SCALABLE STATISTICAL MODELS FOR PERSONALIZED RECOMMENDATION: RANDOM SLOPES AND LATENT STRUCTURE

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Disha Ghandwani
July 2025

© Copyright by Disha Ghandwani 2025
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Trevor Hastie) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Art Owen)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Jonathan Taylor)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Robert Tibshirani)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Zihuai He)

Approved for the Stanford University Committee on Graduate Studies

Preface

Recommender systems have become foundational to modern digital platforms, guiding users toward relevant content across a range of domains including e-commerce, entertainment, and social media. Central to the design of such systems is the challenge of accurately modeling user preferences and item characteristics to provide personalized recommendations. Statistical modeling offers a principled and interpretable approach to this task, particularly in settings where understanding user behavior is as important as achieving high predictive accuracy. However, with the ever-increasing scale of data—often involving millions of users and items—ensuring computational efficiency becomes critical.

This thesis is motivated by the need for scalable and interpretable statistical methods in the context of recommender systems. The first part of the work addresses this need by developing efficient estimation techniques for crossed random effects models with random slopes. These models extend classical mixed effects frameworks by allowing for individual-specific variation in the effects of covariates, thereby capturing nuanced patterns of user-item interaction. To overcome the computational bottlenecks associated with traditional estimation techniques, we propose scalable algorithms based on method of moments and variational inference. These methods enable estimation at the scale required for real-world applications while preserving the flexibility and interpretability of the underlying models. The proposed approaches are validated through simulations and experiments on large-scale recommendation datasets.

The second part of the thesis focuses on factor analysis as a means of modeling latent structure in user-item interaction data. Factor models provide a low-rank approximation to the high-dimensional covariance structure, uncovering hidden traits such as preference for genre, complexity, or popularity that influence user behavior. Furthermore, we propose hybrid models that integrate factor analysis with crossed random effects, yielding a flexible framework capable of capturing both latent dimensions and observed heterogeneity. This integration strikes a balance between scalability and interpretability, offering a robust modeling paradigm for complex and sparse recommendation data.

Together, the methods developed in this thesis aim to advance the statistical foundations of recommender systems by providing tools that are both theoretically sound and practically scalable.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Trevor Hastie, for his unwavering support and guidance throughout my Ph.D. journey. His mentorship, both academic and personal, has been invaluable, and I could not have asked for a more supportive advisor—especially during the challenging moments of my life.

I am also sincerely thankful to Art Owen, Jonathan Taylor, and Robert Tibshirani for serving as my thesis readers and for providing thoughtful and constructive feedback on my work. I am thankful to Zihuai He for kindly serving as the chair of my thesis defense.

During my time at Stanford, I have had the privilege of forming lasting friendships that have supported me both personally and professionally. I am especially grateful to Elena, Paula, Julie, Aditya, and Rahul for the engaging conversations, their constant motivation, and the intellectually stimulating discussions that enriched my research. I also wish to thank my seniors Swarnadip, Prathyusha, Samyak, Rupak, Ankita, Koushiki, and Alex and for their encouragement and for creating a sense of family away from home. I am thankful to Susie for always making herself available to assist with various matters, despite her busy schedule.

I owe immense gratitude to my parents and siblings for their constant love, encouragement, and the many sacrifices they have made to help me pursue my goals. Their belief in me has been a source of strength throughout this journey. I would also like to thank the CAPS team at Stanford for their compassionate support during a difficult period of my Ph.D.

Finally, I am especially grateful to Aditya and Paula, whose care and support helped me through a significant personal crisis. Their presence played a crucial role in helping me reach this point, and I am truly indebted to them for making it possible for me to complete and defend this thesis.

