MATH 5472 Final Project Report

Wang Xiaopeng

December 2024

1 Overview of the Paper

The paper *"Fitting Multilevel Factor Models"*(arXiv:2409.12067) addresses a significant challenge in statistical modeling: fitting multilevel factor models with covariance given by multilevel low-rank matrix (MLR) efficiently and at scale in linear time and storage complexity per iteration.

1.1 Why is it Interesting?

1.1.1 Prior Definition and Assumption

Factor model
Factor analysis is found to be useful in psychology, finance, economics, and statistics. The main idea is to explain the variability among the observed variables using a smaller number of latent variables called factors. Additionally, factor models can break down a covariance matrix into two components: a low-rank matrix that captures the influence of underlying factors and a diagonal matrix that accounts for idiosyncratic variances.

Multilevel factor model
A multilevel factor model is a statistical framework designed to analyze data with hierarchical structures, such as groups of individuals within larger populations or nested data across multiple levels. These models extend traditional factor models by partitioning the factors into global and local components. This partitioning allows the decomposition of the variances of the observed variables into contributions from each level of the hierarchical data.

In this way, Multilevel factor models are crucial in many real-world applications, such as:

- **Social Science:** Analyzing data where individuals are nested within groups (e.g. students within schools, patients within hospitals) requires modeling both individual-level and group-level variability.
- **Finance:** Modeling dependencies among assets with hierarchical structures, such as stocks grouped by industries.
- **Genomic:** Understanding genetic variations between individuals, families, and populations, where data naturally form hierarchies.

Existing fitting method
Various approaches have been used to fit multilevel models, including maximum likelihood and Bayesian estimation techniques, and Frobenius norm-based fitting methods. Among them, the EM algorithm, the Newton-Raphson algorithm, iterative generalized least squares, the Fisher scoring algorithm, and the Markov Chain Monte Carlo is commonly used. However, their results are not satisfactory under all possible data conditions. For instance, the computation and storage cost of fitting a multilevel factor model on a large-scale dataset is very large.

Matrix Setting
In this part, I attach the definition of each matrix in the original paper, for the facility of the following explanations.
An $n \times n$ contiguous PSD MLR matrix A with L levels has the form:

$$A = A_1 + \cdots + A_L,$$

Where A_l is a PSD block diagonal matrix,

$$A_l = \mathbf{blkdiag}(A_{l,1}, \ldots, A_{l,p_l}), \quad l = 1, \ldots, L,$$

where **blkdiag** represents the direct sum of blocks $A_{l,k} \in R^{n_{l,k} \times n_{l,k}}$ for $k = 1, \ldots, p_l$. Here, p_l is the size of the partition at level l, with $p_1 = 1$, and

$$\sum_{k=1}^{p_l} n_{l,k} = n, \quad l = 1, \ldots, L.$$

We require that the blocks at level l have a rank not exceeding r_l, and they are given in the factored form as:

$$A_{l,k} = F_{l,k} F_{l,k}^T, \quad F_{l,k} \in R^{n_{l,k} \times r_l}, \quad l = 1, \ldots, L-1, \quad k = 1, \ldots, p_l,$$

where $F_{l,k}$ is referred to as the factor of block k on level l.
For each level $l = 1, \ldots, L-1$, define:

$$F_l = \text{blkdiag}(F_{l,1}, \ldots, F_{l,p_l}) \in R^{n \times p_l r_l}.$$

Then we have:

$$A_l = F_l F_l^T, \quad l = 1, \ldots, L-1.$$

Define:

$$F = \begin{bmatrix} F_1 & \cdots & F_{L-1} \end{bmatrix} \in R^{n \times s},$$

with $s = \sum_{l=1}^{L-1} p_l r_l$. Then we can write A as:

$$A = \left(FD^{1/2}\right)\left(FD^{1/2}\right)^T = FF^T + D,$$

Where D is a diagonal matrix.

Problem Setting
In this part, I attach the problem setting in the original paper.
Considering a multilevel factor model:

$$y = Fz + e,$$

Where $F \in R^{n \times s}$ is the factor matrix of a PSD MLR, $z \in R^s$ are the factor scores, with $z \sim N(0, I_s)$, $e \in R^n$ are the idiosyncratic terms, with $e \sim N(0, D)$, then $y \in R^n$ is a Gaussian random vector with zero mean and covariance matrix Σ that is PSD MLR:

$$\Sigma = \left(FD^{1/2}\right)\left(FD^{1/2}\right)^T = FF^T + D.$$

We want to fit F and D from observed y.

