beta = \alpha + 2 \quad (54)$$

and because we have $\alpha \in (-1, 1)$, then we also have $\beta \in (1, 3)$.

Within the first a few iterations $\beta = \alpha + 2$ is not always a good approximation. For example, when $\alpha = 1$, $P_1/P_0 \approx P_0/P_{-1}$ is not a good approximation in the first a few iterations. This bad approximation will cause us numerical problems in practice. Another example is when $\alpha = 0$, $(P_1/P_0)^{1/2} \approx (P_0/P_{-1})^{1/2}$ is a better approximation during the first a few iterations since P_1/P_0 and P_0/P_{-1} are greater than 1. So for choices of α close to or equal to 1, we can choose β starting from 2. So that it gets close to $\alpha+2$ as the iteration increases. We will see this is sometimes necessary in practice.

For π_{1j}, we do the same thing:

$$\begin{aligned} \pi_{0j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} &= \pi_{1j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}-1} \\ &\approx \pi_{0j}\left(\frac{\pi_{0j}}{\pi_{-1j}}\right)^{-\frac{1-\beta}{2}} \quad (55) \end{aligned}$$

Now, we can approximately calculate (52) without knowing $\pi_{1j}, \Lambda_{1j}, \Phi_{1j}$. This requires a power computation of a likelihood ratio. This is computational expensive and becomes intractable as time increases. However another approximation is necessary in view of computational complexity.

4.2.2 Series Expansion

A Taylor expansion can simplify this without discarding merit of the α-log likelihood ratio.

$$P(z_i|x_i,\Lambda_{0j},\Phi_{0j})\left(\frac{Pc(x_i,z_i|\Lambda_{0j},\Phi_{0j})}{Pc(x_i,z_i|\Lambda_{-1j},\Phi_{-1j})}\right)^{-\frac{1-\beta}{2}} = P(z_i|x_i,\Lambda_{-1j},\Phi_{-1j})\left(\frac{Pc(x_i,z_i|\Lambda_{0j},\Phi_{0j})}{Pc(x_i,z_i|\Lambda_{-1j},\Phi_{-1j})}\right)^{\frac{1+\beta}{2}} \tag{56}$$

Let's assume that $f(x) = x^{\frac{1+\beta}{2}}$, according to the Taylor expansion we have $f(x) = f(r) + \frac{f'(r)}{1!}(x-r) + o(1)$. For our case $x = \frac{Pc(x_i,z_i|\Lambda_0,\Phi_0)}{Pc(x_i,z_i|\Lambda_{-1},\Phi_{-1})}$. Assume $r = 1$, so we get:

$$\left(\frac{Pc(x_i,z_i|\Lambda_{0j},\Phi_{0j})}{Pc(x_i,z_i|\Lambda_{-1j},\Phi_{-1j})}\right)^{\frac{1+\beta}{2}} \approx \frac{1-\beta}{2} + \frac{1+\beta}{2}\frac{Pc(x_i,z_i|\Lambda_{0j},\Phi_{0j})}{Pc(x_i,z_i|\Lambda_{-1j},\Phi_{-1j})} \tag{57}$$

now we substitute the right hand side of equation (53) with equation (56) and (57) and we get:

$$P(z_i|x_{i,}\Lambda_{0j},\Phi_{0j})\left(\frac{Pc(x_i,z_i|\Lambda_{1j},\Phi_{1j})}{Pc(x_i,z_i|\Lambda_{0j},\Phi_{0j})}\right)^{\frac{1+\alpha}{2}} \approx \frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1j},\Phi_{-1j}) + \frac{1+\beta}{2}P(z_i|x_i,\Lambda_{0j},\Phi_{0j})$$

Again, we do the same thing for π_{1j}, we get:

$$\begin{aligned}\pi_{0j}\left(\frac{\pi_{0j}}{\pi_{-1j}}\right)^{-\frac{1-\beta}{2}} &= \pi_{-1j}\left(\frac{\pi_{0j}}{\pi_{-1j}}\right)^{\frac{1+\beta}{2}} \\ &\approx \frac{1-\beta}{2}\pi_{-1j} + \frac{1+\beta}{2}\pi_{0j} \end{aligned} \tag{58}$$

Now we substitute the right hand side of equation (55) with equation (58) and we get:

$$\pi_{0j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} \approx \frac{1-\beta}{2}\pi_{-1j} + \frac{1+\beta}{2}\pi_{0j}$$

for h'_{ij} we also get:

$$h'_{ij} \approx \frac{N_{0j}\left(\frac{1-\beta}{2}\pi_{-1j} + \frac{1+\beta}{2}\pi_{0j}\right)}{\sum\limits_{j=1}^{K} N_{0j}\left(\frac{1-\beta}{2}\pi_{-1j} + \frac{1+\beta}{2}\pi_{0j}\right)}$$

So, now we can calculate the expectations in a causal way without using Λ_{1j} and Φ_{1j}. Also, we can update π_{1j} in a causal way. The causal update equations are:

$$\pi_{1j} = \frac{1}{N}\sum_{i=1}^{N} h'_{ij} = \frac{1}{N}\sum_{i=1}^{N} \frac{N_{0j}\left(\frac{1-\beta}{2}\pi_{-1j} + \frac{1+\beta}{2}\pi_{0j}\right)}{\sum\limits_{j=1}^{K} N_{0j}\left(\frac{1-\beta}{2}\pi_{-1j} + \frac{1+\beta}{2}\pi_{0j}\right)} \tag{59}$$

$$\Lambda_{1j} = \frac{\frac{1-\beta}{2}\sum\limits_{i=1}^{N} h'_{ij}x_iE_{-1}[z'_i] + \frac{1+\beta}{2}\sum\limits_{i=1}^{N} h'_{ij}x_iE_0[z'_i]}{\frac{1-\beta}{2}\sum\limits_{i=1}^{N} h'_{ij}E_{-1}[z_iz'_i] + \frac{1+\beta}{2}\sum\limits_{i=1}^{N} h'_{ij}E_0[z_iz'_i]} \tag{60}$$

$$\Phi_{1j} = \frac{1}{\sum\limits_{i=1}^{N} h'_{ij}} diag\left(\sum_{i=1}^{N} h'_{ij}x_ix'_i - \left(\frac{1-\beta}{2}\sum_{i=1}^{N} h'_{ij}x_iE_{-1}[z'_i] + \frac{1+\beta}{2}\sum_{i=1}^{N} h'_{ij}x_iE_0[z'_i]\right)\Lambda'_{1j}\right) \tag{61}$$

where

$$\begin{aligned} E_{-1}[z'_i] &= \beta_{-1j}x_i \text{ and } E_{-1}[z_iz'_i] = C^{-1}_{-1j} + \beta_{-1j}x_ix'_i\beta'_{-1j} \\ E_0[z_i] &= \beta_{0j}x_i \text{ and } E_{0j}[z_iz'_i] = C^{-1}_{0j} + \beta_{0j}x_ix'_i\beta'_{0j} \end{aligned}$$

with

$$\begin{aligned} \beta_{-1j} &= \Lambda'_{-1j}(\Phi_{-1j} + \Lambda_{-1j}\Lambda'_{-1j})^{-1} \\ C_{-1j} &= I - \Lambda'_{-1j}(\Phi_{-1j} + \Lambda_{-1j}\Lambda'_{-1j})^{-1}\Lambda_{-1j} \\ \beta_{0j} &= \Lambda'_{0j}(\Phi_{0j} + \Lambda_{0j}\Lambda'_{0j})^{-1} \\ C_{0j} &= I - \Lambda'_{0j}(\Phi_{0j} + \Lambda_{0j}\Lambda'_{0j})^{-1}\Lambda_{0j} \end{aligned}$$

Now we get three causal update equations and we are able to estimate the mixture of factor models. Therefore, at the kth iteration to obtain π_{kj}, Λ_{kj} and Φ_{kj}, we use $\pi_{k-1j}, \Lambda_{k-1j}, \Phi_{k-1j}$ and $\pi_{k-2j}, \Lambda_{k-2j}, \Phi_{k-2j}$ on the right-hand sides of (59), (60) and (61). Compare with the non-causal or log-EM update equations we need to calculate the $E[z]$ and $E[zz']$ of the previous state and state prior to that. It seems that the α-EM algorithm needs to do more calculations in each update but in practice we can save $E[z]$ and $E[zz']$ of the previous state for the next iteration, so we don't need to recalculate it. For example, for Λ_{1j} and Φ_{1j} we need to calculate $E_{-1j}[z_i']$ and $E_{0j}[z_i']$, when we calculate Λ_{2j} and Φ_{2j} we can reuse $E_{0j}[z_i']$ and we only need to calculate $E_{1j}[z_i']$. We only compute $E[z]$ and $E[zz']$ once for each state, this is exactly the same as the log-EM algorithm. Thus, the total computation time per iteration of the α-EM algorithm and the log-EM algorithm are almost the same. We will see the numerical results in the following section.

4.3 Comparison of single and multiple Φ

If we assume that we have the same idiosyncratic risk then the derivative equation for Φ is:

$$\frac{\partial Q^{\alpha}}{\partial \Phi_1^{-1}} = 0 \Rightarrow \frac{\partial S^{(\alpha)}}{\partial \Phi_1^{-1}} = 0 \Rightarrow \sum_{j=1}^{N} \frac{\frac{\partial \sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]}{\partial \Phi_1^{-1}}}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]} = \sum_{j=1}^{N} \frac{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[\frac{\partial P^{\frac{1+\alpha}{2}}}{\partial \Phi_1^{-1}}\right]}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right]} = 0$$

and the derivative of the expectation is:

$$\frac{\partial E\left[P^{\frac{1+\alpha}{2}}\right]}{\partial \Phi_1^{-1}} = \frac{1+\alpha}{2} E\left[P^{\frac{1+\alpha}{2}} * \left(\frac{1}{2}\Phi_1 - \frac{1}{2}x_i x_i' + \frac{1}{2}x_i z_i' \Lambda_{1j}' + \frac{1}{2}\Lambda_{1j} z_i x_i' - \frac{1}{2}\Lambda_{1j} z_i z_i' \Lambda_{1j}'\right)\right]$$

The non-causal update equation for Φ is:

$$\Phi_1 = \frac{1}{N} diag\left(\sum_{i=1}^{N} x_i x_i' - \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E[z_i'] \Lambda_{1j}'\right)$$

Again, let's consider two special cases:

Case 1. $\alpha = -1$:

$$\begin{aligned}
\pi_{j,1} &= \frac{1}{N}\sum_{i=1}^{N} h_{ij} \\
\Lambda_{1j} &= \sum_{i=1}^{N} h_{ij}' x_i x_i' \delta_{0j} \left(\sum_{i=1}^{N} h_{ij}' \left(\Delta_{0j} + \delta_{0j}' x_i x_i' \delta_{0j}\right)\right)^{-1} \\
\Phi_1 &= diag(C_{xx-} \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i x_i' \delta_{0j} \Lambda_{1j}')
\end{aligned}$$

where

$$h_{ij} = \frac{\pi_{0j} N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_0)}{\sum\limits_{j=1}^{K} \pi_{0j} N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_0)}$$

These update equations are the same as Ghahramani and Hinton [51]. We also proved that the log-EM algorithm is a special case of the α-EM algorithm.

Case 2. $\alpha = 1$:

$$\begin{aligned}
\pi_{j,1} &= \frac{1}{N}\sum_{i=1}^{N} h'_{ij} \\
\Lambda_{1j} &= \sum_{i=1}^{N} h'_{ij} x_i x'_i \delta_{1j} \left(\sum_{i=1}^{N} h'_{ij} \left(\Delta_{1j} + \delta'_{1j} x_i x'_i \delta_{1j}\right)\right)^{-1} \\
\Phi_1 &= diag(C_{xx-} \sum_{i=1}^{N}\sum_{j=1}^{K} h'_{ij} x_i x'_i \delta_{1j} \Lambda'_{1j})
\end{aligned}$$

where

$$h'_{ij} = \frac{\pi_{1j} N(x_i, \Lambda_{1j}\Lambda'_{1j} + \Phi_1)}{\sum\limits_{j=1}^{K} \pi_{1j} N(x_i, \Lambda_{1j}\Lambda'_{1j} + \Phi_1)}$$

Again, in practice $\alpha = -1$ and $\alpha = 1$ are the same, we need causal update equations.

After causal approximation and series expansion, the causal update equations for Φ_1 is:

$$\Phi_1 = \frac{1}{N} diag\left(\sum_{i=1}^{N} x_i x'_i - \left(\frac{1-\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{K} h'_{ij} x_i E_{-1}[z'_i] + \frac{1+\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{K} h'_{ij} x_i E_0[z'_i]\right)\Lambda'_{1j}\right)$$

4.4 Empirical Results

4.4.1 Mixture of factor models on artificial data (k = 2)

Here, first we create two factor structured matrices $D + B_1 * B'_1$ and $D + B_2 * B'_2$ with random entries. D is $p \times p$ and B_1, B_2 are $p \times d$. The elements of D are exponentially distributed idiosyncratic variances and B_1, B_2 have Gaussian distributed factor loads. We choose $p = 100$ and $d = 20$ then generate 5000 samples with each covariance $D + B_1 * B_1'$ and $D + B_2 * B_2'$. After that we get our 10000 samples. We calculate sample covariance C first, then with the sample covariance we can estimate mixture of factor analysers by both the log-EM algorithm and the α-EM algorithm.

In order to do the 1st iteration to obtain $\pi_{21}, \pi_{22}, \Lambda_{21}, \Lambda_{22}$ and Φ_2, we require the previous two estimates which are $\pi_{01}, \pi_{02}, \Lambda_{01}, \Lambda_{02}, \Phi_0$ and $\pi_{11}, \pi_{12}, \Lambda_{11}, \Lambda_{12}, \Phi_1$. For $\Lambda_{01}, \Lambda_{02}$ and Φ_0, we can use a random guess:

$$\begin{aligned}
\Phi_0 &= diag(C_{xx}) \\
\Lambda r_1 &= \text{rand}(p, d) \text{ and } \pi_{r1} = \text{rand}(1, 2) \\
\Lambda r_2 &= \text{rand}(p, d) \text{ and } \pi_{r2} = \text{rand}(1, 2) \\
\Lambda_{01} &= \Lambda r_1 * \sqrt{\|C_{xx}\|_F \,/\, \|\Lambda r_1\|_F} \\
\Lambda_{02} &= \Lambda r_2 * \sqrt{\|C_{xx}\|_F \,/\, \|\Lambda r_2\|_F} \\
\pi_{01} &= \pi_{r1}/sum(\pi_{r1}) \\
\pi_{02} &= \pi_{r2}/sum(\pi_{r2})
\end{aligned}$$

where $p = 100$, $d = 20$. Then for $\pi_{11}, \pi_{12}, \Lambda_{11}, \Lambda_{12}$ and Φ_1 we can do the log-EM by using (46), (47) and (48) with $\pi_{01}, \pi_{02}, \Lambda_{01}, \Lambda_{02}$ and Φ_0. With $\pi_{01}, \pi_{02}, \Lambda_{01}, \Lambda_{02}, \Phi_0$ and $\pi_{11}, \pi_{12}, \Lambda_{11}, \Lambda_{12}, \Phi_1$ we can apply the α-EM algorithm. Figure 7 illustrates the different convergence curves of α-EM with different values of α. Remember that when $\alpha = -1$, this is log-EM.

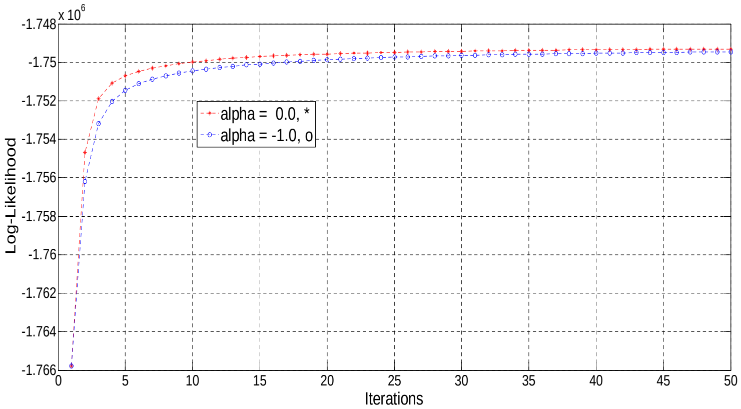

Figure 7: Convergence speed for various alpha

Table 4 shows a speedup comparison. The second column shows that α-EM (the case of $\alpha = 0$) is 45/25=1.80 times faster than log-EM (the case of $\alpha = -1$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The α-EM algorithm didn't require more CPU time per iteration. So we can see that the α-EM algorithm is much faster than the log-EM algorithm by a total CPU-time ratio of 45t/25t=1.80.

Table 4: Speedup Ratio For Factor Model Estimation(p=100, d=20)

α	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
-1.00	45	1t	45t	1.00
0	25	1t	25t	1.80

Now let's consider a higher dimension problem. Let $p = 200$, $d = 30$ and we use the same method to generate our samples. This time we generate 10000 samples with each covariance $D + B_1 * B_1'$ and $D + B_2 * B_2'$. After that we get our 20000 samples. We calculate sample covariance C first, then with the sample covariance we can estimate mixture of factor analysers by both the log-EM algorithm and the

α-EM algorithm. Figure 8 illustrates the different convergence curves of the α-EM algorithm with different values of α. Remember that when $\alpha = -1$, this is log-EM.

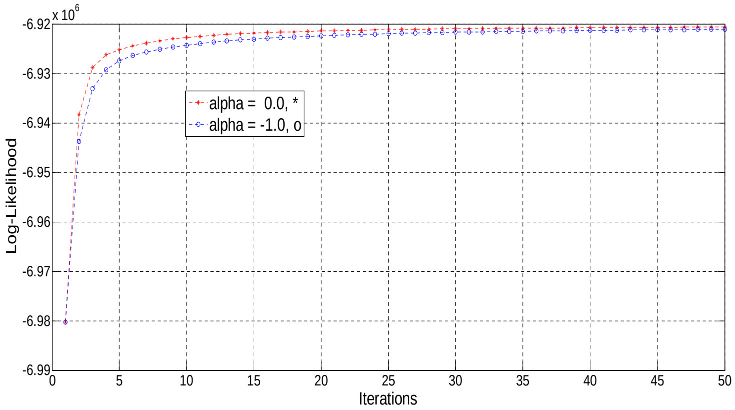

Figure 8: Convergence speed for various alpha

Table 5 shows a speedup comparison. The second column shows that α-EM (the case of $\alpha = 0$) is 45/25=1.80 times faster than log-EM (the case of $\alpha = -1$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The α-EM algorithm didn't require more CPU time per iteration. So we can see that the α-EM algorithm is much faster than the log-EM algorithm by a total CPU-time ratio of 45t/25t=1.80.

Table 5: Speedup Ratio For Factor Model Estimation(p=200, d=30)

α	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
-1.00	45	1t	45t	1.00
0	25	1t	25t	1.80

Therefore, for small dimension problems or large dimension problems the α-EM algorithm will not require more CPU time per iteration than the log-EM algorithm. So as long as the number iterations of the α-EM algorithm is smaller than the log-

EM for the same accuracy, the α-EM algorithm will save us the total computation time. That's why we should choose the α-EM algorithm over the log-EM algorithm.

4.4.2 Mixture of factor models on financial data (k = 3)

Here, first we create three factor structured matrices $D + B_1 * B_1', D + B_2 * B_2'$ and $D + B_3 * B_3'$. We choose $p = 50$, $d = 20$ and generate 2500 samples for each factor model. After that we get our 7500 samples. We calculate sample covariance C first, then with the sample covariance we can estimate mixture of factor models using both the log-EM algorithm and the α-EM algorithm. To do the 1st iteration to obtain $\{\pi_2\}_{j=1}^{3}, \{\Lambda_2\}_{j=1}^{3}$ and Φ_2, we require previous two estimates which are $\{\pi_1\}_{j=1}^{3}, \{\Lambda_1\}_{j=1}^{3}, \Phi_1$ and $\{\pi_0\}_{j=1}^{3}, \{\Lambda_0\}_{j=1}^{3}, \Phi_0$. For $\{\pi_0\}_{j=1}^{3}, \{\Lambda_0\}_{j=1}^{3}$ and Φ_0, we can use a random guess as we described previously. Figure 9 illustrates the different convergence curves of the α-EM algorithm with different values of α. Remember that when $\alpha = -1$, this is log-EM.

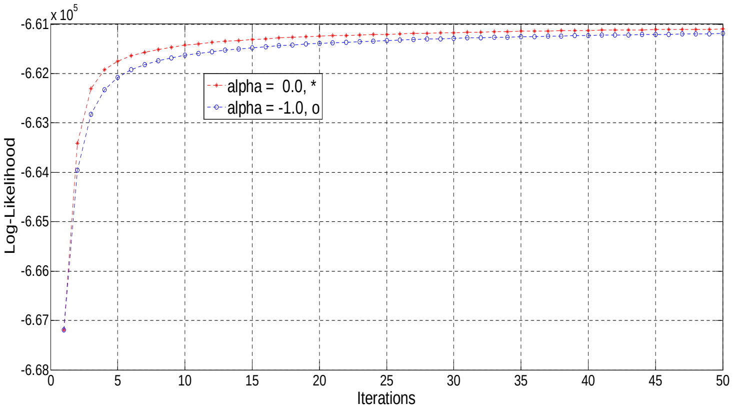

Figure 9: Convergence speed for various alpha

Table 6 shows a speedup comparison. The second column shows that α-EM (the case of $\alpha = 0$) is 45/25=1.80 times faster than log-EM (the case of $\alpha = -1$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The α-EM algorithm didn't require more CPU time per iteration. So we can see that the α-EM algorithm is much faster than the log-EM algorithm by a total CPU-time ratio of 45t/25t=1.80.