Contents

Preface iv

Acknowledgments v

1 Introduction 1

2 Crossed random effects model with random slopes 3

2.1 Introduction . . . 4

2.1.1 Paper Outline . . . 8

2.2 Estimation of $\boldsymbol{\beta}$ by GLS when the covariance parameters are known . . . 8

2.2.1 Vanilla Backfitting . . . 9

2.2.2 Backfitting with clubbing . . . 10

2.2.3 An efficient way to implement clubbing . . . 11

2.3 Estimation of covariance parameters using the method of moments . . . 12

2.3.1 Σ_a and Σ_b are diagonal matrices. . . . 13

2.3.2 Σ_a and Σ_b are unstructured . . . 14

2.4 Likelihood approaches for estimating all the parameters . . . 15

2.4.1 Maximum likelihood estimation . . . 15

2.4.2 Expectation Maximization Algorithm . . . 15

2.4.3 Variational EM Algorithm . . . 16

2.4.4 Connection of variational EM with backfitting . . . 19

2.4.5 Estimation of covariance of $\hat{\boldsymbol{\beta}}$ . . . 20

2.5 Results . . . 21

2.5.1 Simulations . . . 21

2.5.2 Application of crossed random effects model on 'Stitch Fix" data . . . 26

2.5.3 Naivety and inefficiencies with respect to OLS . . . 30

2.5.4 Fixed effects for Stitch Fix data . . . 31

2.5.5 Prediction error . . . 35

2.5.6 Application of crossed random effects model on Movie lens data . . . 36

2.6 Conclusion . . . 37

3 Scalable recommender system based on factor analysis **38**
3.1 Introduction . . . 38
3.2 Related work . . . 39
3.2.1 Crossed random effects model . . . 40
3.3 Factor analysis . . . 41
3.3.1 Introduction to factor analysis . . . 41
3.3.2 Use of the factor model in recommender systems . . . 43
3.4 EM algorithm . . . 45
3.4.1 E-step . . . 45
3.4.2 M-step . . . 46
3.5 Recommendation on Movie Lens . . . 48
3.6 Variational EM algorithm . . . 51
3.6.1 Variational E-step . . . 54
3.6.2 M-step . . . 55
3.7 Conclusion . . . 56

A Some proofs **58**
A.0.1 Proof of Theorem 1 . . . 58
A.0.2 Consistency of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ . . . 59
A.0.3 Method of Moments . . . 60
A.0.4 Proof of (2.26) . . . 61
A.0.5 Simplication of covariance of $\hat{\boldsymbol{\beta}}$ . . . 64
A.0.6 Proof of Lemma 1 . . . 65

B Results under general setup **67**

List of Tables

2.1 Each column represents the estimated fixed effects for the *Stitch Fix* data under different random effect regimes: OLS (no random effects), crossed random effects model with random intercepts, and two instances of the crossed random effects model with random intercepts as well as random slopes. "Match" is a continuous variable taking values in the range 0 to 1 representing the prediction of the probability that the client will keep the item purchased (based on an independent baseline model). The indicator variables named "client edgy", "client boho", "item edgy", "item boho" refer to fashion styles that describe some items and some client's preferences. The asterisks used to indicate significance follow the standard R convention, where * represents significance at the 5% level ($p \leq 0.05$), ** at the 1% level ($p \leq 0.01$), and *** at the 0.1% level ($p \leq 0.001$). 34
2.2 Prediction error for different models among different categories 35
2.3 Prediction error of ratings in MovieLens data 36

3.1 Prediction accuracy (MSE) of different models on the MovieLens dataset. The red value highlights the model achieving the best prediction accuracy for each split. Each split represents a different training and test split of the MovieLens dataset. 51

List of Figures

2.1 In the left plot, we illustrate a model with random intercepts and a fixed slope for the quantitative variable "match". In the right plot, we have random intercepts as well as random slopes for "match". The blue line represents the fixed intercept and slope in both plots. 6

2.2 Diagonal case: Comparison of estimation time and number of iterations by backfitting, clubbed backfitting, variational EM, and clubbed variational EM (four proposed algorithms) and "lmer"(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer". 23

2.3 Diagonal case: Comparison of accuracy in estimating Σ_a and Σ_b based on KL divergence and Frobenius norm by the method of moments (proposed algorithm) and lmer(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer". 25

2.4 Diagonal case: Comparison of accuracy in estimating $\boldsymbol{\beta}$ and σ_e^2 by method of moments (proposed algorithm) and lmer (existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer". The errors in estimating $\boldsymbol{\beta}$ are close to each other for the illustrated algorithms, which leads the lines in the plot to coincide. 26

2.5 Non-diagonal case: Comparison of accuracy in estimating Σ_a and Σ_b based on KL divergence and Frobenius norm by variational EM (proposed algorithm) and lmer(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to large computational time for "lmer". 27

2.6 Non diagonal case: Comparison of accuracy in estimating $\boldsymbol{\beta}$ and σ_e^2 by variational EM (proposed algorithm) and lmer(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer". The errors in estimating $\boldsymbol{\beta}$ are close to each other for the illustrated algorithms, which leads the lines in the plot to coincide. 28