Assumption
In this paper, we focus on a special case of a multi-level factor model under the following assumptions. First, the model does not have intercept and linear covariates. Second, observations follow a normal distribution with a multilevel low-rank matrix (MLR) as the covariance matrix. Third, both rank allocation and hierarchical partition are given, and we only focus on the fitting part.

1.1.2 Contribution of this paper

This paper compares its proposed MLE method with the method that focuses on fitting the covariance matrix using the Frobenius norm-based method. However, such Frobenius method has the following shortages for fitting covariance models. First, the Frobenius norm is invariant to changes in coordinates, while in maximum likelihood estimation (MLE), changes in different coordinates have different meanings. Secondly, the Frobenius norm may result in a model with smaller eigenvalues in the covariance matrix, which is naturally avoided by MLE. Thirdly, the Frobenius norm-based loss does not rely on any specific data distribution, making it distribution-independent, whereas MLE exploits knowledge of the underlying distribution. Therefore, the author proposes an MLE method based on the EM algorithm to prevent those shortages.

As for the proposed method, what makes it interesting most is its focus on overcoming computational bottlenecks that have traditionally limited the use of multilevel models in large-scale applications. In this way, we can apply the multilevel factor model to more real-world scenarios utilizing a larger dataset than before, such as fitting the high-frequency stock returns from various industries. Additionally, during the calculation of the EM algorithm, the author introduces an innovative method for efficiently calculating the inverse of a positive semi-definite (PSD) multilevel low-rank (MLR) matrix utilizing properties of structures matrices and recursive Sherman-Morrison-Woodbury (SMW) algorithm. As a result, the proposed method can achieve higher log-likelihood than the baseline Frobenius norm-based method.

In summary, I have generalized its contributions as follows:

- **Scalability in time and memory:** This paper proposes an MLE method utilizing the EM algorithm with data augmentation, which scales linearly with the data size per iteration, both in terms of time and memory. This is a significant improvement over conventional methods, which often scale quadratically or worse.

- **Efficient inverse matrix computation method:** The author presents a novel approach for computing the inverse of a positive semi-definite (PSD) multilevel low-rank (MLR) matrix, which is a fundamental step in fitting these models, which significantly decreases the computation time of fitting.

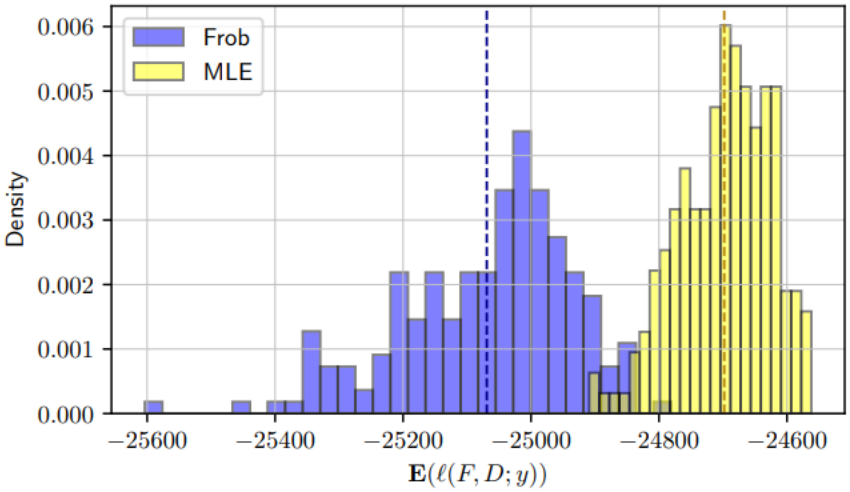

Figure 1: Expexted log-likelihood for proposed MLE method and Frobenius norm-based fitting method on synthetic dataset

- **Practical improvement in performance:** The proposed method can achieve a higher log-likelihood compared to the fitting of the Frobenius norm, which is a commonly used fitting method for factor models, in both a synthetic dataset, as shown in Figure 1, and a real asset covariance matrix containing daily returns of 5000 assets over 300 trading days.

These contributions collectively improve the scalability, efficiency, and accessibility of multilevel factor models, making them more feasible for large-scale applications in fields such as social science, finance, and genomics. Therefore, this paper provides a strong foundation and opens up valuable opportunities for further exploration and development in that area.

1.2 What are the Major Challenges to Solve?

There are two main challenges to solve in this paper. One is choosing a suitable algorithm for maximizing the log-likelihood, and the other one is relieving the costly computation of Σ^{-1}.