Table 6: Speedup Ratio For Factor Model Estimation(p=50, d=20)

α	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
-1.00	45	1t	45t	1.00
0	25	1t	25t	1.80

We also consider a higher dimension problem. Let $p = 100$, $d = 20$ and we use the same method to generate our samples. This time we generate 5000 samples with each covariance $D+B_1*B_1'$, $D+B_2*B_2'$ and $D+B_3*B_3'$. After that we get our 15000 samples. We calculate sample covariance C first, then with the sample covariance we can estimate mixture of factor models using both the log-EM algorithm and the α-EM algorithm. Figure 10 illustrates the different convergence curves of the α-EM algorithm with different values of α. Remember that when $\alpha = -1$, this is log-EM.

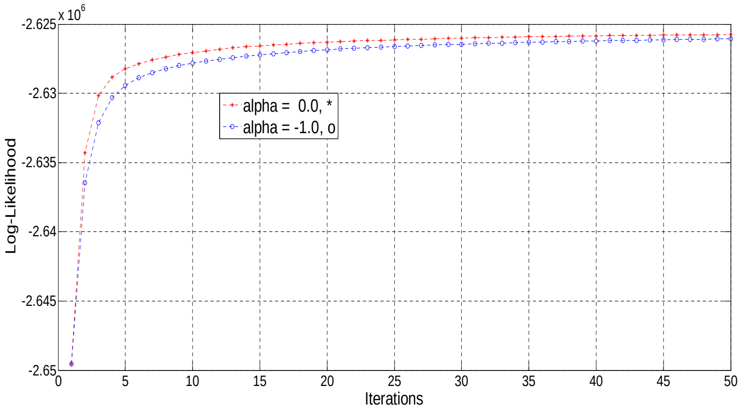

Figure 10: Convergence speed for various alpha

Table 7 shows a speedup comparison. The second column shows that α-EM (the case of $\alpha = 0$) is 45/25=1.80 times faster than log-EM (the case of $\alpha = -1$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The α-EM algorithm didn't require more CPU time per iteration. So we can see that the α-EM algorithm is much faster than the log-EM algorithm by a total CPU-time ratio of 45t/25t=1.80.

Table 7: Speedup Ratio For Factor Model Estimation(p=100, d=20)

α	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
-1.00	45	1t	45t	1.00
0	25	1t	25t	1.80

Therefore, for $k = 2$ or $k = 3$ the α-EM algorithm will not require more CPU time per iteration than the log-EM algorithm. No matter how large dimension of the problem and how many mixtures, the α-EM algorithm is constantly faster than the log-EM algorithm for the same accuracy. So we can choose proper value for α to save us the total computation time when estimates mixture of factor models.

4.5 Concluding Remarks

In this section, we have proposed an α-EM algorithm for fitting mixture of factor models. Unlike the existing log-EM, we use α-logarithm instead of just logarithm. We show that the log-EM algorithm is a special case of the α-EM algorithm. Even after causal approximation and series expansion when $\beta = 1$ which means $\alpha = -1$ we still find that the log-EM is a special case of the α-EM. By choosing proper α, we showed that the α-EM algorithm converges much faster than the log-EM algorithm in mixture of factor models estimation, and also gives more accurate estimates. In CPU-time, as long as we save the results from the previous updates, the α-EM algorithm doesn't require more CPU time than the log-EM algorithm. However more importantly, the speedup in convergence is significant, so the α-EM algorithm can save us the total computation time for the same accuracy.

For problems of different dimensions and different numbers of mixtures, the α-EM algorithm always appears to be faster and better than the log-EM algorithm. There are no dimension constrains or number of mixtures constrains. Besides causal approximation and series expansion, there must be other methods to solve the non-causality, such as moving all the future states on one side of the original update equations (43), (44) and (45). That would be the most accurate method but it is also harder than the approximation method. Thus, further exploration of practical issues pertaining to the α-EM family is needed.

For the α-EM algorithm we used, we focus our attention on the convex divergence (1) because of its general capacity on convex optimization. We would like to consider other types of surrogate functions.

5 Conjugate Gradient Acceleration of the α-EM Algorithm

We apply conjugate gradient method to the α-EM algorithm. Since it has been shown that conjugate gradient method can be used to accelerate the log-EM algorithm, it had been expected that conjugate gradient acceleration of the α-EM algorithm would exist. In this section, it is shown that this is true. The key is that the α-EM step can be viewed (approximately at least) as a generalized gradient, making it natural to apply generalized conjugate gradient methods in an attempt to speed up the α-EM algorithm. The proposed method is relatively simple to implement and can handle problems with a large number of parameters. To demonstrate the effectiveness of the proposed acceleration method, we consider its application to both artificial data and financial data.

5.1 Model Description

We consider the factor analysis model

$$x = \Lambda z + u$$

where x is a vector of observed values, Λ is a $p \times d$ matrix of factor loadings, and z and u are independent normally distributed random vectors with mean 0 and covariance matrices Ψ and Φ. By assumption Φ is diagonal. Following common practice and for simplicity, we have assumed that the mean of x is 0. The covariance matrix of x is

$$\Sigma = \Phi + \Lambda \Psi \Lambda^T$$

We allow priori restrictions that fix arbitrary elements of Φ and Λ at specified values. Following Rubin and Thayer [107], we allow Ψ to be set equal to the identity or be totally free.

The α-logarithm function is defined as follows [81]:

$$L^{(\alpha)}(r) \overset{def}{=} \frac{2}{1+\alpha}\left(r^{\frac{1+\alpha}{2}} - 1\right)$$

and if $r = r(\theta) \in (0, \infty)$ is twice differentiable with respect to θ, the following equalities hold

$$\frac{\partial L^{(\alpha)}(r)}{\partial\theta} = r^{\frac{1+\alpha}{2}} \frac{\partial \log r}{\partial\theta} \tag{62}$$

Given N independent observations x_i, let $C_{xx} = \sum_{j=1}^{N} \frac{x_i x_i'}{N}$. Given C_{xx}, the α-log-likelihood of (Ψ, Φ, Λ) is

$$\begin{aligned}
f(\theta) &= \frac{2}{1+\alpha}\left(\left(\prod_{i=1}^{N}(2\pi)^{-p/2}\,|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}x_i^T\Sigma x_i\right)\right)^{\frac{1+\alpha}{2}} - 1\right)\\
&= \frac{2}{1+\alpha}\left(\left((2\pi)^{-pN/2}\,|\Sigma|^{-N/2}\exp\left(-\frac{1}{2}\sum_{i=1}^{N}x_i^T\Sigma x_i\right)\right)^{\frac{1+\alpha}{2}} - 1\right)\\
&= \frac{2}{1+\alpha}\left(\left((2\pi)^{-pN/2}\,|\Sigma|^{-N/2}\exp\left(-\frac{N}{2}trS\Sigma^{-1}\right)\right)^{\frac{1+\alpha}{2}} - 1\right)\\
&= \frac{2}{1+\alpha}\left((L)^{\frac{1+\alpha}{2}} - 1\right)
\end{aligned}$$

where we define

$$\begin{aligned}
L &= (2\pi)^{-pN/2}\,|\Sigma|^{-N/2}\exp\left(-\frac{N}{2}trS\Sigma^{-1}\right)\\
l &= \log L = -\frac{N}{2}\left(p\log 2\pi + \log|\Sigma| + trS\Sigma^{-1}\right)
\end{aligned}$$

From equality (62), we have

$$\frac{\partial f}{\partial\theta} = L^{\frac{1+\alpha}{2}}\frac{\partial \log L}{\partial\theta} = L^{\frac{1+\alpha}{2}}\frac{\partial l}{\partial\theta}$$

The derivatives of f are given by (see Appendix E for details) :

$$\begin{aligned}
\frac{\partial f}{\partial\Lambda} &= L^{\frac{1+\alpha}{2}} N\Sigma^{-1}(S-\Sigma)\Sigma^{-1}\Lambda\Phi\\
\frac{\partial f}{\partial\Phi} &= L^{\frac{1+\alpha}{2}}\frac{N}{2}diag[\Sigma^{-1}(S-\Sigma)\Sigma^{-1}]\\
\frac{\partial f}{\partial\Psi} &= L^{\frac{1+\alpha}{2}}\Lambda^T\Sigma^{-1}(S-\Sigma)\Sigma^{-1}\Lambda
\end{aligned}$$

If there are n free parameters in Λ, Ψ and Φ, then f is defined on $\mathbb{R}^n$. We consider metrics on $\mathbb{R}^n$ which are defined by a positive definite matrix W via the inner product

$$\langle \theta_1, \theta_2 \rangle = \theta_1' W \theta_2 \tag{63}$$

The gradient of f in the metric defined by (63) is

$$\nabla f(\theta) = g(\theta) = W^{-1} s(\theta)$$

where

$$s(\theta) = \left[\frac{\partial f}{\partial \Lambda}, \frac{\partial f}{\partial \Phi}, \frac{\partial f}{\partial \Psi} \right]$$

It is natural to call $s(\theta)$ the raw gradient of $f(\theta)$ at θ. It is twice the negative of the Fisher-score vector. Let $\widetilde{\theta}$ be the modified α-EM update of θ_k and let

$$\begin{aligned} g_k &= -(\widetilde{\theta} - \theta_k) \\ d_k &= -g_k \end{aligned}$$

Choosing α to minimize $f(\theta_k + ad_k)$ gives an optimized version of the modified α-EM algorithm.

5.2 Empirical Results

In the following, we do some comparisons between accelerated α-EM and ordinary α-EM with both artificial data and real financial data.

5.2.1 Accelerated α-EM on artificial data

First we create a factor structured matrix $D + B * B'$ with random entries. D is $p \times p$ and B_1, B_2 are $p \times d$. The elements of D are exponentially distributed idiosyncratic variances and B_1, B_2 have Gaussian distributed factor loads. We choose $p = 100$ and $d = 20$ then generate 5000 samples with covariance $D + B * B'$. After that we calculate sample covariance C first, then with the sample covariance we can estimate factor model by both α-EM algorithm and accelerated α-EM algorithm. Figure 11 shows that the convergence speed of accelerated α-EM is faster than α-EM in terms of the log-likelihood.

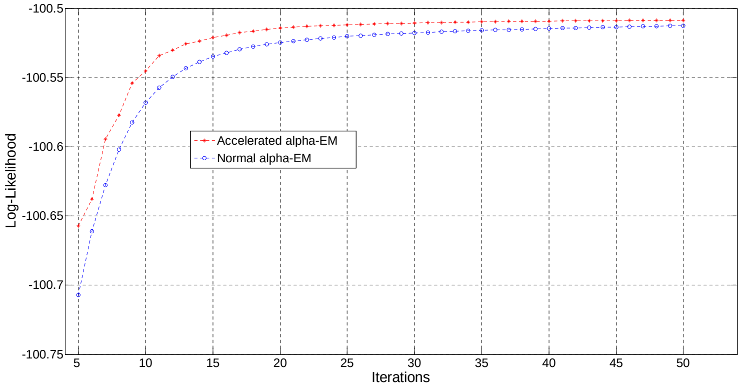

Figure 11: Convergence Speed Comparison

Table 8 shows a speedup comparison. The second column shows that accelerated α-EM (the case of $\alpha = 0$) is 50/25=2.0 times faster than ordinary α-EM (the case of $\alpha = 0$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The accelerated α-EM algorithm requires 50% more CPU time per iteration. So we can see that the accelerated α-EM algorithm is faster than the ordinary α-EM algorithm by a total CPU-time ratio of 50t/37.5t=1.33.

Table 8: Speedup Ratio Comparison(p=100, d=20)

$\alpha = 0$	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
Ordinary α-EM	50	1t	50t	1.00
Accelerated α-EM	25	1.5t	37.5t	1.33

Figure 12 illustrates the different rates at which the Hellinger distance decreases in the accelerated α-EM algorithm and ordinary α-EM algorithm. We can see that in terms of the Hellinger distance, accelerated α-EM still preforms better than ordinary

α-EM.

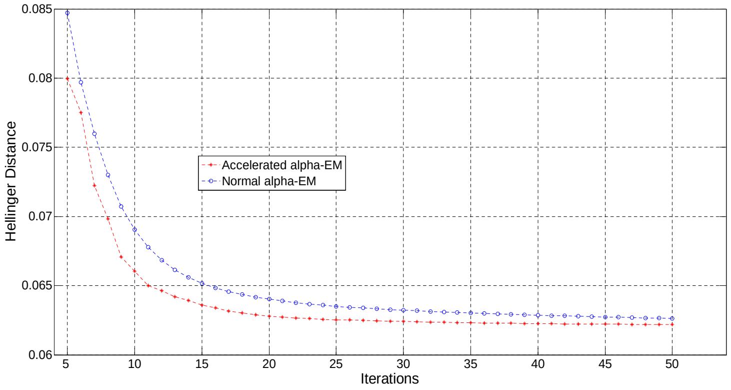

Figure 12: Convergence speed comparison

Table 9 shows a speedup comparison. The second column shows that the accelerated α-EM algorithm (the case of $\alpha = 0$) is 50/22=2.3 times faster than the ordinary α-EM algorithm (the case of $\alpha = 0$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The accelerated α-EM algorithm requires 50% more CPU time per iteration. So we can see that the accelerated α-EM algorithm is faster than the ordinary α-EM algorithm by a total CPU-time ratio of 50t/33t=1.5.

Table 9: Speedup Ratio Comparison(p=100, d=20)

$\alpha = 0$	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
α-EM	50	1t	50t	1.00
Accelerated α-EM	22	1.5t	33t	1.5

In the accelerated α-EM algorithm we need to do a line search. In that line search we need to calculate gradients of α-log-likelihood function. This requires extra time. This extra time can be dominant in total CPU-time when the dimension of the problem gets large.

5.2.2 Accelerated α-EM on financial data

Here, we download daily close prices of each member of S&P500 from Jan-01-2001 to Aug-31-2013 from Yahoo. Then we select members which have prices since Jan-01-2001 and we got 426 members. Let's randomly choose 100 members from 426 members. After we get 100 members we calculate the sample covariance C, then we use the sample covariance to calculate the factor model with 20 factors by the accelerated α-EM algorithm and the ordinary α-EM algorithm. Figure 13 shows that the convergence speed of accelerated α-EM is faster than ordinary α-EM in terms of log-likelihood.

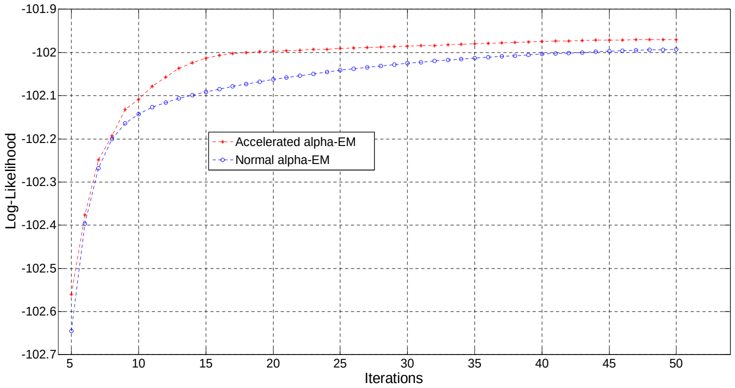

Figure 13: Convergence speed comparison

Table 10 shows a speedup comparison. The second column shows that the accelerated α-EM algorithm (the case of $\alpha = 0$) is 49/23=2.1 times faster than the ordinary α-EM algorithm (the case of $\alpha = 0$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The accelerated α-EM algorithm requires 50% more CPU time per iteration. So we can see that the accelerated α-EM algorithm is faster than the ordinary α-EM algorithm by a total CPU-time ratio of 49t/34.5t=1.42.

Table 10: Speedup Ratio Comparison(p=100, d=20)

$\alpha = 0$	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
α-EM	49	1t	49t	1.00
Accelerated α-EM	23	1.5t	34.5t	1.42

Figure 14 illustrates different decreasing speed in the Hellinger distance of accelerated α-EM and ordinary α-EM. We can see that in terms of the Hellinger distance, accelerated α-EM still preforms better than ordinary α-EM.

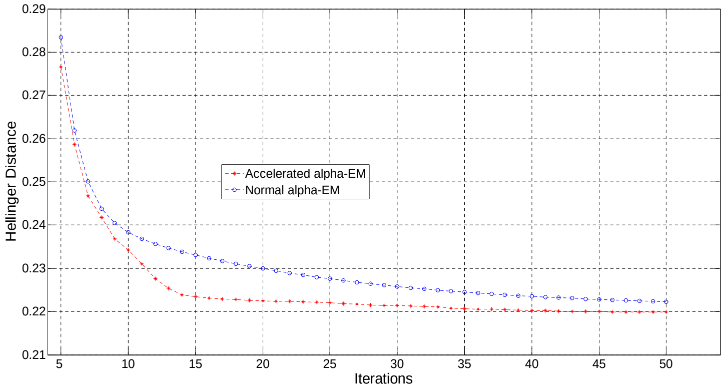

Figure 14: Convergence speed comparison

Table 11 shows a speedup comparison. The second column shows that the accelerated α-EM algorithm (the case of $\alpha = 0$) is 49/21=2.3 times faster than the ordinary α-EM algorithm (the case of $\alpha = 0$) for the same convergence. The third and fourth columns show a more practical comparison based upon CPU time. We use t to denote the total time per iteration. The accelerated α-EM algorithm requires 50% more CPU time per iteration. So we can see that the accelerated α-EM algorithm is faster than the ordinary α-EM algorithm by a total CPU-time ratio of

49t/31.5t=1.55.

Table 11: Speedup Ratio Comparison(p=100, d=20)

$\alpha = 0$	Iterations	Time per Iteration	Total CPU-Time	Speedup Ratio
α-EM	49	1t	49t	1.00
Accelerated α-EM	21	1.5t	31.5t	1.55

Again, the extra CPU time of the accelerated α-EM varies at different dimensions.

5.3 Concluding Remarks

In this section, we apply conjugate gradient acceleration to the α-EM algorithm. The α-EM often works well and what we have done here is to attempt to extend the range of its applicability without sacrificing too much of the simplicity it usually enjoys. We compute gradients of α-log-likelihood of (Ψ, Φ, Λ). We show that the α-EM algorithm can also be accelerated by conjugate gradient method.

The accelerated α-EM algorithm requires extra CPU-time to compute the best step size in each iteration. We need to calculate a common piece, $\Sigma^{-1}(S - \Sigma)\Sigma^{-1}$, in the gradient of the α-log-likelihood function where Σ is the factor model and S is the sample covariance. The extra CPU-time varies at different admissions of S. In order to calculate the inverse of Σ, we apply the Woodbury inverse lemma to it. Instead of computing the inverse of a $p \times p$ problem, we only need to compute a $d \times d$ problem.

We do empirical studies with both artificial data and financial data. We know that the accelerated α-EM algorithm requires more CPU time than the ordinary α-EM algorithm. However more importantly, when the speedup in convergence is more significant, the accelerated α-EM algorithm can still save us the total computation time for the same accuracy. We compare the covergence speed of the accelerated α-EM algorithm and the ordinary α-EM algorithm in terms of both log-likelihood and the Hellinger distance. The accelerated α-EM is about 40% faster for 100 dimensional problems.

6 Band Fraction Representation

We present band fraction representations as a new structure for covariance estimates. We believe that this new band fraction representation performs better than the well known and widely used factor model under the Hellinger distance. One of the important reasons that factor models are so popular is that factor model is easy to interpret and can be directly connected to financial sectors. We introduce a band fraction representation of covariance matrices, in practice this new structure is closer to covariance matrices than factor model under the Hellinger distance. Because, intuitively factor model is a special case of band fraction representation which we can use for any factor model. But there is no easy interpretation of the band fraction similar to that of the factor model. Empirical results from real financial data are given. In order to make it clear we consider simple cases first and use the same procedure for complex cases by induction.