2.7 Non-diagonal case: Comparison of estimation time by backfitting, clubbed backfitting, variational EM, and clubbed variational EM (four proposed algorithms) and "lmer"(existing algorithm). 29
2.8 Non-diagonal case: Comparison of number of iterations by backfitting, clubbed backfitting, variational EM, and clubbed variational EM (four proposed algorithms) and "lmer"(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer". 30
2.9 Naivety and efficiency of β_{OLS} with respect to $\beta_{\text{GLS},1}$ 32
2.10 Naivety and efficiency of β_{OLS} with respect to $\beta_{\text{GLS},2}$ 33

3.1 Histogram of age for females and males 48
3.2 Frequency of genre of movies 49
3.3 Visualisation of prediction accuracy of HardImpute algorithm and predictions by Model 5. The red curve shows a smoothed fit of the predicted ratings against the true ratings, while the blue line represents the ideal reference line $x = y$. The prediction accuracy of the two models is visually very similar, with Model 5 having a slight edge. 52
3.4 Violin plot showing the distribution of predictions from Model 5 at each level of the outcome variable (True Rating). The plot shows that the mean predicted rating conditional on the true rating steadily increases with the true rating, and that predictions are approximately normally distributed around the true rating. 53
3.5 This figure examines how the latent factor estimated by Model 5 relates to the covariates Age and Gender. The scatter plot displays the joint relationship between the hidden factor and Age, showing no clear dependence. The histogram shows the conditional distribution of the factor by gender, revealing nearly identical distributions for both groups. Together, this suggests that the hidden factor captures latent user characteristics distinct from Age and Gender. 54

B.1 General setup: Comparison of accuracy in estimating $\boldsymbol{\beta}$ and σ_e^2 by variational EM (proposed algorithm) and lmer(existing algorithm). The errors in estimating $\boldsymbol{\beta}$ are close to each other for the illustrated algorithms, which leads the lines in the plot to coincide. 68
B.2 General setup: Comparison of accuracy in estimating Σ_a and Σ_b based on KL divergence and Frobenius norm by variational EM (proposed algorithm) and lmer(existing algorithm). 69
B.3 General setup: Comparison of estimation time by backfitting, clubbed backfitting, variational EM, and clubbed variational EM (four proposed algorithms) and "lmer"(existing algorithm). 70

Chapter 1

Introduction

In today's digital ecosystem, personalized recommendations are at the heart of user engagement. Platforms such as Netflix, YouTube, Amazon, and Stitch Fix rely on recommender systems to filter vast catalogs of content and deliver tailored suggestions that match users' preferences. As these platforms scale, two fundamental challenges emerge: data sparsity and computational efficiency. Users typically engage with only a small subset of available items, leading to a sparse matrix of observed interactions. Additionally, the missingness is irregular—different users rate different numbers of items, with no consistent pattern. Recommender systems must operate on this incomplete data, infer preferences, and do so at a scale where millions of users and products are involved. This creates a pressing need for models that are both statistically sound and computationally scalable.

This thesis explores the use of crossed random effects models and factor analysis to develop scalable and interpretable recommender systems capable of handling such sparse and massive datasets. We consider a common setting in which R clients rate C items, but only $N \ll R \times C$ of the ratings are observed. Along with the rating information, we have access to covariates for both users and items—such as a client's age, gender, or purchase history, and item features like genre, material, or duration. These covariates provide valuable context and are used to improve rating predictions for unobserved (user, item) pairs.

To capture correlations in the data—such as the tendency of some items to receive higher ratings overall, or certain users to rate items more generously—we begin with crossed random effects models with random intercepts for users and items. These models are attractive for their interpretability and ability to account for nested structure in the data. However, traditional fitting methods, such as maximum likelihood estimation via standard mixed-effects modeling software, scale poorly with the number of users and items, often requiring computation on the order of $(R + C)^3$. This limitation has motivated recent work on more efficient algorithms. Notably, Ghosh et al. [2022a] and others have developed moment-based estimators and backfitting algorithms that reduce computational complexity to linear in N, making crossed random effects viable for large-scale recommendation

tasks.