1.2.1 Difficulty in maximizing log-likelihood

Let observed samples be $y_1, \ldots, y_N \in R^n$, then:

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix} \in R^{N \times n}.$$

Following the problem setting in Section 1.1.1, the log-likelihood is:

$$\ell(F, D; Y) = -\frac{nN}{2}\log(2\pi) - \frac{N}{2}\log\det(FF^T + D) - \frac{1}{2}\operatorname{Tr}\left((FF^T + D)^{-1}Y^TY\right).$$

It could be very difficult and consuming to maximize $\ell(F, D; Y)$ directly. This paper utilizes the EM algorithm. However, it's still not easy for an EM algorithm. The author makes use of data augmentation to further simplify such a problem.

1.2.2 Difficulty in efficient computation of the algorithm

In the calculation of the proposed EM algorithm, we need to compute $(\Sigma^i)^{-1}$ recursively for F^i, D^i in each iteration, and its complexity is $O(n^3)$ that accounts for the high cost of fitting a multilevel factor model. Hence, it is crucial to address such challenges for the practical use of the aforementioned EM algorithm.

1.3 How Does the Proposed Method Address the Challenges?

In this paper, the author proposes an EM algorithm with data augmentation to maximize the log-likelihood. Plus, for the efficient computation of Σ^{-1}, the author exploits the hierarchical structure of $Sigma$ that is composed of a low-rank matrix plus a diagonal one and applies the Sherman-Morrison-Woodbury matrix identity. Next, I'd like to explain these methods in the following subsections one by one.

1.3.1 EM Algorithm for MLE

In the original log-likelihood formula, there is a term FF^T where rows of F are mixed such that it's hard to apply the expectation step. To solve that challenge, the author augments a new observed latent data $z_1, \ldots, z_N \in R^s$, organized into the matrix $Z \in R^{N\times s}$ together with Y. As a result, the new log-likelihood of data (Y, Z) is

$$\ell(F, D; Y, Z) = -\frac{(n+s)N}{2}\log(2\pi) - \frac{N}{2}\log\det D - \frac{1}{2}\|D^{-1/2}(Y - ZF^T)\|_F^2 - \frac{1}{2}\|Z\|_F^2.$$

Thanks to the data augmentation, $\ell(F, D; Y, Z)$ is separable with respect to rows of F and it's easier to compute $D^{-\frac{1}{2}}$ because D is diagonal compared with computing $(FF^T + D)^{-1}$ in the original log-likelihood formula. All the remaining steps in the EM algorithm of this problem are tractable and conventional. I have attached the full EM algorithm in the paper in the Appendix. In brief, the utilization of data augmentation makes it feasible and efficient for us to maximize the log-likelihood using the EM algorithm with respect to F and D.

1.3.2 Efficient computation of Σ^{-1}

As is shown in the EM algorithm in the Appendix, we need to compute Σ_i^{-1} for the ith iteration in the Expectation step. As is known to all, the computation of a matrix is expensive, and we need to solve such costly computations iteratively. Hence, the author comes up with an efficient method utilizing the property of structured matrices and recursive SMW algorithm.

Structured matrices property.

The following structured matrices' properties make sure that we can use the SMW algorithm for computing the inverse of a large matrix and its sparsity is maintained.

- Most importantly, the inverse of a block diagonal matrix is still a block diagonal matrix containing inverses of each block. By this property, we can compute Σ^{-1} by computing each block's inverse and adding them together instead, which can significantly decrease n in the complexity of $O(n^3)$.

- Matrix
$$F_{(l+1)+} + F_{(l+1)+}^T = \sum_{l'=l+1}^{L-1} F_{l'} F_{l'}^T$$
has the same sparsity as $F_{l+1}F_{l+1}^T$

- Matrix
$$M_0 = \big(F_{l+1} + F_{l+1}^T + D\big)^{-1} F_l.$$
Has the same sparsity as F_l. Hence, $F_l^T M_0$ is a block diagonal matrix with p_l blocks of size r_l

Recursive SMW algorithm.
SMW matrix identity is as follows:

$$(FF^T + D)^{-1} = D^{-1} - D^{-1}F\big(I_s + F^T D^{-1} F\big)^{-1} F^T D^{-1}.$$

Thus, we have:

$$\begin{aligned}\big(F_{l+} + F_{l+}^T + D\big)^{-1} &= \big(F_{l+1} + F_{l+1}^T + D\big)^{-1} \\ &\quad - \big(F_{l+1} + F_{l+1}^T + D\big)^{-1} F_l \Big(I_{p_l r_l} + F_l^T \big(F_{l+1} + F_{l+1}^T + D\big)^{-1} F_l\Big)^{-1} \\ &\quad F_l^T \big(F_{l+1} + F_{l+1}^T + D\big)^{-1} \\ &= \big(F_{l+1} + F_{l+1}^T + D\big)^{-1} - M_0 \Big(I_{p_l r_l} + F_l^T M_0\Big)^{-1} M_0^T.\end{aligned}$$

Define
$$H_l = M_0 \big(I_{p_l r_l} + F_l^T M_0\big)^{-1/2}.$$

According to the structured matrices property mentioned above, H_l, M_0 and F_l have the same sparsity and, hence, we can iterate the calculation from the bottom level to the top one and :

$$\Sigma^{-1} = -H_1 H_1^T - \cdots - H_{L-1} H_{L-1}^T + D^{-1}.$$

In this way, instead of computing the inverse of a large matrix. We can compute the inverses of many block matrices and a diagonal matrix and significantly improve the efficiency of the EM algorithm.