6.1 Semiseparability rank 2 and bandwidth 2

Let's start from a simple case first. Suppose S is a 6×6 lower triangular semiseparable matrix of semiseparability rank 2. According to the above theory we can find

M and N with bandwidth $d = 2$,s.t. $S = M^{-1}N$ or $MS = N$.

$$\begin{bmatrix} M_{1,1} & & & & & \\ M_{2,1} & M_{2,2} & & & & \\ & M_{3,2} & M_{3,3} & & & \\ & & M_{4,3} & M_{4,4} & & \\ & & & M_{5,4} & M_{5,5} & \\ & & & & M_{6,5} & M_{6,6} \end{bmatrix} \begin{bmatrix} S_{1,1} & & & & & \\ S_{2,1} & S_{2,2} & & & & \\ S_{3,1} & S_{3,2} & S_{3,3} & & & \\ S_{4,1} & S_{4,2} & S_{4,3} & S_{4,4} & & \\ S_{5,1} & S_{5,2} & S_{5,3} & S_{5,4} & S_{5,5} & \\ S_{6,1} & S_{6,2} & S_{6,3} & S_{6,4} & S_{6,5} & S_{6,6} \end{bmatrix}$$

$$= \begin{bmatrix} N_{1,1} & & & & & \\ N_{2,1} & N_{2,2} & & & & \\ & N_{3,2} & N_{3,3} & & & \\ & & N_{4,3} & N_{4,4} & & \\ & & & N_{5,4} & N_{5,5} & \\ & & & & N_{6,5} & N_{6,6} \end{bmatrix}$$

In order to solve it, let's start from the bottom. We can get

$$\begin{bmatrix} S_{5,1} & S_{6,1} \\ S_{5,2} & S_{6,2} \\ S_{5,3} & S_{6,3} \\ S_{5,4} & S_{6,4} \\ S_{5,5} & S_{6,5} \end{bmatrix} \begin{bmatrix} M_{6,5} \\ M_{6,6} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ N_{6,5} \end{bmatrix}$$

and because S has semiseparability rank 2 then we have unique solution of $M_{6,5}, M_{6,6}$ for nonzero $N_{6,5}$ (We can set $N_{6,5} = 1$). We repeat this procedure until the 3rd row, because solving the first two rows is trivial.

$$\begin{bmatrix} M_{1,1} & \\ M_{2,1} & M_{2,2} \end{bmatrix} \begin{bmatrix} S_{1,1} & \\ S_{2,1} & S_{2,2} \end{bmatrix} = \begin{bmatrix} N_{1,1} & \\ N_{2,1} & N_{2,2} \end{bmatrix}$$

6.2 Semiseparability rank 3 and bandwidth 3

Now, let's upgrade the problem to semiseparability rank 3 and bandwidth $d = 3$.

$$\begin{bmatrix} M_{1,1} & & & & & \\ M_{2,1} & M_{2,2} & & & & \\ M_{3,1} & M_{3,2} & M_{3,3} & & & \\ & M_{4,2} & M_{4,3} & M_{4,4} & & \\ & & M_{5,3} & M_{5,4} & M_{5,5} & \\ & & & M_{6,4} & M_{6,5} & M_{6,6} \end{bmatrix} \begin{bmatrix} S_{1,1} & & & & & \\ S_{2,1} & S_{2,2} & & & & \\ S_{3,1} & S_{3,2} & S_{3,3} & & & \\ S_{4,1} & S_{4,2} & S_{4,3} & S_{4,4} & & \\ S_{5,1} & S_{5,2} & S_{5,3} & S_{5,4} & S_{5,5} & \\ S_{6,1} & S_{6,2} & S_{6,3} & S_{6,4} & S_{6,5} & S_{6,6} \end{bmatrix}$$

$$= \begin{bmatrix} N_{1,1} & & & & & \\ N_{2,1} & N_{2,2} & & & & \\ N_{3,1} & N_{3,2} & N_{3,3} & & & \\ & N_{4,2} & N_{4,3} & N_{4,4} & & \\ & & N_{5,3} & N_{5,4} & N_{5,5} & \\ & & & N_{6,4} & N_{6,5} & N_{6,6} \end{bmatrix}$$

We also start from the bottom. We can get

$$\begin{bmatrix} S_{4,1} & S_{5,1} & S_{6,1} \\ S_{4,2} & S_{5,2} & S_{6,2} \\ S_{4,3} & S_{5,3} & S_{6,3} \\ S_{4,4} & S_{5,4} & S_{6,4} \\ 0 & S_{5,5} & S_{6,5} \end{bmatrix} \begin{bmatrix} M_{6,4} \\ M_{6,5} \\ M_{6,6} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ N_{6,4} \\ N_{6,5} \end{bmatrix}$$

and because S has semiseparability rank 3, then $rank(S_{4:6,1:3}) = 2$. Then we can set $M_{6,6} = 1$ and solve $M_{6,4}$ and $M_{6,5}$ with the first three rows.

$$\begin{bmatrix} S_{4,1} & S_{5,1} \\ S_{4,2} & S_{5,2} \\ S_{4,3} & S_{5,3} \end{bmatrix} \begin{bmatrix} M_{6,4} \\ M_{6,5} \end{bmatrix} = \begin{bmatrix} -S_{6,1} \\ -S_{6,2} \\ -S_{6,3} \end{bmatrix}$$

Because we can find a combination of $S_{4,1:3}$ and $S_{5,1:3}$ to represent $-S_{6,1:3}$. Then we can compute $N_{6,4}$ and $N_{6,5}$ by

$$\begin{bmatrix} S_{4,4} & S_{5,4} & S_{6,4} \\ 0 & S_{5,5} & S_{6,5} \end{bmatrix} \begin{bmatrix} M_{6,4} \\ M_{6,5} \\ M_{6,6} \end{bmatrix} = \begin{bmatrix} N_{6,4} \\ N_{6,5} \end{bmatrix}$$

We repeat this procedure until the 4th row, and we set $M_{4,4} = 1$ then we have:

$$\begin{bmatrix} S_{2,1} & S_{3,1} & S_{4,1} \\ S_{2,2} & S_{3,2} & S_{4,2} \\ 0 & S_{3,3} & S_{4,3} \end{bmatrix} \begin{bmatrix} M_{4,2} \\ M_{4,3} \\ M_{4,4} \end{bmatrix} = \begin{bmatrix} 0 \\ N_{4,2} \\ N_{4,3} \end{bmatrix} \Rightarrow \begin{bmatrix} S_{2,1} & S_{3,1} \\ S_{2,2} & S_{3,2} \end{bmatrix} \begin{bmatrix} M_{4,2} \\ M_{4,3} \end{bmatrix} = \begin{bmatrix} -S_{4,1} \\ -S_{4,2} + N_{4,2} \end{bmatrix}$$

We can set a random value to $N_{4,2}$, in order to solve for $M_{4,3}$ and $M_{4,3}$. Then we can compute $N_{4,3}$ by

$$\begin{bmatrix} 0 & S_{3,3} & S_{4,3} \end{bmatrix} \begin{bmatrix} M_{4,2} \\ M_{4,3} \\ M_{4,4} \end{bmatrix} = \begin{bmatrix} N_{4,3} \end{bmatrix}$$

After that it is trivial to solve the first three rows.

$$\begin{bmatrix} M_{1,1} & & \\ M_{2,1} & M_{2,2} & \\ M_{3,1} & M_{3,2} & M_{3,3} \end{bmatrix} \begin{bmatrix} S_{1,1} & & \\ S_{2,1} & S_{2,2} & \\ S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} N_{1,1} & & \\ N_{2,1} & N_{2,2} & \\ N_{3,1} & N_{3,2} & N_{3,3} \end{bmatrix}$$

Note that this procedure is different from what we have for semiseparability rank and bandwidth 2, but this procedure can be applied when semiseparability rank and bandwidth are 2.

6.3 Semiseparability rank d and bandwidth d

Now, let's consider a general case that S is $n \times n$ lower triangular semiseparable matrix of semiseparability rank d. We want to find M and N with bandwidth d.

$$\begin{bmatrix} M_{1,1} & & & & & \\ \vdots & \ddots & & & & \\ M_{d,1} & \cdots & M_{d,d} & & & \\ 0 & M_{d+1,2} & \cdots & M_{d+1,d+1} & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ 0 & \cdots & 0 & M_{n,n-d+1} & \cdots & M_{n,n} \end{bmatrix} \begin{bmatrix} S_{1,1} & & & & & \\ \vdots & \ddots & & & & \\ S_{d,1} & \cdots & S_{d,d} & & & \\ S_{d+1,1} & \cdots & \cdots & S_{d+1,d+1} & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ S_{n,1} & \cdots & \cdots & \cdots & \cdots & S_{n,n} \end{bmatrix}$$

$$= \begin{bmatrix} N_{1,1} & & & & & \\ \vdots & \ddots & & & & \\ N_{d,1} & \cdots & N_{d,d} & & & \\ 0 & N_{d+1,2} & \cdots & N_{d+1,d+1} & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ 0 & \cdots & 0 & N_{n,n-d+1} & \cdots & N_{n,n} \end{bmatrix}$$

Again, we start from the bottom and get:

$$\begin{bmatrix} S_{n-d+1,1} & \cdots & S_{n,1} \\ \vdots & \vdots & \vdots \\ S_{n-d+1,n-d} & \cdots & S_{n,n-d} \\ S_{n-d+1,n-d+1} & \cdots & S_{n,n-d+1} \\ \vdots & \vdots & \vdots \\ 0 & \cdots & S_{n,n-1} \end{bmatrix} \begin{bmatrix} M_{n,n-d+1} \\ \vdots \\ M_{n,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ N_{n,n-d+1} \\ \vdots \\ N_{n.n-1} \end{bmatrix}$$

Since S has semiseparability rank d, then $rank(S_{n-d+1:n,1:n-d}) = d - 1$, we can set $M_{n,n} = 1$ and solve $M_{n,n-d+1}, \ldots, M_{n,n-1}$ with the first $n - d$ rows.

$$\begin{bmatrix} S_{n-d+1,1} & \cdots & S_{n-1,1} \\ \vdots & \vdots & \vdots \\ S_{n-d+1,n-d} & \cdots & S_{n-1,n-d} \end{bmatrix} \begin{bmatrix} M_{n,n-d+1} \\ \vdots \\ M_{n,n-1} \end{bmatrix} = \begin{bmatrix} -S_{n,1} \\ \vdots \\ -S_{n,n-d} \end{bmatrix}$$

Because we can find a combination of $S_{n-d,1:n-d}, \ldots, S_{n-1,1:n-d}$ to represent $-S_{n,1:n-d}$. Then we can compute $N_{n,n-d+1}, \ldots N_{n.n-1}$ by:

$$\begin{bmatrix} S_{n-d+1,n-d+1} & \cdots & S_{n,n-d+1} \\ \vdots & \vdots & \vdots \\ 0 & \cdots & S_{n,n-1} \end{bmatrix} \begin{bmatrix} M_{n,n-d+1} \\ \vdots \\ M_{n,n} \end{bmatrix} = \begin{bmatrix} N_{n,n-d+1} \\ \vdots \\ N_{n.n-1} \end{bmatrix}$$

We can repeat the same procedure until the $d+1$th row, we get:

$$\begin{bmatrix} S_{2,1} & S_{3,1} & \cdots & S_{d+1,1} \\ S_{2,2} & S_{3,2} & \cdots & S_{d+1,2} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & S_{d,d} & S_{d+1,d} \end{bmatrix} \begin{bmatrix} M_{d+1,2} \\ M_{d+1,3} \\ \vdots \\ M_{d+1,d+1} \end{bmatrix} = \begin{bmatrix} 0 \\ N_{d+1,2} \\ \vdots \\ N_{d+1,d+1} \end{bmatrix}$$

and we set $M_{d+1,d+1} = 1$ then the first two rows would be:

$$\begin{bmatrix} S_{2,1} & S_{3,1} & \cdots & S_{d,1} \\ S_{2,2} & S_{3,2} & \cdots & S_{d,2} \end{bmatrix} \begin{bmatrix} M_{d+1,2} \\ M_{d+1,3} \\ \vdots \\ M_{d+1,d} \end{bmatrix} = \begin{bmatrix} -S_{d+1,1} \\ -S_{d+1,2} + N_{d+1,2} \end{bmatrix}$$

For $d+1$th row we can solve $M_{d+1,2}, \ldots, M_{d+1,d}$ by setting a random value to $N_{d+1,2}$. Then we can compute $N_{d+1,3}, \ldots, N_{d+1,d}$ by

$$\begin{bmatrix} 0 & S_{3,3} & \cdots & S_{d+1,3} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & S_{d,d} & S_{d+1,d} \end{bmatrix} \begin{bmatrix} M_{d+1,2} \\ M_{d+1,3} \\ \vdots \\ M_{d+1,d+1} \end{bmatrix} = \begin{bmatrix} N_{d+1,3} \\ \vdots \\ N_{d+1,d} \end{bmatrix}$$

After that it is trivial to solve the first d rows.

$$\begin{bmatrix} M_{1,1} & & \\ \vdots & \ddots & \\ M_{d,1} & \cdots & M_{d,d} \end{bmatrix} \begin{bmatrix} S_{1,1} & & \\ \vdots & \ddots & \\ S_{d,1} & \cdots & S_{d,d} \end{bmatrix} = \begin{bmatrix} N_{1,1} & & \\ \vdots & \ddots & \\ N_{d,1} & \cdots & N_{d,d} \end{bmatrix}$$

So when $d >= n$, we can always set N to be identity matrix and M will be the inverse of S.

6.4 Empirical Results

In the following, we do some comparisons between factor models and band fraction representations with both artificial data and real financial data.

6.4.1 Artificial data

Let's assume that the covariance matrix has a factor structure $C = D + V * V'$ where D is $p \times p$ and V is $p \times d$.Then we fit a band fraction representation. Now the covariance matrix is a diagonal matrix plus a semiseparable matrix and the semiseparability rank is $d + 1$. According to the method we have above we can find M and N with bandwidth $d + 1$. So even if the market is a factor structure we can use band fraction representation. But the factor model can't estimate band fraction representation. We fit factor model with d factors and we find the Hellinger distance between factor model and band fraction representation with bandwidth $d + 1$ is large. We will show empirical results later.

Now, let's consider a special case that C is a $p \times p$ matrix:

$$\begin{array}{cccccc}
1 & & & & & 1/2 \\
 & \ddots & & & & \\
 & & 1 & 1/2 & & \\
 & & 1/2 & 1 & & \\
 & & & & \ddots & \\
1/2 & & & & & 1
\end{array}$$

We can fit both the factor model and band fraction representation. In a factor model a permutation won't change the results but a permutation can help in the band fraction representation. According to Cuthill–McKee algorithm which is an algorithm to permute a sparse matrix that has a symmetric sparsity pattern into a band matrix form with a small bandwidth, we will get Cp:

$$\begin{array}{cccccc}
1 & 1/2 & & & & \\
1/2 & 1 & & & & \\
& & \ddots & \ddots & & \\
& & \ddots & \ddots & & \\
& & & & 1 & 1/2 \\
& & & & 1/2 & 1
\end{array}$$

For C_P, we can find a very good band fraction representation with only bandwidth 2. Because the semiseparability rank of C is $ceil(p/2)$ and it is 1 of C_P. For example, suppose $p = 100$ we chose $d = 20$ for factor model. Table 12 shows the Hellinger distance between factor model/band Fraction and C/C_P.

Table 12: Hellinger distance comparison

Hellinger distance	$d = 20$	$d = 1$
Factor Model and C	0.6956	0.8567
Factor Model and C_P	0.6956	0.8567
Band Fraction and C	0.6956	0.8567
Band Fraction and C_P	0	0

6.4.2 Low dimension data

Here, we apply both the factor model and the band fraction representation to the same data used in Rubin and Thayer 1982 [107].

$$C_{xx} = \begin{bmatrix} 1.0 & 0.554 & 0.227 & 0.189 & 0.461 & 0.506 & 0.408 & 0.280 & 0.241 \\ 0.554 & 1.0 & 0.296 & 0.219 & 0.479 & 0.530 & 0.425 & 0.311 & 0.311 \\ 0.227 & 0.296 & 1.0 & 0.769 & 0.237 & 0.243 & 0.304 & 0.718 & 0.730 \\ 0.189 & 0.219 & 0.769 & 1.0 & 0.212 & 0.226 & 0.291 & 0.681 & 0.661 \\ 0.461 & 0.479 & 0.237 & 0.212 & 1.0 & 0.520 & 0.514 & 0.313 & 0.245 \\ 0.506 & 0.530 & 0.243 & 0.226 & 0.520 & 1.0 & 0.473 & 0.348 & 0.290 \\ 0.408 & 0.425 & 0.304 & 0.291 & 0.514 & 0.473 & 1.0 & 0.374 & 0.306 \\ 0.280 & 0.311 & 0.718 & 0.681 & 0.313 & 0.348 & 0.374 & 1.0 & 0.672 \\ 0.241 & 0.311 & 0.730 & 0.661 & 0.245 & 0.290 & 0.306 & 0.672 & 1.0 \end{bmatrix}$$

In order to do a fair comparison, we choose $d = 2$ as the number of factors and $d+1$ as the bandwidth of M and N. Figure 15 shows that the band fraction representation is much closer to the sample covariance matrix in terms of the log-likelihood.

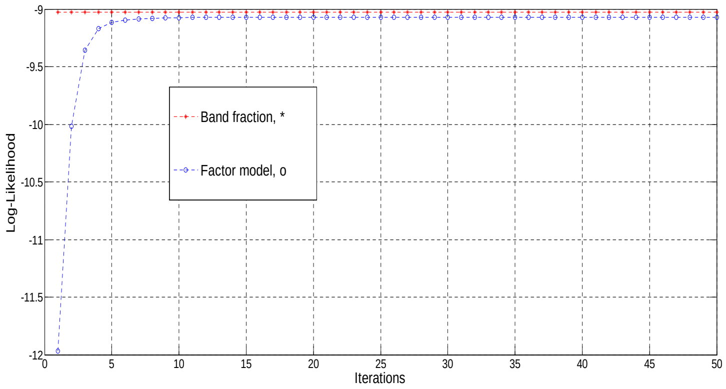

Figure 15: Log-likelihood comparison between factor model and band fraction representation

The log-likelihood on the y-axis is calculated by

$$LL = \log\det((\Phi + \Lambda\Lambda')^{-1}C_{xx}) - trace((\Phi + \Lambda\Lambda')^{-1}C_{xx})$$

so for the optimal results we have $C_{xx} = \Phi + \Lambda\Lambda'$ then $LL = -p$ where p is the dimensionality of the problem. Whether you can reach the optimal results depends on the condition of the problem. Factor analysis can be only as good as the data allows. Note that the log-likelihood value of factor model after 50 iterations is -9.0712 and the log-likelihood value of band fraction representation is -9.0249.

Besides the log-likelihood, we can also compare the Hellinger distance between factor model and sample covariance, and the band fraction representation and sample covariance. The Hellinger distance is the same under coordinate change. In our case $P_0 \sim N(0, C_{xx})$ and $P_1 \sim N(0, \Phi + \Lambda\Lambda')$ then we have:

$$\begin{aligned} H^2(C_{xx}, \Phi + \Lambda\Lambda') &= H^2(I, Cxx^{-1/2}(\Phi + \Lambda\Lambda')Cxx^{-1/2}) \\ &= H^2(I, \Sigma) \end{aligned}$$

where

$$\Sigma = Cxx^{-1/2}(\Phi + \Lambda\Lambda')Cxx^{-1/2}$$

We have

$$\begin{aligned} H^2(\Sigma, I) &= \frac{1}{2}\int \left(\sqrt{\Sigma} - \sqrt{I}\right)^2 dx \\ &= 1 - \left|\frac{\Sigma^{1/2} + \Sigma^{-1/2}}{2}\right|^{-1/2} \end{aligned}$$

assume that

$$\lambda_k = eig(\Sigma^{1/2}) = sqrt(eig(\Sigma)) \text{ where } k = 1, \ldots n$$

then we get (see Appendix C for details)

$$\begin{aligned} H^2(\Sigma, I) &= 1 - \prod_{k=1}^{n}\left(\frac{\lambda_k^{-1} + \lambda_k}{2}\right)^{-1/2} \\ &= 1 - \prod_{k=1}^{n}\sqrt{\frac{2\lambda_k}{1 + \lambda_k^2}} \end{aligned}$$

Figure 16 illustrates that band fraction representation is closer to the sample covariance matrix in terms of the Hellinger distance. Although it looks like they are close we only did 50 iterations, the difference between factor model and band fraction representation is big enough that EM algorithm won't change that much after 50 iterations.

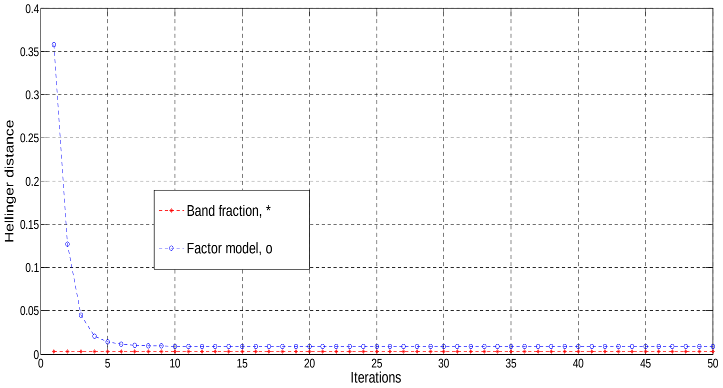

Figure 16: Hellinger distance comparison between factor model and band fraction representation

Note that the log-likelihood value of factor model after 50 iterations is 0.0086 and the log-likelihood value of band fraction representation is 0.0031. From [94] we know that the Hellinger distance is actually bounded by the Information distance.

$$H\left(\Theta_0, \Theta_1\right) \leq \frac{1}{\sqrt{8}} I\left(\Theta_0, \Theta_1\right)$$

Especially for sufficiently small $I\left(\Theta_0, \Theta_1\right)$, we have:

$$\frac{K}{\sqrt{8}} I\left(\Theta_0, \Theta_1\right) \leq H\left(\Theta_0, \Theta_1\right)$$

where $0 < K < 1$. This shows that the upper bound of the Hellinger distance is the best possible. So if we can get the the sample covariance and factor model close in the Hellinger distance, it also implies that they are close in Information distance.