Building on these developments, we extend the model to incorporate random slopes, allowing the effect of covariates to vary across users or items. This added flexibility captures heterogeneity in user preferences that fixed effects alone cannot explain. For example, while most users might rate romantic comedies positively, some may rate them significantly higher or lower based on individual taste. By modeling this heterogeneity through user-specific or item-specific slopes, we achieve finer personalization.

Further, we explore factor analysis models, which can be viewed as a generalization of random slope models. Factor models aim to learn low-dimensional latent representations of users and items, and predict ratings through their interactions. These models are especially well-suited to settings with limited covariate information, as they infer latent structure directly from observed ratings. We investigate scalable parameter estimation in these models using Expectation-Maximization (EM) algorithms, adapted to sparse and incomplete data matrices commonly found in recommender system applications.

Throughout the thesis, we apply these methods to real-world datasets, including those from the online clothing retailer Stitch Fix and the popular MovieLens platform. These applications serve to both illustrate the practicality of the proposed methods and benchmark their predictive performance. For instance, we show that the crossed random effects model with random intercepts significantly outperforms ordinary least squares regression, increasing the R^2 on Stitch Fix data from 4.61% to 18.42%. Such improvements demonstrate the value of incorporating structured random effects in recommender systems.

In summary, this thesis makes the following contributions:

- Develops scalable algorithms for fitting crossed random effects models with both random intercepts and slopes.
- Extends the framework to factor analysis for learning latent user-item interactions.
- Demonstrates practical relevance through applications to real-world recommendation data, highlighting gains in prediction accuracy and interpretability.

By addressing the dual challenges of sparsity and scalability, we aim to contribute effective modeling tools for the next generation of recommender systems.

Chapter 2

Scalable solution to crossed random effects model with random slopes

Abstract

The crossed random effects model is widely used, finding applications in various fields such as longitudinal studies, e-commerce, and recommender systems, among others. However, these models encounter scalability challenges, as the computational time for standard algorithms grows superlinearly with the number N of observations in the data set, commonly $O(N^{3/2})$ or worse. Recent published works present scalable methods for crossed random effects in linear models and some generalized linear models, but those methods only allow for random intercepts. In this paper, we devise scalable algorithms for models that include random slopes. This addition brings substantial difficulty in estimating the random-effect covariance matrices in a scalable way. We address this issue by using a variational EM algorithm. Our proposed approach accommodates both diagonal covariance matrices and cases where no structure is assumed—a scenario common in fields such as psychology and neuroscience. In simulations, the proposed method is substantially faster than standard methods for large N. It is also more efficient than ordinary least squares which has a problem of greatly underestimating the sampling uncertainty in parameter estimates. We illustrate the new method on a *MovieLens* data set, as well as a large data set (five million observations) from the online retailer *Stitch Fix*.

2.1 Introduction

In the era of *over the top* (OTT) platforms like Netflix, where customer acquisition is booming and a vast array of titles are hosted, the need for computationally efficient recommender systems has become crucial. In such scenarios, it is common for different clients to rate varying numbers of items, which gives a haphazard missingness pattern. We deal with one such problem where R clients rate C items, and it is usually the case that only $N \ll (R \times C)$ of the ratings are observed. We consider the situation where we have access to various characteristics of each (item, client) pair as well as the client's rating of the item. For instance, with the online clothing retailer we have information such as the client's age, sex, and purchase history, as well as item features such as material and clothing style. For the movie data, we have genre, language, and duration for the online content.

Our motivating data set comes from the online clothing retailer Stitch Fix. Quoting from Ghosh et al. [2022b]:

> Stitch Fix is an online personal styling service. One of their business models involves sending customers a sample of clothing items. The customer may keep and purchase any of those items and return the others. They have provided us with some of their client ratings data. That data was anonymized, void of personally identifying information, and as a sample it does not reflect their total numbers of clients or items at the time they provided it. It is also from 2015. While it does not describe their current business, it is a valuable data set for illustrative purposes.