2 Experiment

Due to the dataset restriction. I only replicate the experiment on synthetic data. My code for the experiment part can be found in Github.

3 Conclusion

The paper *"Fitting Multilevel Factor Models"* provides a significant advancement in statistical modeling by addressing computational and scalability challenges in multilevel factor analysis. The proposed methods are not only theoretically sound but also practical, as evidenced by their linear time and storage complexities and the open-source implementation. These innovations make this work a valuable resource for researchers and practitioners working with hierarchical data structures in various domains.

Appendix

3.1 Expectation step

$$
\begin{aligned}
Q(F, D; F_0, D_0) &= E\big[\ell(F, D; Y, Z) \mid Y, F_0, D_0\big] \\
&= -\frac{(n+s)N}{2}\log(2\pi) - \frac{N}{2}\log\det D \\
&\quad - \frac{1}{2}\sum_{i=1}^{N} E\big[\operatorname{Tr}\left(D^{-1}(y_i - Fz_i)(y_i - Fz_i)^T\right) + \operatorname{Tr}(z_i z_i^T) \mid Y, F_0, D_0\big] \\
&= -\frac{(n+s)N}{2}\log(2\pi) - \frac{N}{2}\log\det D - \frac{1}{2}\sum_{i=1}^{N}\operatorname{Tr}\left(E[z_i z_i^T \mid y_i, F_0, D_0]\right) \\
&\quad - \frac{1}{2}\sum_{i=1}^{N}\operatorname{Tr}\Big(D^{-1}[y_i y_i^T - 2FE[z_i \mid y_i, F_0, D_0]y_i) + FE[z_i z_i^T \mid y_i, F_0, D_0]F^T]\Big).
\end{aligned}
$$

where,

$$
\begin{aligned}
\operatorname{cov}(y, z) &= E[Fzz^T] = F, \\
\operatorname{cov}(y, y) &= FF^T + D.
\end{aligned}
$$

Thus, (z, y) follows:

$$
\mathcal{N}(\mathbf{0}, \begin{bmatrix} I_s & F^T \\ F & \Sigma \end{bmatrix}),
$$

and $(z_i \mid y_i, F_0, D_0)$ follows:

$$
\mathcal{N}\big(F_0^T(\Sigma_0)^{-1}y_i, I_s - F_0^T(\Sigma_0)^{-1}F_0\big),
$$

Therefore, we have

$$
\begin{aligned}
\sum_{i=1}^{N} E\big[z_i z_i^T \mid y_i, F_0, D_0\big] &= \sum_{i=1}^{N}\operatorname{cov}(z_i, z_i \mid y_i, F_0, D_0) \\
&\quad + E[z_i \mid y_i, F_0, D_0]E[z_i \mid y_i, F_0, D_0]^T \\
&= N\big(I_s - F_0^T(\Sigma_0)^{-1}F_0\big) \\
&\quad + F_0^T(\Sigma_0)^{-1}Y^T Y(\Sigma_0)^{-1}F_0,
\end{aligned}
$$

Hence,

$$Q(F, D; F_0, D_0) = -\frac{(n+s)N}{2}\log(2\pi) - \frac{N}{2}\log\det D - \frac{1}{2}\operatorname{Tr}(W) \\ - \frac{1}{2}\operatorname{Tr}\Big(D^{-1}\big(Y^TY - 2FV + FWF^T\big)\Big),$$

where:

$$W = \sum_{i=1}^{N} E\big[z_i z_i^T \mid y_i, F_0, D_0\big]$$

and

$$V = \sum_{i=1}^{N} E\big[z_i \mid y_i, F_0, D_0\big] y_i^T = F_0^T(\Sigma_0)^{-1}Y^TY$$

3.2 Maximization step

Since $Q(F, D; F_0, D_0)$ is separable with respect to rows of F, we can solve the least square problem for each row of F. Having F_1, we have

$$D_1 = \frac{1}{N}\operatorname{diag}\big(\operatorname{diag}\big(Y^TY - 2F_1V + F_1WF_1^T\big)\big),$$

Keep iterating the algorithm until convergence.