6.4.3 High dimension financial data

Here, we download daily close prices of each member of S&P500 from Jan-01-2001 to Aug-31-2013 from Yahoo. Then we select members which have prices since Jan-01-2001 and we got 426 members. Let's randomly choose 400 members from 426 members. After we get 400 members we calculate the sample covariance C first, then with the sample covariance we can calculate the factor model with 100 factors by the EM algorithm and band fraction representation with bandwidth 101 by the semiseparable approximate factorization. Note that we do 50 iterations in the EM algorithm. Figure 17 shows that band fraction representation is closer to the sample covariance matrix in terms of the log-likelihood.

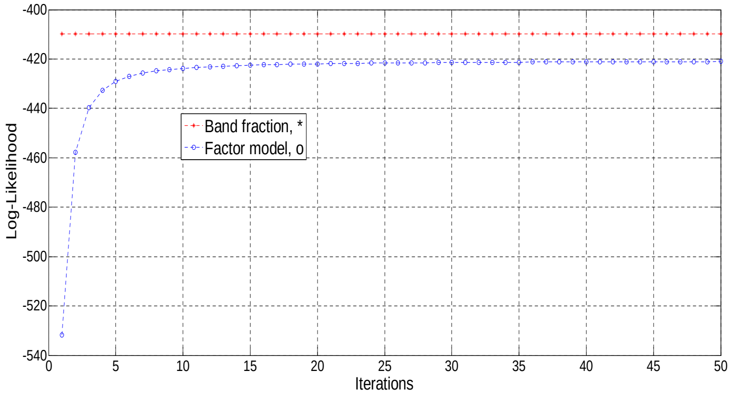

Figure 17: Log-likelihood comparison between two models and sample covariance

Figure 18 illustrates that band fraction representation is more close to the sample covariance matrix in terms of the Hellinger distance. We can see the factor model is not even close to band fraction representation in terms of both log-likelihood and the Hellinger distance. Although we only did 50 iterations in the EM algorithm, it is enough to show that factor model can't reach band fraction representation. The higher the dimension of the problem, the bigger the difference between factor model and band fraction representations.

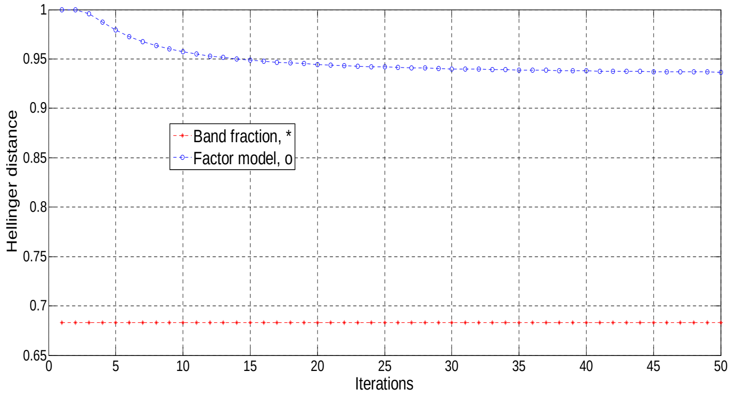

Figure 18: Hellinger distance comparison between two models and sample covariance

We also fit factor model with d factors for band fraction representation $C_B = (M\backslash N)(M\backslash N)^T$ with $d+1$ bandwidth M and N which we estimate from C. Figure 19 shows that if the market is really a band fraction representation we shouldn't use factor models to estimate it, because the Hellinger distance is not close to zero. On the contrary, if the market is really a factor model we can use band fraction representation.

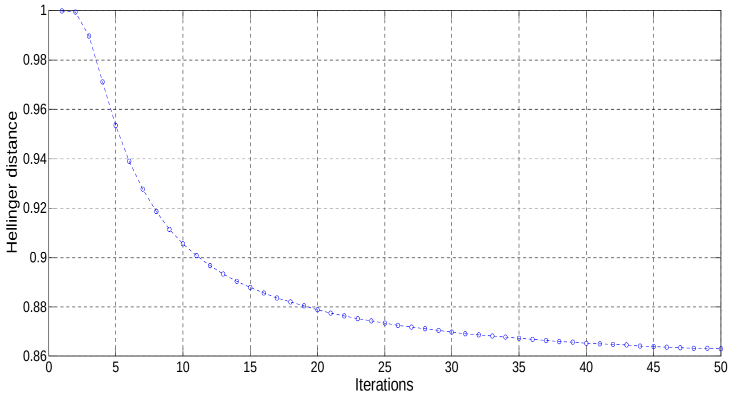

Figure 19: Hellinger distance between factor model and band fraction representation

Instead of comparing one d we choose several ds to compare. Figure 20 illustrates the different log-likelihood values of various d for both factor model and band fraction representation. Note that we do 50 iterations in the EM algorithm.

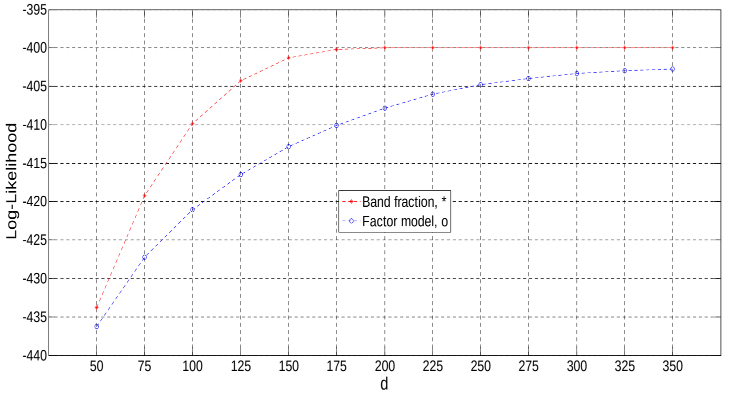

Figure 20: Log-likelihood comparison of various alpha

TABLE 13 shows the values of Figure 5. We can see that when d is greater than 200 the log-likelihood value of band fraction representation is -400. Because the largest semiseparability rank of the Cholesky factor of the covariance matrix is ceil(p/2). When $d \geq ceil(p/2)$, then the semiseparable approximate factorization would be really close to the Cholesky factorization. That's why with the band fraction representation we get really close to the covariance matrix when $d \geq 200$. Note that ceil(n) means the smallest integer that is greater than or equals to n.

Table 13: Log-likelihood Comparison (p=400)

Log-likelihood	$d = 50$	100	150	200	250	300	350
Factor Model	-436.3	-421.0	-412.8	-407.8	-404.7	-403.4	-402.7
Band Fraction	-433.8	-409.9	-401.3	-400	-400	-400	-400

Figure 21 illustrates the different Hellinger distance of various d for both factor model and band fraction representation. Note that we do 50 iterations in the EM algorithm.

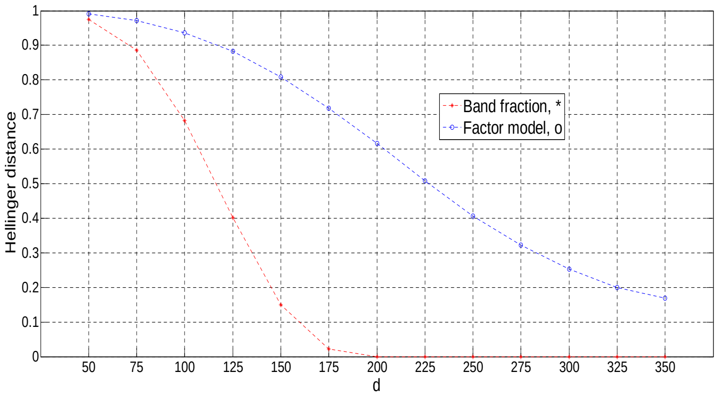

Figure 21: Hellinger distance comparison of various alpha

TABLE 14 shows the values of Figure 6. We can see that when d is greater than 200 the Hellinger distance between band fraction representation and covariance is 0. Because $d \geq p/2$.

Table 14: Hellinger distance Comparison (p=400)

Hellinger distance	$d = 50$	100	150	200	250	300	350
Factor Model	0.991	0.937	0.808	0.616	0.407	0.252	0.170
Band Fraction	0.974	0.683	0.149	0.00	0.00	0.00	0.00

Therefore, whenever it's small dimensional problems ($p = 9$) or large dimensional problems ($p = 400$), the band fraction representation is always better than the factor model in terms of both log-likelihood and the Hellinger distance. That's why we believe that the band fraction representation is a better structure for covariance matrices than factor models.

6.4.4 Portfolio Selection

Here, we do a comparison between factor models and band fraction representations in portfolio optimization. We use the same 400 members we get from the previous section and we choose $d = 250$. We consider a simple portfolio optimization problem with no transaction costs and boundaries:

$$\min_w \frac{1}{2} w^T \Sigma w - f^T w \Rightarrow w_{opt} = \Sigma^{-1} f^T$$

We do a 250 days back test from Aug-31-2012 to Aug-31-2013. First we calculate covariance matrices and their corresponding factor models $\Sigma_F = \Phi + \Lambda\Lambda^T$ and band fraction representations $\Sigma_B = (M^{-1}N)(M^{-1}N)^T$. Second, suppose we know tomorrow's returns and use those to compute the optimal weights. Then with the optimal weights we can compute the daily returns. We assume that sample covariance is the true covariance then compare factor model with $d = 100$ factors and band fraction with $d + 1$ band width. Figure 22 illustrates that if we keep all other parts the same except covariance estimation then the band fraction representation is better than factor models.

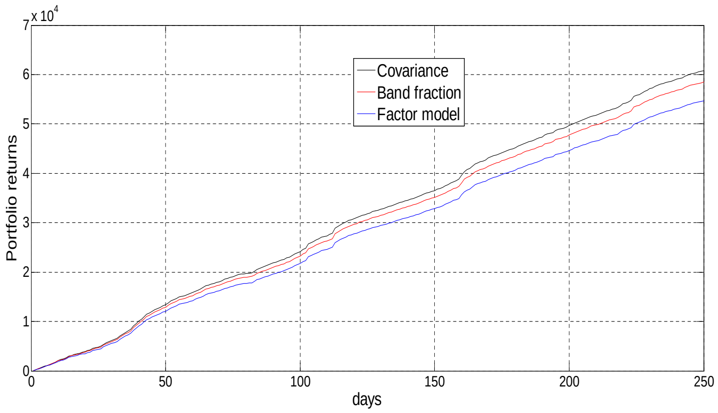

Figure 22: Portfolio returns comparison between two models and sample covariance

The total money we make with band fraction representation and factor model are 58493 and 54711 respectively. So band fraction representation makes 6.91% more than factor model in 250 days. Figure 23 illustrates that the band fraction representation is consistently better than the factor model.

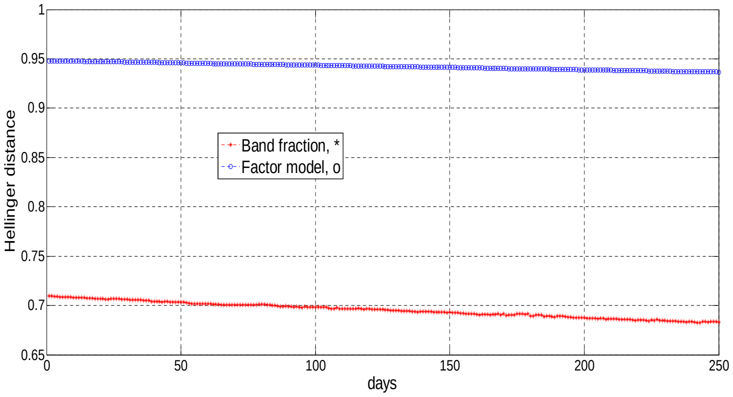

Figure 23: Hellinger distance comparison

6.5 Concluding Remarks

In this section, we introduced a new structure: band fraction representations for covariance matrix estimates. The band fraction representation has many good properties, such as the inverse of a band fraction representation is still a band fraction representation and the band fraction representations are in a convex set. Factor models are probably the most popular structure for covariance matrix estimates, but they don't have the above properties. More importantly, covariance matrices have those properties.

We compare band fraction representations and factor models in many ways, such as low dimension problems and high dimension problems with real financial data, and portfolio optimization. We use both log-likelihood and the Hellinger distance. Likelihood functions play a key role in statistical inference, especially methods of estimating a parameter from a set of statistics. Meanwhile the Hellinger distance is used to quantify the similarity between two probability distributions. Unlike the Frobenius Norm, the Hellinger distance is affine invariant. We have more confidence that two probability distributions are close when the Hellinger distance between them is small. In this section, we assume that all the data is from normal distributions. Because band fraction representations are consistently better than factor models, we show that in portfolio optimization if we set all other parts the same except covariance estimates then portfolio optimization using band fraction representations is better than using factor models. Factor models may over estimate the risk which will prevent you from getting more returns. That's what we find out through our experiments.

In band fraction representations when you choose $d \geq ceil(p/2)$ where d is the bandwidth and p is the dimension of the problem, the Hellinger distance between band fraction representation and covariance matrix is close to zero. But this is not true for factor models. In factor models you need to choose d closes to p, in order to make the Hellinger distance between factor model and covariance matrix close to zero.

The good thing about factor models is that they are easy to interpret and can be directly connected to financial sectors directly. People like to use financial sectors to estimate the stock market and each individual stock in the market. We like statistical factors better than financial sectors, in that case we believe that band fraction representation maybe a better choice. But we couldn't find a financial interpretation for those off-diagonal items. It would better if we could find some "factor loadings" in band fraction representation.

References

[1] Anderson, T. W. "Statistical inference for covariance matrices with linear structure." Multivariate Analysis, II (Proc. Second Internat. Sympos., Dayton, Ohio, 1968). 1969.

[2] Anderson, T. W. "Asymptotically efficient estimation of covariance matrices with linear structure." The Annals of Statistics 1.1 (1973): 135-141.

[3] Baek, Jangsun, Geoffrey J. McLachlan, and Lloyd K. Flack. "Mixtures of factor analyzers with common factor loadings: Applications to the clustering and visualization of high-dimensional data." Pattern Analysis and Machine Intelligence, IEEE Transactions on 32.7 (2010): 1298-1309.

[4] Bai, Jushan, and Shuzhong Shi. "Estimating high dimensional covariance matrices and its applications." (2011).

[5] Barnard, John, Robert McCulloch, and Xiao-Li Meng. "Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage." Statistica Sinica 10.4 (2000): 1281-1312.

[6] Beran, Rudolf. "Minimum Hellinger distance estimates for parametric models." The Annals of Statistics 5.3 (1977): 445-463.

[7] Bickel, Peter J., and Elizaveta Levina. "Regularized estimation of large covariance matrices." The Annals of Statistics (2008): 199-227.

[8] Bickel, Peter J., and Elizaveta Levina. "Covariance regularization by thresholding." The Annals of Statistics 36.6 (2008): 2577-2604.

[9] Bickel, Peter J., and Yulia R. Gel. "Banded regularization of autocovariance matrices in application to parameter estimation and forecasting of time series." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73.5 (2011): 711-728.

[10] Bilmes, Jeff A. "Factored sparse inverse covariance matrices." Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. Vol. 2. IEEE, 2000.

[11] Boik, Robert J. "Spectral models for covariance matrices." Biometrika 89.1 (2002): 159-182.

[12] Boudreaux-Bartels, G., and T. Parks. "Time-varying filtering and signal estimation using Wigner distribution synthesis techniques." Acoustics, Speech and Signal Processing, IEEE Transactions on 34.3 (1986): 442-451.

[13] Burg, John Parker, David G. Luenberger, and Daniel L. Wenger. "Estimation of structured covariance matrices." Proceedings of the IEEE 70.9 (1982): 963-974.

[14] Cai, T. Tony, Cun-Hui Zhang, and Harrison H. Zhou. "Optimal rates of convergence for covariance matrix estimation." The Annals of Statistics 38.4 (2010): 2118-2144.

[15] Chandrasekaran, Shiv, and Ming Gu. "Fast and stable algorithms for banded plus semiseparable systems of linear equations." SIAM Journal on Matrix Analysis and Applications 25.2 (2003): 373-384.

[16] Chandrasekaran, S., and M. Gu. "A divide-and-conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semiseparable matrices." Numerische Mathematik 96.4 (2004): 723-731.

[17] Chandrasekaran, S., M. Gu, and W. Lyons. A fast and stable adaptive solver for hierarchically semi-separable representations. Technical Report UCSB Math 2004-20, UC Santa Barbara, 2004.

[18] Chandrasekaran, Shiv, et al. "Some fast algorithms for sequentially semiseparable representations." SIAM Journal on Matrix Analysis and Applications 27.2 (2005): 341-364.

[19] Chandrasekaran, Shiv, Ming Gu, and T. Pals. "A fast ULV decomposition solver for hierarchically semiseparable representations." SIAM Journal on Matrix Analysis and Applications 28.3 (2006): 603-622.

[20] Chandrasekaran, Shiv, et al. "A superfast algorithm for Toeplitz systems of linear equations." SIAM Journal on Matrix Analysis and Applications 29.4 (2007): 1247-1266.

[21] Chang, Changgee, and Ruey S. Tsay. "Estimation of covariance matrix via the sparse Cholesky factor with lasso." Journal of Statistical Planning and Inference 140.12 (2010): 3858-3873.

[22] Chaudhuri, Sanjay, Mathias Drton, and Thomas S. Richardson. "Estimation of a covariance matrix with zeros." Biometrika 94.1 (2007): 199-216.

[23] Chen, Yilun, Ami Wiesel, and Alfred O. Hero. "Shrinkage estimation of high dimensional covariance matrices." Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on. IEEE, 2009.

[24] Chen, Yilun, et al. "Shrinkage algorithms for MMSE covariance estimation." Signal Processing, IEEE Transactions on 58.10 (2010): 5016-5029.

[25] Chen, Yilun, Ami Wiesel, and Alfred O. Hero. "Robust shrinkage estimation of high-dimensional covariance matrices." Signal Processing, IEEE Transactions on 59.9 (2011): 4097-4107.

[26] Chen, Yilun. Regularized Estimation of High-dimensional Covariance Matrices. Diss. Hebrew University of Jerusalem, 2011.

[27] Chen, Yilun, Ami Wiesel, and Alfred O. Hero. "Robust shrinkage estimation of high-dimensional covariance matrices." Signal Processing, IEEE Transactions on 59.9 (2011): 4097-4107.

[28] Chiu, Tom YM, Tom Leonard, and Kam-Wah Tsui. "The matrix-logarithmic covariance model." Journal of the American Statistical Association 91.433 (1996): 198-210.

[29] Christensen, Lars PB. "An EM-algorithm for band-toeplitz covariance matrix estimation." Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. Vol. 3. IEEE, 2007.

[30] Chu, Moody T., Robert E. Funderlic, and Robert J. Plemmons. "Structured low rank approximation." Linear algebra and its applications 366 (2003): 157-172.

[31] Curran, Patrick J., Stephen G. West, and John F. Finch. "The robustness of test statistics to nonnormality and specification error in confirmatory factor analysis." Psychological methods 1.1 (1996): 16.

[32] Daniels, Michael J., and Robert E. Kass. "Shrinkage estimators for covariance matrices." Biometrics 57.4 (2001): 1173-1184.

[33] Daniels, M. J., and M. Pourahmadi. "Modeling covariance matrices via partial autocorrelations." Journal of multivariate analysis 100.10 (2009): 2352-2363.

[34] Delvaux, Steven, and Marc Van Barel. "Structures preserved by Schur complementation." SIAM journal on matrix analysis and applications 28.1 (2006): 229-252.

[35] Delvaux, Steven, and Marc Van Barel. "A Givens-weight representation for rank structured matrices." SIAM Journal on Matrix Analysis and Applications 29.4 (2007): 1147-1170.

[36] Dembo, A. "The relation between maximum likelihood estimation of structured covariance matrices and periodograms." Acoustics, Speech and Signal Processing, IEEE Transactions on 34.6 (1986): 1661-1662.

[37] Dembo, Amir, Colin L. Mallows, and Lawrence A. Shepp. "Embedding nonnegative definite Toeplitz matrices in nonnegative definite circulant matrices, with application to covariance estimation." Information Theory, IEEE Transactions on 35.6 (1989): 1206-1212.

[38] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the Royal Statistical Society. Series B (Methodological) (1977): 1-38.

[39] Deng, Xinwei, and Ming Yuan. "Large Gaussian covariance matrix estimation with Markov structures." Journal of Computational and Graphical Statistics 18.3 (2009).

[40] Dey, Dipak K., and C. Srinivasan. "Estimation of a covariance matrix under Stein's loss." The Annals of Statistics (1985): 1581-1591.