We present results using these data in Section 2.5.2, and we use the problem setup throughout this paper to explain our choices. The goal is to model clients' ratings of items. Let y_{ij} represent the rating provided by client i for item j, and $\mathbf{x}_{ij}$ (typically a vector) the covariate information for the specific client-item pair (i, j). In such settings, ordinary least squares is not advisable as the ratings are not independent. Instead, we expect some positive correlation between ratings by different clients on the same item, because some items are simply more popular than others in ways that cannot always be attributed to $\mathbf{x}_{ij}$. We can similarly expect some positive correlation between ratings by the same client on different items because clients vary in their preferences. To account for such covariance structure, crossed random-effects models are employed. These models are designed to capture the dependencies present in the ratings provided by the same client or for the same item. One notable drawback of such models is their limited scalability. As the data set grows in size, the computational burden associated with fitting crossed random effects models becomes challenging and inefficient. Ghosh et al. [2022a] considered the crossed random-effects model with random intercept terms for clients and items, given by

$$y_{ij} = \beta_0 + a_i + b_j + \mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + \varepsilon_{ij}, \qquad \forall i \in \{1, \cdots, R\}, \quad j \in \{1, \cdots, C\}. \tag{2.1}$$

Here, the random effects are $a_i \in \mathbb{R}$, $b_j \in \mathbb{R}$ and an error $\varepsilon_{ij} \in \mathbb{R}$. The fixed effects are the regression parameters $\boldsymbol{\beta} \in \mathbb{R}^p$ and the intercept $\beta_0 \in \mathbb{R}$. They assume that $a_i \sim \mathcal{N}(0, \sigma_a^2)$, $b_j \sim \mathcal{N}(0, \sigma_b^2)$, and $\varepsilon_{ij} \sim \mathcal{N}(0, \sigma_e^2)$ are all independent. Under this model,

$$\mathrm{Cov}(y_{ij}, y_{ij'}) = \sigma_a^2 \text{ and } \mathrm{Cov}(y_{ij}, y_{i'j}) = \sigma_b^2 \text{ for } i \neq i', j \neq j'.$$

Standard approaches for fitting such models (including "lmer" function of "lme4" library in R) come with a computational burden proportionate to $(R + C)^3$, where R and C are the number of clients and items respectively. However, due to the relationship $RC \geq N$, this computational cost can be expressed as $O(N^{3/2})$. To address this limitation, Gao and Owen [2020] introduced a moment-based method for estimating variance parameters in the intercept model (3.1) with a computational cost that is at most linear. More recently, Ghosh et al. [2022a] developed a backfitting algorithm inspired by the work of Buja et al. [1989] that achieves linear computational cost in terms of N. In contrast, the plain Gibbs sampler is not scalable, as it exhibits a computational cost of $O(N^{3/2})$, as shown by Gao and Owen [2017]. They also show that many other Bayesian approaches do not work well in crossed random effects settings. Recent work has shown that collapsed Gibbs sampling can be made to scale well in crossed random effects settings. See Papaspiliopoulos et al. [2020] and Ghosh and Zhong [2021]. Ghosh [2022] presents an insightful connection between the convergence of backfitting and collapsed Gibbs. These works do not include the random slopes that we focus on here.

During the process of model fitting, Ghosh et al. [2022a] also produce (posterior or BLUP) estimates for $(a_i)_{i=1}^R$ and $(b_j)_{j=1}^C$, which can be used for making predictions on test data. This implies that the crossed random effects model can serve as a scalable recommender system. To assess its prediction accuracy, we measured the performance of the crossed random effects model with a random intercept on Stitch Fix data using a random train-test split. In comparison to ordinary least squares, which yielded an R^2 of 4.61%, the crossed random effects model (3.1) achieved an impressive R^2 of 18.42 %. This significant improvement in R^2 shows the efficiency gain from crossed random effects models. We also demonstrate the gains achieved on the *MovieLens* data.

These developments motivated us to develop an efficient and scalable implementation for crossed-random effects with both random intercepts and slopes.

$$y_{ij} = \beta_0 + a_i + b_j + \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \tilde{\boldsymbol{a}}_i + \tilde{\boldsymbol{b}}_j) + \varepsilon_{ij}, \quad \forall i \in \{1, \cdots, R\}, \quad j \in \{1, \cdots, C\}, \tag{2.2}$$

where we have augmented (3.1) with random slopes $\tilde{\boldsymbol{a}}_i \in \mathbb{R}^p$ and $\tilde{\boldsymbol{b}}_j \in \mathbb{R}^p$. We assume that $\tilde{\boldsymbol{a}}_i \sim \mathcal{N}_p(0, \tilde{\Sigma}_a)$ and $\tilde{\boldsymbol{b}}_j \sim \mathcal{N}_p(0, \tilde{\Sigma}_b)$. We further assume that $(a_i, \tilde{\boldsymbol{a}}_i)$ follows a multivariate gaussian distribution and is independent of $(b_j, \tilde{\boldsymbol{b}}_j)$. The crossed random effects model with random slopes helps us capture the variability of the effect size of covariates among different clients and items. Figure 2.1 provides a visual representation of the model with random intercept and random slopes.

The first plot represents a random effects model with a random intercept for clients and a fixed slope for the quantitative feature "match"; the second plot has a random intercept as well as a random slope for "match". We can simpify the notation in (3.2) by including 1 as the first variable in $\mathbf{x}_{ij}$ to

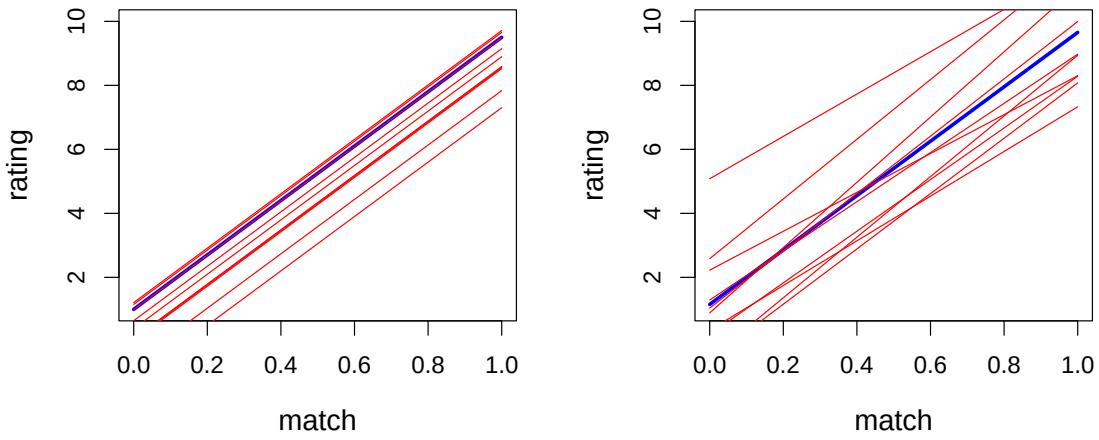


Figure 2.1: In the left plot, we illustrate a model with random intercepts and a fixed slope for the quantitative variable "match". In the right plot, we have random intercepts as well as random slopes for "match". The blue line represents the fixed intercept and slope in both plots.

accommodate the intercept:

$$y_{ij} = \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \boldsymbol{a}_i + \boldsymbol{b}_j) + \varepsilon_{ij}, \qquad \forall i \in \{1, \cdots, R\}, \quad j \in \{1, \cdots, C\}. \tag{2.3}$$

Now $\mathbf{x}_{ij} \in \mathbb{R}^{p+1}$, and we assume that $\boldsymbol{a}_i \sim \mathcal{N}_{p+1}(0, \Sigma_a)$, $\boldsymbol{b}_j \sim \mathcal{N}_{p+1}(0, \Sigma_b)$, and $\varepsilon_{ij} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma_e^2)$ are all independent. Hence the random effects include both intercepts and slopes.

Remark 1. *For notational simplicity, we have included random slopes for all covariates when describing our method. In practice, several variations are attractive and easily accommodated*[1]*:*

- *We may want random slopes for just a subset of the covariates, depending on their nature.*
- *We may want different subsets of covariates having random slopes for clients than for items.*
- *Some covariates are specific to the client (e.g., age or gender) and do not change with different items. Therefore, client random slopes do not make sense and should not be included. However, item random slopes can still be used. The reverse applies to item-specific covariates*

[1]The R and Python code provided at https://github.com/dishaghandwani/randomslopes handles these variations.

The goal is to estimate the fixed effect $\boldsymbol{\beta}$ and covariance parameters Σ_a, Σ_b, and σ_e^2. When $\boldsymbol{a}_i, \boldsymbol{b}_j$ and ε_{ij} are all Gaussian in (2.3), the likelihood for $(\boldsymbol{\beta}, \Sigma_a, \Sigma_b, \sigma_e^2)$ is

$$\frac{1}{\sqrt{(2\pi)^N |\mathcal{V}|}} \exp\left(-\frac{1}{2}(y - X\boldsymbol{\beta})^\mathsf{T} \mathcal{V}^{-1}(y - X\boldsymbol{\beta})\right). \tag{2.4}$$