[41] Eckart, Carl, and Gale Young. "The approximation of one matrix by another of lower rank." Psychometrika 1.3 (1936): 211-218.

[42] Eidelman, Y., and I. Gohberg. "Fast inversion algorithms for diagonal plus semiseparable matrices." Integral Equations and Operator Theory 27.2 (1997): 165-183.

[43] Fama, Eugene F., and Kenneth R. French. "Common risk factors in the returns on stocks and bonds." Journal of financial economics 33.1 (1993): 3-56.

[44] Fan, Jianqing, Yingying Fan, and Jinchi Lv. "High dimensional covariance matrix estimation using a factor model." Journal of Econometrics 147.1 (2008): 186-197.

[45] Fasino, Dario, Nicola Mastronardi, and Marc Van Barel. "Fast and stable algorithms for reducing diagonal plus semiseparable matrices to tridiagonal and bidiagonal form." Contemporary Mathematics 323 (2003): 105-118.

[46] Firth, David. "Bias reduction of maximum likelihood estimates." Biometrika 80.1 (1993): 27-38.

[47] Forni, Mario, et al. "The generalized dynamic-factor model: Identification and estimation." Review of Economics and statistics 82.4 (2000): 540-554.

[48] Frieze, Alan, Ravi Kannan, and Santosh Vempala. "Fast Monte-Carlo algorithms for finding low-rank approximations." Journal of the ACM (JACM) 51.6 (2004): 1025-1041.

[49] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. "Sparse inverse covariance estimation with the graphical lasso." Biostatistics 9.3 (2008): 432-441.

[50] Furrer, Reinhard, and Thomas Bengtsson. "Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants." Journal of Multivariate Analysis 98.2 (2007): 227-255.

[51] Ghahramani, Zoubin, and Geoffrey E. Hinton. The EM algorithm for mixtures of factor analyzers. Vol. 60. Technical Report CRG-TR-96-1, University of Toronto, 1996.

[52] Gilbert, Jean Charles, and Jorge Nocedal. "Global convergence properties of conjugate gradient methods for optimization." SIAM Journal on Optimization 2.1 (1992): 21-42.

[53] Gohberg, I., T. Kailath, and I. Koltracht. "Linear complexity algorithms for semiseparable matrices." Integral Equations and Operator Theory 8.6 (1985): 780-804.

[54] Gray, R. M. (2006). Toeplitz and circulant matrices: A review. Now Pub.

[55] Gu, Ming, Xiaoye S. Li, and Panayot S. Vassilevski. "Direction-preserving and Schur-monotonic semiseparable approximations of symmetric positive definite

matrices." SIAM Journal on Matrix Analysis and Applications 31.5 (2010): 2650-2664.

[56] Haff, L. R. "Empirical Bayes estimation of the multivariate normal covariance matrix." The Annals of Statistics 8.3 (1980): 586-597.

[57] Hoff, Peter D., and Xiaoyue Niu. "A covariance regression model." arXiv preprint arXiv:1102.5721 (2011).

[58] Huang, Jianhua Z., et al. "Covariance matrix selection and estimation via penalised normal likelihood." Biometrika 93.1 (2006): 85-98.

[59] Jamshidian, Mortaza, and Robert I. Jennrich. "Conjugate gradient acceleration of the EM algorithm." Journal of the American Statistical Association 88.421 (1993): 221-228.

[60] Jamshidian, Mortaza, and Robert I. Jennrich. "Conjugate gradient methods in confirmatory factor analysis." Computational statistics & data analysis 17.3 (1994): 247-263.

[61] Jamshidian, Mortaza, and Robert I. Jennrich. "Acceleration of the EM Algorithm by using Quasi-Newton Methods." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 59.3 (1997): 569-587.

[62] Jansson, Magnus, and Bjorn Ottersten. "Structured covariance matrix estimation: a parametric approach." Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. Vol. 5. IEEE, 2000.

[63] Jones, Christopher S. "Extracting factors from heteroskedastic asset returns." Journal of Financial economics 62.2 (2001): 293-325.

[64] Jöreskog, Karl G. "A general approach to confirmatory maximum likelihood factor analysis." Psychometrika 34.2 (1969): 183-202.

[65] Karlis, Dimitris. "An EM type algorithm for maximum likelihood estimation of the normal–inverse Gaussian distribution." Statistics & probability letters 57.1 (2002): 43-52.

[66] Kaufman, Cari G., Mark J. Schervish, and Douglas W. Nychka. "Covariance tapering for likelihood-based estimation in large spatial data sets." Journal of the American Statistical Association 103.484 (2008).

[67] Lam, Clifford, and Jianqing Fan. "Sparsistency and rates of convergence in large covariance matrix estimation." Annals of statistics 37.6B (2009): 4254.

[68] Lawley, Derrick N. "The estimation of factor loadings by the method of maximum likelihood." Proceedings of the Royal Society of Edinburgh 60.2 (1940): 64-82.

[69] Ledoit, Olivier, and Michael Wolf. "Improved estimation of the covariance matrix of stock returns with an application to portfolio selection." Journal of Empirical Finance 10.5 (2003): 603-621.

[70] Ledoit, Olivier, and Michael Wolf. "A well-conditioned estimator for large-dimensional covariance matrices." Journal of multivariate analysis 88.2 (2004): 365-411.

[71] Ledoit, Olivier, and Michael Wolf. "Nonlinear shrinkage estimation of large-dimensional covariance matrices." The Annals of Statistics 40.2 (2012): 1024-1060.

[72] Liu, Chuanhai, and Donald B. Rubin. "The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence." Biometrika 81.4 (1994): 633-648.

[73] Liu, Chuanhai, and Donald B. Rubin. "ML estimation of the t distribution using EM and its extensions, ECM and ECME." Statistica Sinica 5.1 (1995): 19-39.

[74] Liu, Chuanhai, and Donald B. Rubin. "Maximum likelihood estimation of factor analysis using the ECME algorithm with complete and incomplete data." Statistica Sinica 8.3 (1998): 729-747.

[75] Markovsky, Ivan. "Structured low-rank approximation and its applications." Automatica 44.4 (2008): 891-909.

[76] Maronna, Ricardo Antonio. "Robust M-estimators of multivariate location and scatter." The annals of statistics (1976): 51-67.

[77] Mastronardi, Nicola, Shivkumar Chandrasekaran, and Sabine Van Huffel. "Fast and stable two-way algorithm for diagonal plus semi-separable systems of linear equations." Numerical linear algebra with applications 8.1 (2001): 7-12.

[78] Matsuyama, Yasuo. "Non-logarithmic information measures, α-weighted EM algorithms and speedup of learning." Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on. IEEE, 1998.

[79] Matsuyama, Yasuo. "The α-EM algorithm and its basic properties." Systems and Computers in Japan 31.11 (2000): 12-23.

[80] Matsuyama, Yasuo., et al. "α-EM algorithm and ■-ICA learning based upon extended logarithmic information measures." Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on. Vol. 3. IEEE, 2000.

[81] Matsuyama, Yasuo. "The α-EM algorithm: Surrogate likelihood maximization using ■-logarithmic information measures." Information Theory, IEEE Transactions on 49.3 (2003): 692-706.

[82] Matsuyama, Yasuo, and Ryunosuke Hayashi. "Alpha-EM gives fast hidden Markov model estimation: Derivation and evaluation of alpha-HMM." Neural Networks (IJCNN), The 2010 International Joint Conference on. IEEE, 2010.

[83] Matsuyama, Yasuo. "Hidden Markov model estimation based on alpha-EM algorithm: Discrete and continuous alpha-HMMs." Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 2011.

[84] Matsuyama, Yasuo, Ryunosuke Hayashi, and Ryota Yokote. "Fast estimation of Hidden Markov Models via alpha-EM algorithm." Statistical Signal Processing Workshop (SSP), 2011 IEEE. IEEE, 2011.

[85] McLachlan, Geoffrey J., David Peel, and R. W. Bean. "Modelling high-dimensional data by mixtures of factor analyzers." Computational Statistics & Data Analysis 41.3 (2003): 379-388.

[86] McLachlan, Geoffrey J., R. W. Bean, and L. Ben-Tovim Jones. "Extension of the mixture of factor analyzers model to incorporate the multivariate *t*-distribution." Computational Statistics & Data Analysis 51.11 (2007): 5327-5338.

[87] McMurry, Timothy L., and Dimitris N. Politis. "Banded and tapered estimates for autocovariance matrices and the linear process bootstrap." Journal of Time Series Analysis 31.6 (2010): 471-482.

[88] Meng, Xiao-Li, and Donald B. Rubin. "Maximum likelihood estimation via the ECM algorithm: A general framework." Biometrika 80.2 (1993): 267-278.

[89] Meng, X-L., and David Van Dyk. "Fast EM-type implementations for mixed effects models." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 60.3 (1998): 559-578.

[90] Miller, Michael I., and Donald L. Snyder. "The role of likelihood and entropy in incomplete-data problems: applications to estimating point-process intensities and Toeplitz constrained covariances." Proceedings of the IEEE 75.7 (1987): 892-907.

[91] Mullhaupt, Andrew P., and Kurt S. Riedel. "Fast adaptive identification of stable innovation filters." Signal Processing, IEEE Transactions on 45.10 (1997): 2616-2619.

[92] Mullhaupt, Andrew P., and Kurt S. Riedel. "Low grade matrices and matrix fraction representations." Linear algebra and its applications 342.1 (2002): 187-201.

[93] Mullhaupt, Andrew P., and Kurt S. Riedel. "Banded matrix fraction representation of triangular input normal pairs." Automatic Control, IEEE Transactions on 46.12 (2001): 2018-2022.

[94] Mullhaupt, Andrew P. Personal communication.

[95] Newey, Whitney K., and Kenneth D. West. "A simple, positive semi-definite, heteroskedasticity and autocorrelationconsistent covariance matrix." (1986).

[96] Nguyen, A. "On the uniqueness of the maximum-likeliwood estimate of structured covariance matrices." Acoustics, Speech and Signal Processing, IEEE Transactions on 32.6 (1984): 1249-1251.

[97] Patra, Rohit Kumar, Abhijit Mandal, and Ayanendranath Basu. "Minimum Hellinger distance estimation with inlier modification." Sankhyā: The Indian Journal of Statistics, Series B (2008-) (2008): 310-322.

[98] Pourahmadi, Mohsen. "Maximum likelihood estimation of generalised linear models for multivariate normal covariance matrix." Biometrika 87.2 (2000): 425-435.

[99] Pourahmadi, Mohsen. "Cholesky decompositions and estimation of a covariance matrix: orthogonality of variance–correlation parameters." Biometrika 94.4 (2007): 1006-1013.

[100] Pourahmadi, Mohsen. "Simultaneous modeling of covariance matrices: GLM, Bayesian and nonparametric perspective." Correlated Data Modelling 2004 (2007): 41-64.

[101] Pourahmadi, Mohsen. "Covariance estimation: The GLM and regularization perspectives." Statistical Science 26.3 (2011): 369-387.

[102] Rajaratnam, Bala, Hélene Massam, and Carlos M. Carvalho. "Flexible covariance estimation in graphical Gaussian models." The Annals of Statistics 36.6 (2008): 2818-2849.

[103] Rothman, Adam J., et al. "Sparse permutation invariant covariance estimation." Electronic Journal of Statistics 2 (2008): 494-515.

[104] Rothman, Adam J., Elizaveta Levina, and Ji Zhu. "Generalized thresholding of large covariance matrices." Journal of the American Statistical Association 104.485 (2009): 177-186.

[105] Rothman, Adam J., Elizaveta Levina, and Ji Zhu. "A new approach to Cholesky-based covariance regularization in high dimensions." Biometrika 97.3 (2010): 539-550.

[106] Roweis, Sam. "EM algorithms for PCA and SPCA." Advances in neural information processing systems (1998): 626-632.

[107] Rubin, Donald B., and Dorothy T. Thayer. "EM algorithms for ML factor analysis." Psychometrika 47.1 (1982): 69-76.

[108] Rubin, Donald B., and Ted H. Szatrowski. "Finding maximum likelihood estimates of patterned covariance matrices by the EM algorithm." Biometrika 69.3 (1982): 657-660.

[109] Salakhutdinov, Ruslan, Sam Roweis, and Zoubin Ghahramani. "Relationship between gradient and EM steps in latent variable models." (2004): 261.

[110] Salakhutdinov, Ruslan, Sam Roweis, and Zoubin Ghahramani. "Optimization with EM and expectation-conjugate-gradient." ICML. 2003.

[111] Simpson, Douglas G. "Minimum Hellinger distance estimation for the analysis of count data." Journal of the American statistical Association 82.399 (1987): 802-807.

[112] Tamura, Roy N., and Dennis D. Boos. "Minimum Hellinger distance estimation for multivariate location and covariance." Journal of the American Statistical Association 81.393 (1986): 223-229.

[113] Turmon, Michael J., and Michael I. Miller. "Maximum-likelihood estimation of complex sinusoids and Toeplitz covariances." Signal Processing, IEEE Transactions on 42.5 (1994): 1074-1086.

[114] Tyler, David E. "A distribution-free $ M $-estimator of multivariate scatter." The Annals of Statistics 15.1 (1987): 234-251.

[115] Ueda, Naonori, et al. "SMEM algorithm for mixture models." Neural computation 12.9 (2000): 2109-2128.

[116] Van Camp, Ellen, Nicola Mastronardi, and Marc Van Barel. "Two fast algorithms for solving diagonal-plus-semiseparable linear systems." Journal of Computational and Applied Mathematics 164 (2004): 731-747.

[117] Vandebril, Raf, Marc Van Barel, and Nicola Mastronardi. Matrix computations and semiseparable matrices: linear systems. Vol. 1. JHU Press, 2007.

[118] Vandebril, Raf, Marc Van Barel, and Nicola Mastronardi. Matrix computations and semiseparable matrices: eigenvalue and singular value methods. Vol. 2. JHU Press, 2008.

[119] van der Veen, A. J. "Approximate inversion of a large semiseparable positive matrix." Proc. 17th Int. Symp. on Mathematical Theory of Networks and Systems (MTNS-04), Brussels (BE). 2004.

[120] Williams, Douglas B., and Don H. Johnson. "Robust estimation of structured covariance matrices." Signal Processing, IEEE Transactions on 41.9 (1993): 2891-2906.

[121] Wirfalt, P., and Magnus Jansson. "On Toeplitz and Kronecker structured covariance matrix estimation." Sensor Array and Multichannel Signal Processing Workshop (SAM), 2010 IEEE. IEEE, 2010.

[122] Witten, Daniela M., and Robert Tibshirani. "Covariance-regularized regression and classification for high dimensional problems." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71.3 (2009): 615-636.

[123] Wu, Wei Biao, and Mohsen Pourahmadi. "Banding sample autocovariance matrices of stationary processes." Statistica Sinica 19.4 (2009): 1755.

[124] Wu, Wei Biao, Yinxiao Huang, and Wei Zheng. "Covariances estimation for long-memory processes." Advances in Applied Probability 42.1 (2010): 137-157.

[125] Wu, Wei Biao, and Han Xiao. "Covariance Matrix Estimation in Time Series." Handbook of Statistics, Vol. 30: Time Series Analysis: Methods and Applications (2012): 187-209.

[126] Yang, Ruoyong, and James O. Berger. "Estimation of a covariance matrix using the reference prior." The Annals of Statistics (1994): 1195-1211.

[127] Yuan, Ming, and Yi Lin. "Model selection and estimation in the Gaussian graphical model." Biometrika 94.1 (2007): 19-35.

[128] Zhao, J-H., L. H. Philip, and Qibao Jiang. "ML estimation for factor analysis: EM or non-EM?." Statistics and computing 18.2 (2008): 109-123.

[129] Zhao, Jian-Hua, and Philip LH Yu. "Fast ML estimation for the mixture of factor analyzers via an ECM algorithm." Neural Networks, IEEE Transactions on 19.11 (2008): 1956-1961.

A Appendix: Conditional distribution of factors

For the distribution of the observed data $p(x)$

$$\begin{aligned} E[X] &= 0 \\ Cov(X) &= \Sigma = \Phi + \Lambda\Lambda^T \end{aligned}$$

For the complete data distribution $p(x, z)$

$$\begin{aligned} Y &= \begin{bmatrix} X \\ Z \end{bmatrix} \\ E[Y] &= E \begin{bmatrix} X \\ Z \end{bmatrix} = 0 \end{aligned}$$

$$\begin{aligned} Cov(Y) &= E[YY^T] = E \left[\begin{bmatrix} X \\ Z \end{bmatrix} \begin{bmatrix} X^T & Z^T \end{bmatrix} \right] \\ &= E \begin{bmatrix} XX^T & XZ^T \\ ZX^T & ZZ^T \end{bmatrix} = \begin{bmatrix} \Phi + \Lambda\Lambda^T & \Lambda \\ \Lambda^T & I \end{bmatrix} = \Delta \end{aligned}$$

Also, since

$$\Delta = \begin{bmatrix} \Phi + \Lambda\Lambda^T & \Lambda \\ \Lambda^T & I \end{bmatrix} = \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix}$$

we have (Partitioned Matrix Inversion Theorem)

$$\begin{aligned} \Delta^{-1} &= \begin{bmatrix} \Delta^{-1,11} & \Delta^{-1,12} \\ \Delta^{-1,21} & \Delta^{-1,22} \end{bmatrix} \\ &= \begin{bmatrix} (\Delta_{11} - \Delta_{12}\Delta_{22}^{-1}\Delta_{21})^{-1} & \Delta_{11}^{-1}\Delta_{12}(\Delta_{21}\Delta_{11}^{-1}\Delta_{12} - \Delta_{22})^{-1} \\ (\Delta_{21}\Delta_{11}^{-1}\Delta_{12} - \Delta_{22})^{-1}\Delta_{21}\Delta_{11}^{-1} & (\Delta_{22} - \Delta_{21}\Delta_{11}^{-1}\Delta_{12})^{-1} \end{bmatrix} \end{aligned}$$

For the complete data distribution $p(z|x)$

$$\begin{aligned}
p(z|x) &= \frac{p(x,z)}{p(x)} = \frac{(2\pi)^{-(p+d)/2}\,|\Delta|^{-1/2}\exp\left(-\frac{1}{2}y^T\Delta^{-1}y\right)}{(2\pi)^{-p/2}\,|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)} \\
&= (blah)\exp\left(-\frac{1}{2}(y^T\Delta^{-1}y - x^T\Sigma^{-1}x)\right) \\
&= (blah)\exp\left(-\frac{1}{2}\alpha\right)
\end{aligned}$$

where we define

$$\begin{aligned}
\alpha &= y^T\Delta^{-1}y - x^T\Sigma^{-1}x \\
&= \left[\begin{array}{cc} x^T & z^T \end{array}\right]\Delta^{-1}\left[\begin{array}{c} x \\ z \end{array}\right] - x^T\Sigma^{-1}x \\
&= x^T\Delta^{-1,11}x + x^T\Delta^{-1,12}z + z^T\Delta^{-1,21}x + z^T\Delta^{-1,22}z - x^T\Sigma^{-1}x \\
&= x^T\left(\Delta^{-1,11} - \Sigma^{-1}\right)x + 2x^T\Delta^{-1,12}z + z^T\Delta^{-1,22}z
\end{aligned}$$

Consider the term

$$\begin{aligned}
\left(\Delta^{-1,11} - \Sigma^{-1}\right) &= (\Delta_{11} - \Delta_{12}\Delta_{22}^{-1}\Delta_{21}) - \Delta_{11}^{-1} \\
&= \Delta_{11}^{-1} + \Delta_{11}^{-1}\Delta_{12}(\Delta_{22} - \Delta_{21}\Delta_{11}^{-1}\Delta_{12})\Delta_{21}\Delta_{11}^{-1} - \Delta_{11}^{-1} \\
&= \Delta_{11}^{-1}\Delta_{12}(\Delta_{22} - \Delta_{21}\Delta_{11}^{-1}\Delta_{12})\Delta_{21}\Delta_{11}^{-1} \\
&= \beta^T\Delta^{-1,22}\beta
\end{aligned}$$

where we define

$$\begin{aligned}
\beta &= \Delta_{21}\Delta_{11}^{-1} \\
&= \Lambda^T(\Phi + \Lambda\Lambda^T)^{-1}
\end{aligned}$$

So

$$\begin{aligned}
\alpha &= x^T\beta^T\Delta^{-1,22}\beta x + 2x^T\Delta^{-1,12}z + z^T\Delta^{-1,22}z \\
&= (z - \beta x)^T\Delta^{-1,22}(z - \beta x) + 2x^T(\beta^T\Delta^{-1,22} + \Delta^{-1,12})z
\end{aligned}$$

and

$$\begin{aligned}\beta^T\Delta^{-1,22}+\Delta^{-1,12} &= \Delta_{11}^{-1}\Delta_{12}(\Delta_{22}-\Delta_{21}\Delta_{11}^{-1}\Delta_{12})^{-1}+\Delta_{11}^{-1}\Delta_{12}(\Delta_{21}\Delta_{11}^{-1}\Delta_{12}-\Delta_{22})^{-1} \\ &= 0\end{aligned}$$

Hence

$$\alpha = (z-\beta x)^T\Delta^{-1,22}(z-\beta x)$$

Therefore

$$p(z|x) = (blah)\exp\left(-\frac{1}{2}(z-\beta x)^T\Delta^{-1,22}(z-\beta x)\right)$$

from which we can deduce

$$\begin{aligned}E[z|x] &= \beta x \\ Cov(z|x) &= (\Delta^{-1,22})^{-1}\end{aligned}$$

and (Matrix Inversion Theorem)

$$\begin{aligned}[Cov(z|x)]^{-1} &= \Delta^{-1,22} \\ &= (\Delta_{22}-\Delta_{21}\Delta_{11}^{-1}\Delta_{12})^{-1} \\ &= \Delta_{22}^{-1}-\Delta_{22}^{-1}\Delta_{21}(-\Delta_{11}+\Delta_{12}\Delta_{22}^{-1}\Delta_{21})^{-1}\Delta_{12}\Delta_{22}^{-1} \\ &= I-\Lambda^T(-\Phi-\Lambda\Lambda^T+\Lambda\Lambda^T)^{-1}\Lambda \\ &= I+\Lambda^T\Phi^{-1}\Lambda\end{aligned}$$

Then we have

$$\begin{aligned}E[zz^T|x] &= Cov(z|x)+E[z|x]E[z|x]^T \\ &= (I+\Lambda^T\Phi^{-1}\Lambda)^{-1}+\beta xx^T\beta^T \\ &= I-\Lambda^T(\Phi+\Lambda\Lambda^T)^{-1}\Lambda+\beta xx^T\beta^T \\ &= I-\beta\Lambda+\beta xx^T\beta^T\end{aligned}$$

B Appendix: Update equation of factor analysis

B.1 Non-Causal

From equation (6) we have

$$\begin{aligned}
&\Rightarrow \sum_{j=1}^{N} \frac{\frac{1+\alpha}{2}\int P^{\frac{1+\alpha}{2}} * \left(\Phi_1^{-1}x_iz_i' - \Phi_1^{-1}\Lambda_1 z_iz_i'\right) * p(z_i)dz_i}{\int P^{\frac{1+\alpha}{2}} * p(z_i)dz_i} = 0 \\
&\Rightarrow \sum_{j=1}^{N} \frac{\frac{1+\alpha}{2} * blah * \int e^{-\frac{1}{2}(z_i-\Sigma A')'\Sigma^{-1}(z_i-\Sigma A')} * \left(\Phi_1^{-1}x_iz_i' - \Phi_1^{-1}\Lambda_1 z_iz_i'\right)dz_i}{blah * \int e^{-\frac{1}{2}(z_i-\Sigma A')'\Sigma^{-1}(z_i-\Sigma A')}dz_i} = 0 \\
&\Rightarrow \sum_{j=1}^{N} \Phi_1^{-1}x_iE[z_i'] - \Phi_1^{-1}\Lambda_1 E[z_iz_i'] = 0 \\
&\Rightarrow \Lambda_1 = \sum_{j=1}^{N} x_iE[z_i'] \left(\sum_{j=1}^{N} E[z_iz_i']\right)^{-1}
\end{aligned}$$

where

$$\begin{aligned}
E[z_i] &= \Sigma A' \Rightarrow E[z_i'] = A\Sigma \\
E[z_iz_i'] &= Var[z_i] + E[z_i]E[z_i]' = \Sigma + \Sigma A'A\Sigma
\end{aligned}$$

Then

$$\begin{aligned}
\Lambda_1 &= \sum_{j=1}^{N} x_iA\Sigma \left(\sum_{j=1}^{N} \Sigma + \Sigma A'A\Sigma\right)^{-1} = \sum_{j=1}^{N} x_ix_i'W\Sigma \left(\sum_{j=1}^{N} \Sigma + \Sigma W'x_ix_i'W\Sigma\right)^{-1} \\
C_{xx} &= \sum_{j=1}^{N} \frac{x_ix_i'}{N} \\
\Lambda_1 &= C_{xx}W\Sigma\left(\Sigma + \Sigma W'C_{xx}W\Sigma\right)^{-1}
\end{aligned}$$

From equation (7) we have

$$
\begin{aligned}
&\Rightarrow \quad \sum_{j=1}^{N} \frac{\frac{1}{2}\frac{1+\alpha}{2}\int P^{\frac{1+\alpha}{2}} * (\Phi_1 - x_i x_i' + x_i z_i' \Lambda_1' + \Lambda_1 z_i x_i' - \Lambda_1 z_i z_i' \Lambda_1') * p(z_i) dz_i}{\int P^{\frac{1+\alpha}{2}} * p(z_i) dz_i} = 0 \\
&\Rightarrow \quad \sum_{j=1}^{N} \frac{blah * \int e^{-\frac{1}{2}(z_i - \Sigma A')'\Sigma^{-1}(z_i - \Sigma A')} * (\Phi_1 - x_i x_i' + x_i z_i' \Lambda_1' + \Lambda_1 z_i x_i' - \Lambda_1 z_i z_i' \Lambda_1')\, dz_i}{blah * \int e^{-\frac{1}{2}(z_i - \Sigma A')'\Sigma^{-1}(z_i - \Sigma A')} dz_i} = 0 \\
&\Rightarrow \quad \sum_{j=1}^{N} \Phi_1 - x_i x_i' + x_i E[z_i'] \Lambda_1' + \Lambda_1 E[z_i] x_i' - \Lambda_1 E[z_i z_i'] \Lambda_1' = 0 \\
&\Rightarrow \quad \sum_{j=1}^{N} \Phi_1 - \sum_{j=1}^{N} x_i x_i' + \sum_{j=1}^{N} x_i E[z_i'] \Lambda_1' + \sum_{j=1}^{N} \Lambda_1 E[z_i] x_i' - \Lambda_1 \sum_{j=1}^{N} E[z_i z_i'] \Lambda_1' = 0
\end{aligned}
$$

we know that

$$
\begin{aligned}
\Lambda_1 &= \sum_{j=1}^{N} x_i E[z_i'] \left(\sum_{j=1}^{N} E[z_i z_i'] \right)^{-1} \\
\Lambda_1' &= \left(\sum_{j=1}^{N} E[z_i z_i'] \right)^{-1} \sum_{j=1}^{N} E[z_i] x_i'
\end{aligned}
$$

so we get

$$
\begin{aligned}
&\Rightarrow \quad \sum_{j=1}^{N} \Phi_1 - \sum_{j=1}^{N} x_i x_i' + \sum_{j=1}^{N} x_i E[z_i'] \Lambda_1' + \sum_{j=1}^{N} \Lambda_1 E[z_i] x_i' - \Lambda_1 \sum_{j=1}^{N} E[z_i z_i'] \left(\sum_{j=1}^{N} E[z_i z_i'] \right)^{-1} \sum_{j=1}^{N} E[z_i] x_i' = 0 \\
&\Rightarrow \quad \sum_{j=1}^{N} \Phi_1 - \sum_{j=1}^{N} x_i x_i' + \sum_{j=1}^{N} x_i E[z_i'] \Lambda_1' + \sum_{j=1}^{N} \Lambda_1 E[z_i] x_i' - \Lambda_1 \sum_{j=1}^{N} E[z_i] x_i' = 0 \\
&\Rightarrow \quad \sum_{j=1}^{N} \Phi_1 - \sum_{j=1}^{N} x_i x_i' + \sum_{j=1}^{N} x_i E[z_i'] \Lambda_1' = 0 \\
&\Rightarrow \quad \Phi_1 = diag(C_{xx-} C_{xx} W \Sigma \Lambda_1')
\end{aligned}
$$

When $-1 < \alpha < 1$

$$
\begin{aligned}
\Sigma^{-1} &= \frac{1+\alpha}{2} \Lambda_1' \Phi_1^{-1} \Lambda_1 - \frac{1+\alpha}{2} \Lambda_0' \Phi_0^{-1} \Lambda_0 + C^{-1} \\
A &= \frac{1+\alpha}{2} x_i' \Phi_1^{-1} \Lambda_1 - \frac{1+\alpha}{2} x_i' \Phi_0^{-1} \Lambda_0 + x_i' \beta' C^{-1} = x_i' W \\
W &= \frac{1+\alpha}{2} \Phi_1^{-1} \Lambda_1 - \frac{1+\alpha}{2} \Phi_0^{-1} \Lambda_0 + \beta' C^{-1}
\end{aligned}
$$

and

$$\begin{aligned}\beta &= \Lambda_0'(\Phi_0+\Lambda_0\Lambda_0')^{-1}\\ C &= I-\Lambda_0'(\Phi_0+\Lambda_0\Lambda_0')^{-1}\Lambda_0\end{aligned}$$

then

$$\begin{aligned}\beta' C^{-1} &= (\Phi_0+\Lambda_0\Lambda_0')^{-1}\Lambda_1 * (I+\Lambda_0'\Phi_0^{-1}\Lambda_0)\\ &= (\Phi_0+\Lambda_0\Lambda_0')^{-1}\Lambda_1 * (\Lambda_0^{-1}\Phi_0\Phi_0^{-1}\Lambda_0+\Lambda_0'\Phi_0^{-1}\Lambda_0)\\ &= (\Phi_0+\Lambda_0\Lambda_0')^{-1}\Lambda_1 * (\Lambda_0^{-1}\Phi_0+\Lambda_0') * \Phi_0^{-1}\Lambda_0\\ &= (\Phi_0+\Lambda_0\Lambda_0')^{-1}(\Phi_0+\Lambda_0\Lambda_0') * \Phi_0^{-1}\Lambda_0\\ &= \Phi_0^{-1}\Lambda_0\end{aligned}$$

so we get

$$\begin{aligned}\Sigma^{-1} &= I+\frac{1+\alpha}{2}\Lambda_1'\Phi_1^{-1}\Lambda_1+\frac{1-\alpha}{2}\Lambda_0'\Phi_0^{-1}\Lambda_0\\ W &= \frac{1+\alpha}{2}\Phi_1^{-1}\Lambda_1+\frac{1-\alpha}{2}\Phi_0^{-1}\Lambda_0\end{aligned}$$

Set

$$\Phi^{-1}=\begin{bmatrix}\frac{1+\alpha}{2}\Phi_1^{-1} & \\ & \frac{1-\alpha}{2}\Phi_0^{-1}\end{bmatrix},\ \Phi=\begin{bmatrix}\frac{2}{1+\alpha}\Phi_1 & \\ & \frac{2}{1-\alpha}\Phi_0\end{bmatrix},\ \Lambda=\begin{bmatrix}\Lambda_1\\ \Lambda_0\end{bmatrix}$$

we have

$$\begin{aligned}W &= \begin{bmatrix}I_n\\ I_n\end{bmatrix}'\Phi^{-1}\Lambda\\ \Sigma &= I_d-\left(I_d+\Lambda'\Phi^{-1}\Lambda\right)^{-1}\Lambda'\Phi^{-1}\Lambda\end{aligned}$$

$$\begin{aligned}W\Sigma &= \begin{bmatrix}I_n\\ I_n\end{bmatrix}'\Phi^{-1}\Lambda\left(I_d-\left(I_d+\Lambda'\Phi^{-1}\Lambda\right)^{-1}\Lambda'\Phi^{-1}\Lambda\right)\\ &= \begin{bmatrix}I_n\\ I_n\end{bmatrix}'\Phi^{-1}\Lambda-\begin{bmatrix}I_n\\ I_n\end{bmatrix}'\Phi^{-1}\Lambda\left(I_d+\Lambda'\Phi^{-1}\Lambda\right)^{-1}\Lambda'\Phi^{-1}\Lambda\\ &= \begin{bmatrix}I_n\\ I_n\end{bmatrix}'\left(\Phi^{-1}-\Phi^{-1}\Lambda\left(I_d+\Lambda'\Phi^{-1}\Lambda\right)^{-1}\Lambda'\Phi^{-1}\right)\Lambda\\ &= \begin{bmatrix}I_n\\ I_n\end{bmatrix}'\left(\Phi+\Lambda\Lambda'\right)^{-1}\Lambda\end{aligned}$$

If we assume

$$\begin{aligned}\delta &= \begin{bmatrix} I_n \\ I_n \end{bmatrix}' \left(\Phi + \Lambda\Lambda'\right)^{-1} \Lambda \\ \Delta &= I_d - \Lambda'(\Phi + \Lambda\Lambda')^{-1}\Lambda\end{aligned}$$

we get

$$\begin{aligned}\Lambda_1 &= C_{xx}\delta\left(\Delta + \delta' C_{xx}\delta\right)^{-1} \\ \Phi_1 &= diag(C_{xx-}C_{xx}\delta\Lambda_1')\end{aligned}$$

B.2 Causal

From equation (6) we have

$$\begin{aligned}
&\Rightarrow \sum_{i=1}^{N} \frac{\frac{1+\alpha}{2}\int P^{\frac{1+\alpha}{2}} * \left(\Phi_1^{-1}x_i z_i' - \Phi_1^{-1}\Lambda_1 z_i z_i'\right) * p(z_i) dz_i}{\int P^{\frac{1+\alpha}{2}} * p(z_i) dz_i} = 0 \\
&\Rightarrow \sum_{i=1}^{N} \frac{\int \left(\frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1},\Phi_{-1}) + \frac{1+\beta}{2}P(z_i|x_i,\Lambda_0,\Phi_0)\right)\left(\Phi_1^{-1}x_i z_i' - \Phi_1^{-1}\Lambda_1 z_i z_i'\right) dz_i}{\int \frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1},\Phi_{-1}) + \frac{1+\beta}{2}P(z_i|x_i,\Lambda_0,\Phi_0) dz_i} = 0 \\
&\Rightarrow \sum_{i=1}^{N} \frac{1-\beta}{2}x_i E_{-1}[z_i'] - \frac{1-\beta}{2}\Lambda_1 E_{-1}[z_i z_i'] + \frac{1+\beta}{2}x_i E_0[z_i'] - \frac{1+\beta}{2}\Lambda_1 E_0[z_i z_i'] = 0 \\
&\Rightarrow \sum_{i=1}^{N} \frac{1-\beta}{2}x_i E_{-1}[z_i'] + \frac{1+\beta}{2}x_i E_0[z_i'] - \left(\frac{1-\beta}{2}\Lambda_1 E_{-1}[z_i z_i'] + \frac{1+\beta}{2}\Lambda_1 E_0[z_i z_i']\right) = 0 \\
&\Rightarrow \sum_{i=1}^{N} \frac{1-\beta}{2}x_i E_{-1}[z_i'] + \frac{1+\beta}{2}x_i E_0[z_i'] - \Lambda_1\left(\frac{1-\beta}{2}E_{-1}[z_i z_i'] + \frac{1+\beta}{2}E_0[z_i z_i']\right) = 0 \\
&\Rightarrow \sum_{i=1}^{N} \frac{1-\beta}{2}x_i E_{-1}[z_i'] + \frac{1+\beta}{2}x_i E_0[z_i'] = \Lambda_1 \sum_{i=1}^{N} \frac{1-\beta}{2}E_{-1}[z_i z_i'] + \frac{1+\beta}{2}E_0[z_i z_i']
\end{aligned}$$

so the update equation for Λ_1 is

$$\Lambda_1 = \frac{\left(\frac{1-\beta}{2}\sum_{j=1}^{N} x_i E_{-1}[z_i'] + \frac{1+\beta}{2}\sum_{j=1}^{N} x_i E_0[z_i']\right)}{\left(\frac{1-\beta}{2}\sum_{j=1}^{N} E_{-1}[z_i z_i'] + \frac{1+\beta}{2}\sum_{j=1}^{N} E_0[z_i z_i']\right)}$$

From equation (7) we have

$$\Rightarrow \sum_{j=1}^{N} \frac{\frac{1}{2}\frac{1+\alpha}{2}\int P^{\frac{1+\alpha}{2}} * (\Phi_1 - x_i x_i' + x_i z_i' \Lambda_1' + \Lambda_1 z_i x_i' - \Lambda_1 z_i z_i' \Lambda_1') * p(z_i) dz_i}{\int P^{\frac{1+\alpha}{2}} * p(z_i) dz_i} = 0$$

$$\begin{aligned}\Rightarrow \quad & \sum_{j=1}^{N} \frac{\int \frac{1-\beta}{2} P(z_i|x_i, \Lambda_{-1}, \Phi_{-1}) \left(\Phi_1 - x_i x_i' + x_i z_i' \Lambda_1' + \Lambda_1 z_i x_i' - \Lambda_1 z_i z_i' \Lambda_1'\right) dz_i}{\int \frac{1-\beta}{2} P(z_i|x_i, \Lambda_{-1}, \Phi_{-1}) + \frac{1+\beta}{2} P(z_i|x_i, \Lambda_0, \Phi_0) dz_i} \\ & + \frac{\int \frac{1+\beta}{2} P(z_i|x_i, \Lambda_0, \Phi_0) \left(\Phi_1 - x_i x_i' + x_i z_i' \Lambda_1' + \Lambda_1 z_i x_i' - \Lambda_1 z_i z_i' \Lambda_1'\right) dz_i}{\int \frac{1-\beta}{2} P(z_i|x_i, \Lambda_{-1}, \Phi_{-1}) + \frac{1+\beta}{2} P(z_i|x_i, \Lambda_0, \Phi_0) dz_i} = 0\end{aligned}$$

$$\begin{aligned}\Rightarrow \quad & \sum_{j=1}^{N} \Phi_1 - x_i x_i' + \frac{1-\beta}{2} x_i E_{-1}[z_i'] \Lambda_1' + \frac{1+\beta}{2} x_i E_0[z_i'] \Lambda_1' + \frac{1-\beta}{2} \Lambda_1 E_{-1}[z_i] x_i' \\ & + \frac{1+\beta}{2} \Lambda_1 E_0[z_i] x_i' - \left(\frac{1-\beta}{2} \Lambda_1 E_{-1}[z_i z_i'] + \frac{1+\beta}{2} \Lambda_1 E_0[z_i z_i']\right) \Lambda_1' = 0\end{aligned}$$

$$\begin{aligned}\Rightarrow \quad & \sum_{j=1}^{N} \Phi_1 - \sum_{j=1}^{N} x_i x_i' + \frac{1-\beta}{2} \sum_{j=1}^{N} x_i E_{-1}[z_i'] \Lambda_1' + \frac{1+\beta}{2} \sum_{j=1}^{N} x_i E_0[z_i'] \Lambda_1' + \frac{1-\beta}{2} \Lambda_1 \sum_{j=1}^{N} E_{-1}[z_i] x_i' \\ & + \frac{1+\beta}{2} \Lambda_1 \sum_{j=1}^{N} E_0[z_i] x_i' - \Lambda_1 \left(\frac{1-\beta}{2} \sum_{j=1}^{N} E_{-1}[z_i z_i'] + \frac{1+\beta}{2} \sum_{j=1}^{N} E_0[z_i z_i']\right) \Lambda_1' = 0\end{aligned}$$

$$\Rightarrow \sum_{j=1}^{N} \Phi_1 - \sum_{j=1}^{N} x_i x_i' + \frac{1-\beta}{2} \sum_{j=1}^{N} x_i E_{-1}[z_i'] \Lambda_1' + \frac{1+\beta}{2} \sum_{j=1}^{N} x_i E_0[z_i'] \Lambda_1' = 0$$

so the update equation for Φ_1 is

$$\begin{aligned}\Phi_1 &= \frac{1}{N} diag\left(\sum_{j=1}^{N} x_i x_i' - \left(\frac{1-\beta}{2} \sum_{j=1}^{N} x_i E_{-1}[z_i'] + \frac{1+\beta}{2} \sum_{j=1}^{N} x_i E_0[z_i']\right) \Lambda_1'\right) \\ \Phi_1 &= diag\left(C_{xx-} \frac{1}{N} \left(\frac{1-\beta}{2} \sum_{j=1}^{N} x_i E_{-1}[z_i'] + \frac{1+\beta}{2} \sum_{j=1}^{N} x_i E_0[z_i']\right) \Lambda_1'\right)\end{aligned}$$

C Appendix: Hellinger distance between two Gaussian

Proof 1.

$$
\begin{aligned}
H^2(P,Q) &= \frac{1}{2}\int \left(\sqrt{P}-\sqrt{Q}\right)^2 dx = 1-\int\sqrt{P}\sqrt{Q}dx \\
&= 1-\int\frac{1}{(2\pi)^{d/2}\left|P\right|^{1/4}\left|Q\right|^{1/4}}\exp\left(-\frac{x^T\left(P^{-1}+Q^{-1}\right)x}{4}\right)dx \\
&= 1-\frac{\left|\left(\frac{P^{-1}+Q^{-1}}{2}\right)^{-1}\right|^{1/2}}{\left|P\right|^{1/4}\left|Q\right|^{1/4}}\int\frac{1}{(2\pi)^{d/2}\left|\left(\frac{P^{-1}+Q^{-1}}{2}\right)^{-1}\right|^{1/2}}\exp\left(-\frac{x^T\left(P^{-1}+Q^{-1}\right)x}{4}\right)dx \\
&= 1-\frac{\left|\left(\frac{P^{-1}+Q^{-1}}{2}\right)^{-1}\right|^{1/2}}{\left|P\right|^{1/4}\left|Q\right|^{1/4}} = 1-\frac{\left|\left(\frac{P^{-1}+Q^{-1}}{2}\right)\right|^{-1/2}}{\left|P\right|^{1/4}\left|Q\right|^{1/4}} \\
&= 1-\left|\left(\frac{P+Q}{2PQ}\right)\right|^{-1/2}\left|P\right|^{-1/4}\left|Q\right|^{-1/4} \\
&= 1-\left|\frac{P+Q}{2}\right|^{-1/2}\left|P\right|^{1/4}\left|Q\right|^{1/4}
\end{aligned}
$$

Proof 2.

$$
\begin{aligned}
H^2(C,I) &= \frac{1}{2}\int\left(\sqrt{C}-\sqrt{I}\right)^2 dx \\
&= 1-\int\sqrt{C}\sqrt{I}dx \\
&= 1-\left|\frac{C+I}{2}\right|^{-1/2}\left|C\right|^{1/4}\left|I\right|^{1/4} \\
&= 1-\left|\frac{C+I}{2}\right|^{-1/2}\left|C^{-1/2}\right|^{-1/2} \\
&= 1-\left|\frac{C^{1/2}+C^{-1/2}}{2}\right|^{-1/2}
\end{aligned}
$$

assume

$$\lambda_k = eig(C^{1/2}) = sqrt(eig(C)) \ where \ k = 1,\dots n$$

then we have

$$\begin{aligned}
H^2(C,I) &= 1-\prod_{k=1}^{n}\left(\frac{\lambda_k^{-1}+\lambda_k}{2}\right)^{-1/2}\\
&= 1-\prod_{k=1}^{n}\left(\frac{1+\lambda_k^2}{2\lambda_k}\right)^{-1/2}\\
&= 1-\prod_{k=1}^{n}\left(\frac{2\lambda_k}{1+\lambda_k^2}\right)^{1/2}\\
&= 1-\prod_{k=1}^{n}\sqrt{\frac{2\lambda_k}{1+\lambda_k^2}}
\end{aligned}$$

D Appendix: Update equation of mixture of factor analyzers

D.1 multiple Φ

D.1.1 Non-Causal

From equation (33) we have

$$\Rightarrow \sum_{i=1}^{N} \frac{\frac{1+\alpha}{2} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} \int P^{\frac{1+\alpha}{2}} \left(\Phi_{1j}^{-1} x_i z_i' - \Phi_{1j}^{-1} \Lambda_{1j} z_i z_i'\right) p(z_i) dz_i}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} \int P^{\frac{1+\alpha}{2}} p(z_i) dz_i} = 0$$

$$\Rightarrow \sum_{i=1}^{N} \frac{\frac{1+\alpha}{2} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j (2\pi)^{-\frac{d}{2}} |\Sigma_j|^{-\frac{1}{2}} \int e^{-\frac{1}{2}(z_i - \Sigma_j A_j')' \Sigma_j^{-1} (z_i - \Sigma_j A_j')} \left(\Phi_{1j}^{-1} x_i z_i' - \Phi_{1j}^{-1} \Lambda_{1j} z_i z_i'\right) dz_i}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j (2\pi)^{-\frac{d}{2}} |\Sigma_j|^{-\frac{1}{2}} \int e^{-\frac{1}{2}(z_i - \Sigma_j A_j')' \Sigma_j^{-1} (z_i - \Sigma_j A_j')} dz_i} = 0$$

$$\Rightarrow \sum_{i=1}^{N} \frac{\frac{1+\alpha}{2} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j (\Phi_1^{-1} x_i E[z_i'] - \Phi_1^{-1} \Lambda_1 E[z_i z_i'])}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j} = 0$$

$$\Rightarrow \sum_{i=1}^{N} \frac{h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j (x_i E[z_i'] - \Lambda_1 E[z_i z_i'])}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j} = 0$$

$$\Rightarrow \sum_{i=1}^{N} h_{ij}' (x_i E[z_i'] - \Lambda_{1j} E[z_i z_i']) = 0$$

assume

$$\begin{aligned} h_{ij}' &= \frac{h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} b_j}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} b_j} \\ 1 &= \sum_{j=1}^{K} h_{ij}' \end{aligned}$$

so the update equation for Λ_{1j} is

$$\Lambda_{1j} = \sum_{j=1}^{N} h_{ij}' x_i E[z_i'] \left(\sum_{j=1}^{N} h_{ij}' E[z_i z_i'] \right)^{-1}$$

From equation (34) we have

$$\Rightarrow \quad \sum_{i=1}^{N} \frac{\frac{1}{2}\frac{1+\alpha}{2} h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} \int P^{\frac{1+\alpha}{2}} * \left(\Phi_{1j} - x_i x_i' + x_i z_i' \Lambda_{1j}' + \Lambda_{1j} z_i x_i' - \Lambda_{1j} z_i z_i' \Lambda_{1j}'\right) * p(z_i) dz_i}{\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} \int P^{\frac{1+\alpha}{2}} * p(z_i) dz_i}$$

$$\Rightarrow \quad \sum_{i=1}^{N} \frac{h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j \left(\Phi_{1j} - x_i x_i' + x_i E[z_i'] \Lambda_{1j}' + \Lambda_{1j} E[z_i] x_i' - \Lambda_{1j} E[z_i z_i'] \Lambda_{1j}'\right)}{\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j} = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} h_{ij}'(\Phi_{1j} - x_i x_i' + x_i E[z_i'] \Lambda_{1j}' + \Lambda_{1j} E[z_i] x_i' - \Lambda_{1j} E[z_i z_i'] \Lambda_{1j}') = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} h_{ij}'\Phi_{1j} - \sum_{i=1}^{N} h_{ij}' x_i x_i' + \sum_{i=1}^{N} h_{ij}' x_i E[z_i'] \Lambda_{1j}' + \sum_{i=1}^{N} h_{ij}' \Lambda_{1j} E[z_i] x_i' - \sum_{i=1}^{N} h_{ij}' \Lambda_{1j} E[z_i z_i'] \Lambda_{1j}' = 0$$

we know

$$\begin{aligned} \Lambda_{1j} &= \sum_{i=1}^{N} h_{ij}' x_i E[z_i'] \left(\sum_{i=1}^{N} h_{ij}' E[z_i z_i']\right)^{-1} \\ \Lambda_{1j}' &= \left(\sum_{i=1}^{N} h_{ij}' E[z_i z_i']\right)^{-1} \sum_{i=1}^{N} h_{ij}' E[z_i] x_i' \end{aligned}$$

so we get

$$\Rightarrow \quad \sum_{i=1}^{N} h_{ij}'\Phi_{1j} - \sum_{i=1}^{N} h_{ij}' x_i x_i' + \sum_{i=1}^{N} h_{ij}' x_i E[z_i'] \Lambda_{1j}' + \sum_{i=1}^{N} h_{ij}' \Lambda_{1j} E[z_i] x_i' - \sum_{i=1}^{N} \Lambda_{1j} h_{ij}' E[z_i] x_i' = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} h_{ij}'\Phi_{1j} - \sum_{i=1}^{N} h_{ij}' x_i x_i' + \sum_{i=1}^{N} h_{ij}' x_i E[z_i'] \Lambda_1' = 0$$

$$\Rightarrow \quad \Phi_{1j} \sum_{i=1}^{N} h_{ij}' = \sum_{i=1}^{N} h_{ij}' x_i x_i' - \sum_{i=1}^{N} h_{ij}' x_i E[z_i'] \Lambda_1'$$

and the update equation for Φ_{1j} is

$$\Phi_{1j} = \left(\sum_{i=1}^{N} h_{ij}' x_i x_i' - \sum_{i=1}^{N} h_{ij}' x_i E[z_i'] \Lambda_1'\right) \left(\sum_{i=1}^{N} h_{ij}'\right)^{-1}$$

From equation (38) and (39) we have

$$\frac{\partial Q^{(\alpha)}}{\partial \pi_{1j}} = \frac{\partial S^{(\alpha)}}{\partial \pi_{1j}} = \sum_{i=1}^{N} \frac{\frac{\partial W_i^{(\alpha)}}{\partial \pi_{1j}}}{W_i^{(\alpha)}} S^{(\alpha)}$$

where

$$\frac{\partial W_i^{(\alpha)}}{\partial \pi_{1j}} = h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right] \pi_{1j}^{-1}$$

then

$$\begin{aligned}
\frac{\partial Q^{(\alpha)}}{\partial \pi_{1j}} + \lambda &= 0 \Rightarrow \sum_{i=1}^{N} \frac{h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} E\left[P^{\frac{1+\alpha}{2}}\right] \pi_{1j}^{-1}}{W_i^{(\alpha)}} S^{(\alpha)} + \lambda = 0 \\
&\Rightarrow \sum_{i=1}^{N} h'_{ij} \pi_{1j}^{-1} S^{(\alpha)} + \lambda = 0 \\
&\Rightarrow \pi_{1j} = -\frac{\sum_{i=1}^{N} h'_{ij} S^{(\alpha)}}{\lambda}
\end{aligned}$$

We know that $\sum_{j=1}^{K} \pi_{1j} = 1$, then

$$\sum_{j=1}^{K} -\frac{\sum_{i=1}^{N} h'_{ij} S^{(\alpha)}}{\lambda} = 1 \Rightarrow \lambda = -NS^{(\alpha)}$$

so the update equation for π_{1j} is

$$\pi_{1j} = \frac{1}{N} \sum_{i=1}^{N} h'_{ij}$$

When $-1 < \alpha < 1$

$$b_j = \left(\frac{|\Phi_{1j}|^{-\frac{1}{2}}}{|\Phi_{0j}|^{-\frac{1}{2}}}\right)^{\frac{1+\alpha}{2}} |C_j|^{-\frac{1}{2}} |\Sigma_j|^{\frac{1}{2}} e^{-\frac{1}{2} x'_i \beta'_j C_j^{-1} \beta_j x_i} e^{\frac{1}{2} A_j \Sigma_j A'_j} e^{-\frac{1}{2} \frac{1+\alpha}{2} x'_i \left(\Phi_{1j}^{-1} - \Phi_{0j}^{-1}\right) x_i}$$

$$\begin{aligned}
h'_{ij} &= \frac{h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j} = \frac{\frac{\pi_j N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_{0j})}{\sum_{j=1}^{K} \pi_j N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_{0j})} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j}{\sum_{j=1}^{K} \frac{\pi_j N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_{0j})}{\sum_{j=1}^{K} \pi_j N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_{0j})} \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j} \\
&= \frac{\pi_j N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_0) \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j}{\sum_{j=1}^{K} \pi_j N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_0) \left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j}
\end{aligned}$$

where

$$N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_{0j}) = (2\pi)^{-\frac{p}{2}} \left|\Lambda_{0j}\Lambda'_{0j} + \Phi_{0j}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}x'_i\left(\Lambda_{0j}\Lambda'_{0j}+\Phi_{0j}\right)^{-1}x_i}$$

In order to calculate $N(x_i, \Lambda_{0j}\Lambda'_{0j} + \Phi_{0j}) * b_j$ we find that

$$\begin{aligned} |C_j|^{-\frac{1}{2}} &= \left|\left(I_d + \Lambda'_{0j}\Phi_0^{-1}\Lambda_{0j}\right)^{-1}\right|^{-\frac{1}{2}} = \left|I_d + \Lambda'_{0j}\Phi_0^{-1}\Lambda_{0j}\right|^{\frac{1}{2}} \\ \left|\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right|^{-\frac{1}{2}} &= \left|\Phi_0(\Phi_0^{-1}\Lambda_{0j}\Lambda'_{0j} + I)\right|^{-\frac{1}{2}} = |\Phi_0|^{-\frac{1}{2}} \left|(\Phi_0^{-1}\Lambda_{0j}\Lambda'_{0j} + I)\right|^{-\frac{1}{2}} \end{aligned}$$

According to Sylvester's determinant theorem

$$\left|I_d + \Lambda'_{0j}\Phi_0^{-1}\Lambda_{0j}\right| = \left|(\Phi_0^{-1}\Lambda_{0j}\Lambda'_{0j} + I_p)\right|$$

we get

$$|C_j|^{-\frac{1}{2}} \left|\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right|^{-\frac{1}{2}} = |\Phi_0|^{-\frac{1}{2}}$$

Also we find that

$$\begin{aligned} \beta'_j C_j^{-1} \beta_j &= \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \Lambda_{0j} \left(I_d + \Lambda'_{0j}\Phi_0^{-1}\Lambda_{0j}\right) \Lambda'_{0j} \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \\ &= \left(\left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \Lambda_{0j} + \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \Lambda_{0j}\Lambda'_{0j}\Phi_0^{-1}\Lambda_{0j}\right) \Lambda'_{0j} \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \\ &= \left(\left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} + \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \Lambda_{0j}\Lambda'_{0j}\Phi_0^{-1}\right) \Lambda_{0j}\Lambda'_{0j} \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \\ &= \left(\left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \left(I + \Lambda_{0j}\Lambda'_{0j}\Phi_0^{-1}\right)\right) \Lambda_{0j}\Lambda'_{0j} \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \\ &= \left(\left(\left(\Lambda_{0j}\Lambda'_{0j}\Phi_0^{-1} + I\right)\Phi_0\right)^{-1} \left(I + \Lambda_{0j}\Lambda'_{0j}\Phi_0^{-1}\right)\right) \Lambda_{0j}\Lambda'_{0j} \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \\ &= \Phi_0^{-1}\Lambda_{0j}\Lambda'_{0j} \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \end{aligned}$$

so

$$\begin{aligned} \beta'_j C_j^{-1} \beta_j + \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} &= \Phi_0^{-1}\Lambda_{0j}\Lambda'_{0j} \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} + \Phi_0^{-1}\Phi_0 \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} \\ &= \Phi_0^{-1} \left(\Lambda_{0j}\Lambda'_{0j} \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1} + \Phi_0 \left(\Lambda_{0j}\Lambda'_{0j} + \Phi_0\right)^{-1}\right) \\ &= \Phi_0^{-1} \end{aligned}$$

Therefore

$$N(x_i,\Lambda_{0j}\Lambda_{0j}'+\Phi_{0j})*b_j=\left(\frac{|\Phi_{1j}|^{-\frac{1}{2}}}{|\Phi_{0j}|^{-\frac{1}{2}}}\right)^{\frac{1+\alpha}{2}}e^{-\frac{1}{2}\frac{1+\alpha}{2}x_i'\left(\Phi_{1j}^{-1}-\Phi_{0j}^{-1}\right)x_i}N(x_i,\Phi_{0j})|\Sigma_j|^{\frac{1}{2}}e^{\frac{1}{2}A_j\Sigma_jA_j'}$$

and

$$h_{ij}'=\frac{\pi_{j,0}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}}\left(\frac{|\Phi_{1j}|^{-\frac{1}{2}}}{|\Phi_{0j}|^{-\frac{1}{2}}}\right)^{\frac{1+\alpha}{2}}e^{-\frac{1}{2}\frac{1+\alpha}{2}x_i'\left(\Phi_{1j}^{-1}-\Phi_{0j}^{-1}\right)x_i}N(x_i,\Phi_{0j})|\Sigma_j|^{\frac{1}{2}}e^{\frac{1}{2}A_j\Sigma_jA_j'}}{\sum_{j=1}^{K}\pi_{j,0}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}}\left(\frac{|\Phi_{1j}|^{-\frac{1}{2}}}{|\Phi_{0j}|^{-\frac{1}{2}}}\right)^{\frac{1+\alpha}{2}}e^{-\frac{1}{2}\frac{1+\alpha}{2}x_i'\left(\Phi_{1j}^{-1}-\Phi_{0j}^{-1}\right)x_i}N(x_i,\Phi_{0j})|\Sigma_j|^{\frac{1}{2}}e^{\frac{1}{2}A_j\Sigma_jA_j'}}$$

D.1.2 Causal

From equation (33) we have

$$\Rightarrow\quad\sum_{i=1}^{N}\frac{\frac{1+\alpha}{2}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int P^{\frac{1+\alpha}{2}}*\left(\Phi_1^{-1}x_iz_i'-\Phi_1^{-1}\Lambda_1z_iz_i'\right)*p(z_i)dz_i}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int P^{\frac{1+\alpha}{2}}*p(z_i)dz_i}=0$$

$$\Rightarrow\quad\sum_{i=1}^{N}\frac{h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int\frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1j},\Phi_{-1j}\left(\Phi_1^{-1}x_iz_i'-\Phi_1^{-1}\Lambda_1z_iz_i'\right)dz_i}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int\frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1j},\Phi_{-1j})+\frac{1+\beta}{2}P(z_i|x_i,\Lambda_{0j},\Phi_{0j})dz_i}$$

$$+\frac{h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int\frac{1+\beta}{2}P(z_i|x_i,\Lambda_{0j},\Phi_{0j})\left(\Phi_1^{-1}x_iz_i'-\Phi_1^{-1}\Lambda_1z_iz_i'\right)dz_i}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int\frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1j},\Phi_{-1j})+\frac{1+\beta}{2}P(z_i|x_i,\Lambda_{0j},\Phi_{0j})dz_i}=0$$

$$\Rightarrow\quad\sum_{i=1}^{N}\frac{h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\left(\frac{1-\beta}{2}x_iE_{-1}[z_i']-\frac{1-\beta}{2}\Lambda_1E_{-1}[z_iz_i']+\frac{1+\beta}{2}x_iE_0[z_i']-\frac{1+\beta}{2}\Lambda_1E_0[z_iz_i']\right)}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}}=0$$

we assume

$$h_{ij}'=\frac{h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}}$$

and we get

$$\Rightarrow \sum_{i=1}^{N} h'_{ij}\left(\frac{1-\beta}{2}x_iE_{-1}[z'_i]-\frac{1-\beta}{2}\Lambda_1E_{-1}[z_iz'_i]+\frac{1+\beta}{2}x_iE_0[z'_i]-\frac{1+\beta}{2}\Lambda_{1j}E_0[z_iz'_i]\right)=0$$

so the update equation for Λ_{1j} is

$$\Lambda_{1j}=\frac{\frac{1-\beta}{2}\sum_{i=1}^{N}h'_{ij}x_iE_{-1}[z'_i]+\frac{1+\beta}{2}\sum_{i=1}^{N}h'_{ij}x_iE_0[z'_i]}{\frac{1-\beta}{2}\sum_{i=1}^{N}h'_{ij}E_{-1}[z_iz'_i]+\frac{1+\beta}{2}\sum_{i=1}^{N}h'_{ij}E_0[z_iz'_i]}$$

From equation (34) we have

$$\Rightarrow \sum_{i=1}^{N}\frac{\frac{1}{2}\frac{1+\alpha}{2}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int P^{\frac{1+\alpha}{2}}*\left(\Phi_{1j}-x_ix'_i+x_iz'_i\Lambda'_{1j}+\Lambda_{1j}z_ix'_i-\Lambda_{1j}z_iz'_i\Lambda'_{1j}\right)*p(z_i)dz_i}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int P^{\frac{1+\alpha}{2}}*p(z_i)dz_i}=0$$

$$\Rightarrow \sum_{i=1}^{N}\frac{h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int\frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1j},\Phi_{-1j})\left(\Phi_1-x_ix'_i+x_iz'_i\Lambda'_{1j}+\Lambda_{1j}z_ix'_i-\Lambda_{1j}z_iz'_i\Lambda'_{1j}\right)dz_i}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int\frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1j},\Phi_{-1j})+\frac{1+\beta}{2}P(z_i|x_i,\Lambda_{0j},\Phi_{0j})dz_i}$$
$$+\frac{\int\frac{1+\beta}{2}P(z_i|x_i,\Lambda_{0j},\Phi_{0j})\left(\Phi_1-x_ix'_i+x_iz'_i\Lambda'_{1j}+\Lambda_{1j}z_ix'_i-\Lambda_{1j}z_iz'_i\Lambda'_{1j}\right)dz_i}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\int\frac{1-\beta}{2}P(z_i|x_i,\Lambda_{-1j},\Phi_{-1j})+\frac{1+\beta}{2}P(z_i|x_i,\Lambda_{0j},\Phi_{0j})dz_i}=0$$

$$\Rightarrow \sum_{i=1}^{N}\frac{h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\left(\Phi_{1j}-x_ix'_i+\frac{1-\beta}{2}x_iE_{-1}[z'_i]\Lambda'_{1j}+\frac{1-\beta}{2}\Lambda_{1j}E_{-1}[z_i]x'_i-\frac{1-\beta}{2}\Lambda_{1j}E_{-1}[z_iz'_i]\Lambda'_{1j}\right)}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}}$$
$$+\frac{h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}\left(\frac{1+\beta}{2}x_iE_0[z'_i]\Lambda'_{1j}+\frac{1+\beta}{2}\Lambda_{1j}E_0[z_i]x'_i-\frac{1+\beta}{2}\Lambda_{1j}E_0[z_iz'_i]\Lambda'_{1j}\right)}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}}=0$$

we assume

$$h'_{ij}=\frac{h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}}{\sum_{j=1}^{K}h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}}$$

and we get

$$\begin{aligned}\Rightarrow\quad & \sum_{i=1}^{N} h'_{ij}\left(\Phi_{1j} - x_i x'_i + \frac{1-\beta}{2} x_i E_{-1}[z'_i]\Lambda'_{1j} + \frac{1+\beta}{2} x_i E_0[z'_i]\Lambda'_{1j} + \frac{1-\beta}{2}\Lambda_{1j} E_{-1}[z_i] x'_i\right) \\ & + h'_{ij}\left(\frac{1+\beta}{2}\Lambda_{1j} E_0[z_i] x'_i - \frac{1-\beta}{2}\Lambda_{1j} E_{-1}[z_i z'_i]\Lambda'_{1j} - \frac{1+\beta}{2}\Lambda_{1j} E_0[z_i z'_i]\Lambda'_{1j}\right) = 0\end{aligned}$$

$$\begin{aligned}\Rightarrow\quad & \sum_{i=1}^{N} h'_{ij}\Phi_{1j} - \sum_{i=1}^{N} h'_{ij} x_i x'_i \\ & + \frac{1-\beta}{2}\sum_{i=1}^{N} h'_{ij} x_i E_{-1}[z'_i]\Lambda'_{1j} + \frac{1+\beta}{2}\sum_{i=1}^{N} h'_{ij} x_i E_0[z'_i]\Lambda'_{1j} + \frac{1-\beta}{2}\sum_{i=1}^{N} h'_{ij}\Lambda_{1j} E_{-1}[z_i] x'_i \\ & + \frac{1+\beta}{2}\sum_{i=1}^{N} h'_{ij}\Lambda_{1j} E_0[z_i] x'_i - \frac{1-\beta}{2}\sum_{i=1}^{N} h'_{ij}\Lambda_{1j} E_{-1}[z_i z'_i]\Lambda'_{1j} - \frac{1+\beta}{2}\sum_{i=1}^{N} h'_{ij}\Lambda_{1j} E_0[z_i z'_i]\Lambda'_{1j} = 0\end{aligned}$$

$$\Rightarrow \sum_{i=1}^{N} h'_{ij}\Phi_{1j} - \sum_{i=1}^{N} h'_{ij} x_i x'_i + \frac{1-\beta}{2}\sum_{i=1}^{N} h'_{ij} x_i E_{-1}[z'_i]\Lambda'_{1j} + \frac{1+\beta}{2}\sum_{i=1}^{N} h'_{ij} x_i E_0[z'_i]\Lambda'_{1j} = 0$$

so the update equation for Φ_{1j} is

$$\Phi_{1j} = \frac{1}{\sum_{i=1}^{N} h'_{ij}} diag\left(\sum_{i=1}^{N} h'_{ij} x_i x'_i - \left(\frac{1-\beta}{2}\sum_{i=1}^{N} h'_{ij} x_i E_{-1}[z'_i]\Lambda'_{1j} + \frac{1+\beta}{2}\sum_{i=1}^{N} h'_{ij} x_i E_0[z'_i]\Lambda'_{1j}\right)\right)$$

For $\pi_{0j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}}$ we have

$$\begin{aligned}\pi_{0j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} &= \pi_{0j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} = \pi_{0j}\frac{\pi_{1j}}{\pi_{0j}}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}-1} \\ &= \pi_{1j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{-\frac{1-\alpha}{2}} \\ &\approx \pi_{0j}\left(\frac{\pi_{0j}}{\pi_{-1j}}\right)^{-\frac{1-\beta}{2}}\end{aligned}$$

then

$$
\begin{aligned}
\pi_{0j}\left(\frac{\pi_{0j}}{\pi_{-1j}}\right)^{-\frac{1-\beta}{2}} &= \pi_{0j}\left(\frac{\pi_{0j}}{\pi_{-1j}}\right)^{-1}\left(\frac{\pi_{0j}}{\pi_{-1j}}\right)^{1-\frac{1-\beta}{2}} \\
&= \pi_{-1j}\left(\frac{\pi_{0j}}{\pi_{-1j}}\right)^{\frac{1+\beta}{2}} \\
&\approx \pi_{-1j}\left(1+\frac{1+\beta}{2}\left(\frac{\pi_{0j}}{\pi_{-1j}}-1\right)\right) \\
&= \pi_{-1j}\left(\frac{1-\beta}{2}+\frac{1+\beta}{2}\frac{\pi_{0j}}{\pi_{-1j}}\right) \\
&= \frac{1-\beta}{2}\pi_{-1j}+\frac{1+\beta}{2}\pi_{0j}
\end{aligned}
$$

So we have

$$
\pi_{0j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} \approx \frac{1-\beta}{2}\pi_{-1j}+\frac{1+\beta}{2}\pi_{0j}
$$

and

$$
\begin{aligned}
h'_{ij} &= \frac{N_{0j}\pi_{0j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}}}{\sum_{j=1}^{K} N_{0j}\pi_{0j}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}}} \\
&\approx \frac{N_{0j}\left(\frac{1-\beta}{2}\pi_{-1j}+\frac{1+\beta}{2}\pi_{0j}\right)}{\sum_{j=1}^{K} N_{0j}\left(\frac{1-\beta}{2}\pi_{-1j}+\frac{1+\beta}{2}\pi_{0j}\right)}
\end{aligned}
$$

D.2 single Φ

D.2.1 Non-Causal

From equation (34) we have

$$\Rightarrow \sum_{i=1}^{N} \frac{\frac{1}{2}\frac{1+\alpha}{2}\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} \int P^{\frac{1+\alpha}{2}} * \left(\Phi_1 - x_i x_i' + x_i z_i'\Lambda_{1j}' + \Lambda_{1j} z_i x_i' - \Lambda_{1j} z_i z_i' \Lambda_{1j}'\right) * p(z_i) dz_i}{\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} \int P^{\frac{1+\alpha}{2}} * p(z_i) dz_i} = 0$$

$$\Rightarrow \sum_{i=1}^{N} \frac{\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j' \left(\Phi_1 - x_i x_i' + x_i E[z_i']\Lambda_{1j}' + \Lambda_{1j} E[z_i] x_i' - \Lambda_{1j} E[z_i z_i'] \Lambda_{1j}'\right)}{\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{1j}}{\pi_{0j}}\right)^{\frac{1+\alpha}{2}} b_j'} = 0$$

$$\Rightarrow \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}'(\Phi_1 - x_i x_i' + x_i E[z_i']\Lambda_{1j}' + \Lambda_{1j} E[z_i] x_i' - \Lambda_{1j} E[z_i z_i']\Lambda_{1j}') = 0$$

$$\Rightarrow \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}'\Phi_1 - \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i x_i' + \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E[z_i']\Lambda_{1j}'$$
$$+ \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}'\Lambda_{1j} E[z_i] x_i' - \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}'\Lambda_{1j} E[z_i z_i']\Lambda_{1j}'$$

we know

$$\begin{aligned} \Lambda_{1j} &= \sum_{i=1}^{N} h_{ij}' x_i E[z_i'] \left(\sum_{i=1}^{N} h_{ij}' E[z_i z_i']\right)^{-1} \\ \Lambda_{1j}' &= \left(\sum_{i=1}^{N} h_{ij}' E[z_i z_i']\right)^{-1} \sum_{i=1}^{N} h_{ij}' E[z_i] x_i' \end{aligned}$$

so we get

$$\Rightarrow \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}'\Phi_1 - \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i x_i' + \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E[z_i']\Lambda_{1j}' = 0$$

and the update equation for Φ_1 is

$$\Phi_1 = \frac{1}{N} diag\left(\sum_{i=1}^{N} x_i x_i' - \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E[z_i']\Lambda_{1j}'\right)$$

D.2.2 Causal

From equation (34) we have q

$$\Rightarrow \sum_{i=1}^{N} \frac{\frac{1}{2}\frac{1+\alpha}{2}\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} \int P^{\frac{1+\alpha}{2}} * \left(\Phi_1 - x_i x_i' + x_i z_i'\Lambda_{1j}' + \Lambda_{1j} z_i x_i' - \Lambda_{1j} z_i z_i' \Lambda_{1j}'\right) * p(z_i) dz_i}{\sum_{j=1}^{K} h_{ij}\left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} \int P^{\frac{1+\alpha}{2}} * p(z_i) dz_i} = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} \frac{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} \int \frac{1-\beta}{2} P(z_i|x_i, \Lambda_{-1j}, \Phi_{-1}) \left(\Phi_1 - x_i x_i' + x_i z_i' \Lambda_{1j}' + \Lambda_{1j} z_i x_i' - \Lambda_{1j} z_i z_i' \Lambda_{1j}'\right) dz_i}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} \int \frac{1-\beta}{2} P(z_i|x_i, \Lambda_{-1j}, \Phi_{-1j}) + \frac{1+\beta}{2} P(z_i|x_i, \Lambda_{0j}, \Phi_{0j}) dz_i}$$

$$+ \frac{\int \frac{1+\beta}{2} P(z_i|x_i, \Lambda_{0j}, \Phi_0) \left(\Phi_1 - x_i x_i' + x_i z_i' \Lambda_{1j}' + \Lambda_{1j} z_i x_i' - \Lambda_{1j} z_i z_i' \Lambda_{1j}'\right) dz_i}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} \int \frac{1-\beta}{2} P(z_i|x_i, \Lambda_{-1j}, \Phi_{-1j}) + \frac{1+\beta}{2} P(z_i|x_i, \Lambda_{0j}, \Phi_{0j}) dz_i} = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} \frac{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} \left(\Phi_1 - x_i x_i' + \frac{1-\beta}{2} x_i E_{-1}[z_i'] \Lambda_{1j}' + \frac{1-\beta}{2} \Lambda_{1j} E_{-1}[z_i] x_i' - \frac{1-\beta}{2} \Lambda_{1j} E_{-1}[z_i z_i'] \Lambda_{1j}'\right)}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}}$$

$$+ \frac{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}} \left(\frac{1+\beta}{2} x_i E_0[z_i'] \Lambda_{1j}' + \frac{1+\beta}{2} \Lambda_{1j} E_0[z_i] x_i' - \frac{1+\beta}{2} \Lambda_{1j} E_0[z_i z_i'] \Lambda_{1j}'\right)}{\sum_{j=1}^{K} h_{ij} \left(\frac{\pi_{j,1}}{\pi_{j,0}}\right)^{\frac{1+\alpha}{2}}} = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \left(\Phi_1 - x_i x_i' + \frac{1-\beta}{2} x_i E_{-1}[z_i'] \Lambda_{1j}' + \frac{1-\beta}{2} \Lambda_{1j} E_{-1}[z_i] x_i' - \frac{1-\beta}{2} \Lambda_{1j} E_{-1}[z_i z_i'] \Lambda_{1j}'\right)$$

$$+ \sum_{j=1}^{K} h_{ij}' \left(\frac{1+\beta}{2} x_i E_0[z_i'] \Lambda_{1j}' + \frac{1+\beta}{2} \Lambda_{1j} E_0[z_i] x_i' - \frac{1+\beta}{2} \Lambda_{1j} E_0[z_i z_i'] \Lambda_{1j}'\right) = 0$$

$$\Rightarrow \quad \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Phi_1 - \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' x_i x_i' + \frac{1-\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' x_i E_{-1}[z_i'] \Lambda_{1j}' + \frac{1+\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' x_i E_0[z_i'] \Lambda_{1j}'$$

$$+ \frac{1-\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Lambda_{1j} E_{-1}[z_i] x_i' + \frac{1+\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Lambda_{1j} E_0[z_i] x_i'$$

$$- \frac{1-\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Lambda_{1j} E_{-1}[z_i z_i'] \Lambda_{1j}' - \frac{1+\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Lambda_{1j} E_0[z_i z_i'] \Lambda_{1j}' = 0$$

$$\Rightarrow \quad N\Phi_1 - \sum_{i=1}^{N} x_i x_i' + \frac{1-\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' x_i E_{-1}[z_i'] \Lambda_{1j}' + \frac{1+\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' x_i E_0[z_i'] \Lambda_{1j}'$$

$$+ \frac{1-\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Lambda_{1j} E_{-1}[z_i] x_i' + \frac{1+\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Lambda_{1j} E_0[z_i] x_i'$$

$$- \frac{1-\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Lambda_{1j} E_{-1}[z_i z_i'] \Lambda_{1j}' - \frac{1+\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} h_{ij}' \Lambda_{1j} E_0[z_i z_i'] \Lambda_{1j}' = 0$$

$$\Rightarrow N\Phi_1 - \sum_{i=1}^{N} x_i x_i' + \frac{1-\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E_{-1}[z_i']\Lambda_{1j}' + \frac{1+\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E_0[z_i']\Lambda_{1j}' = 0$$

so the update equation for Φ_1 is

$$\Phi_1 = \frac{1}{N} diag\left(\sum_{i=1}^{N} x_i x_i' - \left(\frac{1-\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E_{-1}[z_i']\Lambda_{1j}' + \frac{1+\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E_0[z_i']\Lambda_{1j}'\right)\right)$$

When $\beta = 1(\alpha = -1)$ the update euqations are

$$\begin{aligned}
\Lambda_{1j} &= \left(\sum_{i=1}^{N} h_{ij}' x_i E_0[z_i']\right)\left(\sum_{i=1}^{N} h_{ij}' E_0[z_i z_i']\right)^{-1} \\
\Phi_1 &= \frac{1}{N} diag\left(\sum_{i=1}^{N} x_i x_i' - \left(\sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i E_0[z_i']\right)\Lambda_{1j}'\right)
\end{aligned}$$

where

$$h_{ij}' = \frac{\pi_{0j} N_{0j}}{\sum\limits_{j=1}^{K} \pi_{0j} N_{0j}}$$

We have $E_0[z_i'] = x_i'\beta_{0j}$ and $E_0[z_i z_i'] = C_{0j}^{-1} + \beta_{0j} x_i x_i' \beta_{0j}'$ then

$$\begin{aligned}
\Lambda_{1j} &= \left(\sum_{i=1}^{N} h_{ij}' x_i x_i' \beta_{0j}\right)\left(\sum_{i=1}^{N} h_{ij}'\left(C_{0j}^{-1} + \beta_{0j} x_i x_i' \beta_{0j}'\right)\right)^{-1} \\
\Phi_1 &= \frac{1}{N} diag\left(\sum_{i=1}^{N} x_i x_i' - \sum_{i=1}^{N}\sum_{j=1}^{K} h_{ij}' x_i x_i' \beta_{0j}\Lambda_{1j}'\right)
\end{aligned}$$

where

$$\begin{aligned}
C_{0j}^{-1} &= I_d - \Lambda_{0j}'(\Phi_{0j} + \Lambda_{0j}\Lambda_{0j}')^{-1}\Lambda_{0j} \\
\beta_{0j} &= \Lambda_{0j}'(\Phi_{0j} + \Lambda_{0j}\Lambda_{0j}')^{-1}
\end{aligned}$$

therefore

$$\begin{aligned}
\Lambda_{1j} &= \left(\sum_{i=1}^{N} h'_{ij} x_i x'_i \beta'_{0j}\right)\left(\sum_{i=1}^{N} h'_{ij}\left(I_d - \beta_{0j}\Lambda_{0j} + \beta_{0j} x_i x'_i \beta'_{0j}\right)\right)^{-1} \\
&= \frac{\sum_{i=1}^{N} h'_{ij} x_i x'_i \beta'_{0j}}{\sum_{i=1}^{N} h'_{ij}\left(I_d - \beta_{0j}\Lambda_{0j} + \beta_{0j} x_i x'_i \beta'_{0j}\right)} \\
&= \frac{\sum_{i=1}^{N} h'_{ij} x_i x'_i \beta'_{0j}}{\sum_{i=1}^{N} h'_{ij} I_d - \sum_{i=1}^{N} h'_{ij}\beta_{0j}\Lambda_{0j} + \sum_{i=1}^{N} h'_{ij}\beta_{0j} x_i x'_i \beta'_{0j}} \\
&= \frac{\sum_{i=1}^{N} h'_{ij} x_i x'_i \beta'_{0j}}{I_d \sum_{i=1}^{N} h'_{ij} - \beta_{0j}\Lambda_{0j}\sum_{i=1}^{N} h'_{ij} + \sum_{i=1}^{N} h'_{ij}\beta_{0j} x_i x'_i \beta'_{0j}} \\
&= \frac{\sum_{i=1}^{N} h'_{ij} x_i x'_i \beta'_{0j}\left(\sum_{i=1}^{N} h'_{ij}\right)^{-1}}{I_d - \beta_{0j}\Lambda_{0j} + \sum_{i=1}^{N} h'_{ij}\beta_{0j} x_i x'_i \beta'_{0j}\left(\sum_{i=1}^{N} h'_{ij}\right)^{-1}} \\
&= \frac{\sum_{i=1}^{N} h'_{ij} x_i x'_i \left(\sum_{i=1}^{N} h'_{ij}\right)^{-1} \beta'_{0j}}{I_d - \beta_{0j}\Lambda_{0j} + \beta_{0j}\sum_{i=1}^{N} h'_{ij} x_i x'_i \left(\sum_{i=1}^{N} h'_{ij}\right)^{-1} \beta'_{0j}}
\end{aligned}$$

$$\begin{aligned}
\Phi_1 &= \frac{1}{N} diag\left(\sum_{i=1}^{N} x_i x'_i - \sum_{i=1}^{N}\sum_{j=1}^{K} h'_{ij} x_i x'_i \beta_{0j}\Lambda'_{1j}\right) \\
&= diag\left(C - \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} h'_{ij} x_i x'_i \beta_{0j}\Lambda'_{1j}\right) \\
&= diag\left(C - \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} h'_{ij} x_i x'_i \beta_{0j}\Lambda'_{1j}\right) \\
&= diag\left(C - \sum_{j=1}^{K}\frac{\sum_{i=1}^{N} h'_{ij} x_i x'_i}{N}\beta_{0j}\Lambda'_{1j}\right)
\end{aligned}$$

The above update equations are the same with Ghahramani and Hinton [51].

E Appendix: Gradient of α-log-likelihood function

The derivatives of f are given by

$$\frac{\partial f}{\partial \theta} = L^{\frac{1+\alpha}{2}} \frac{\partial l}{\partial \theta}$$

where

$$l = -\frac{N}{2}\left(p \log 2\pi + \log|\Sigma| + trS\Sigma^{-1}\right)$$

and

$$\Sigma = \Phi + \Lambda\Psi\Lambda^T$$

For Λ

$$\begin{aligned}
\frac{\partial l}{\partial \Lambda} &= -\frac{N}{2}\left(\frac{\partial \log|\Sigma|}{\partial \Lambda} + \frac{\partial trS\Sigma^{-1}}{\partial \Lambda}\right) \\
&= -\frac{N}{2}\left(\frac{\partial \log|\Sigma|}{\partial \Sigma}\frac{\partial \Sigma}{\partial \Lambda} + \frac{\partial trS\Sigma^{-1}}{\partial \Sigma}\frac{\partial \Sigma}{\partial \Lambda}\right) \\
&= -\frac{N}{2}\left(2\Sigma^{-1}\Lambda\Psi - 2\Sigma^{-1}S\Sigma^{-1}\Lambda\Psi\right) \\
&= N(\Sigma^{-1}S\Sigma^{-1}\Lambda\Psi - \Sigma^{-1}\Lambda\Psi) \\
&= N(\Sigma^{-1}S\Sigma^{-1} - \Sigma^{-1})\Lambda\Psi \\
&= N\Sigma^{-1}(S - \Sigma)\Sigma^{-1}\Lambda\Psi
\end{aligned}$$

For Φ

$$\begin{aligned}
\frac{\partial l}{\partial \Phi} &= -\frac{N}{2}\left(\frac{\partial \log|\Sigma|}{\partial \Phi} + \frac{\partial trS\Sigma^{-1}}{\partial \Phi}\right) \\
&= -\frac{N}{2}\left(\frac{\partial \log|\Sigma|}{\partial \Sigma}\frac{\partial \Sigma}{\partial \Phi} + \frac{\partial trS\Sigma^{-1}}{\partial \Sigma}\frac{\partial \Sigma}{\partial \Phi}\right) \\
&= -\frac{N}{2}\left(\Sigma^{-1} - \Sigma^{-1}S\Sigma^{-1}\right) \\
&= \frac{N}{2} diag[\Sigma^{-1}(S - \Sigma)\Sigma^{-1}]
\end{aligned}$$

For Ψ

$$\begin{aligned}
\frac{\partial l}{\partial \Psi} &= -\frac{N}{2}\left(\frac{\partial \log|\Sigma|}{\partial \Psi} + \frac{\partial tr S\Sigma^{-1}}{\partial \Psi}\right) \\
&= -\frac{N}{2}\left(\frac{\partial \log|\Sigma|}{\partial \Sigma}\frac{\partial \Sigma}{\partial \Psi} + \frac{\partial tr S\Sigma^{-1}}{\partial \Sigma}\frac{\partial \Sigma}{\partial \Psi}\right) \\
&= -\frac{N}{2}\left(\Lambda^T\Sigma^{-1}\Lambda - \Lambda^T\Sigma^{-1}S\Sigma^{-1}\Lambda\right) \\
&= \frac{N}{2}\Lambda^T\Sigma^{-1}(S-\Sigma)\Sigma^{-1}\Lambda
\end{aligned}$$