Here, the vector Y denotes all of the y_{ij} placed into a vector in $\mathbb{R}^N$ and X denotes $\mathbf{x}_{ij}$ stacked compatibly into a matrix in $\mathbb{R}^{N\times(p+1)}$. $\mathcal{V} \in \mathbb{R}^{N\times N}$ represents the covariance matrix of the N-vector Y, which is a somewhat complex function of $(\boldsymbol{\beta}, \Sigma_a, \Sigma_b, \sigma_e^2)$:

$$\mathcal{V} = \left(X\Sigma_a X^\mathsf{T}\right) \bullet \left(Z_a Z_a^\mathsf{T}\right) + \left(X\Sigma_b X^\mathsf{T}\right) \bullet \left(Z_b Z_b^\mathsf{T}\right) + \sigma_e^2 I_N, \tag{2.5}$$

where $Z_a \in \{0,1\}^{N\times R}$ denotes the indicator (one-hot encoding) matrix for clients, $Z_b \in \{0,1\}^{N\times C}$ denotes the indicator matrix for items and $A \bullet B$ denotes elementwise (Hadamard) product of matrix A and B. Let data point t_1 refer to the client-item pair (i_1, j_1) and data point t_2 refer to the pair (i_2, j_2), then

$$\mathcal{V}_{t_1,t_2} = \begin{cases} 0, & \text{if } i_1 \neq i_2 \text{ and } j_1 \neq j_2, \\ x_{i_1 j}^\mathsf{T} \Sigma_b x_{i_2 j}, & \text{if } i_1 \neq i_2 \text{ and } j_1 = j_2 = j, \\ x_{i j_1}^\mathsf{T} \Sigma_a x_{i j_2}, & \text{if } i_1 = i_2 = i \text{ and } j_1 \neq j_2, \\ \mathbf{x}_{ij}^\mathsf{T} \Sigma_a \mathbf{x}_{ij}^\mathsf{T} + \mathbf{x}_{ij}^\mathsf{T} \Sigma_b \mathbf{x}_{ij} + \sigma_e^2, & \text{if } (i_1, j_1) = (i_2, j_2) = (i, j). \end{cases} \tag{2.6}$$

The bottom term in (2.6) represents a diagonal term in $\mathcal{V}$, i.e. the variance of y_{ij}. The maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$ is also the generalized least squares (GLS) estimate. Hence if $\mathcal{V}$ were known (i.e. we have estimates for its ingredients), we could directly estimate $\boldsymbol{\beta}$ by GLS. However, since $\mathcal{V}$ is generally dense and not usefully structured, obtaining $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ naively by GLS would be computationally prohibitive ($\mathcal{O}(N^3)$).

- Our first approach tackles this GLS problem using method-of-moment estimates for the covariance parameters. In Section 2.2, we describe an efficient backfitting procedure for estimating $\boldsymbol{\beta}$ by GLS if the the covariance parameters are known. We also provide consistent estimates of the covariance parameters using method-of-moments, as described in Section 2.3. Both of the tasks can be accomplished in $\mathcal{O}(N)$ time.

The drawback of this first approach is that when p is large and the covariance matrices are non-diagonal, the method of moments may not produce accurate estimates for the covariance parameters. Direct maximum likelihood estimation of all the parameters using (3.14) seems out of the question.

- In our second approach, we maximize an approximation to the likelihood (3.14) using *variational EM* to estimate the fixed effects and covariance parameters simultaneously. It helps us in estimating covariance matrices precisely even when they are unstructured or p is sufficiently

large. Another advantage of this approach is its computational efficiency, with each step at most $\mathcal{O}(N)$.

The variational EM algorithm is closely related to the backfitting algorithm. Variational EM has been previously explored for fast and accurate estimation of non-nested binomial hierarchical models by Goplerud [2022]. We speed up the convergence rate for both of the approaches — backfitting and variational EM — using a technique referred to as "clubbing" discussed in Section 2.2.

2.1.1 Paper Outline

- In Section 2.2, we discuss the backfitting approach with and without clubbing.
- In Section 2.3, we discuss the estimation of covariance parameters using the method of moments in two scenarios: when the covariance matrices, Σ_a and Σ_b are diagonal, and when they are non-diagonal.
- In Section 2.4, we discuss the application of variational EM in estimating fixed as well as covariance parameters using an approximate maximum likelihood approach.
- In Section 2.5, we show the performance of these methods using both simulated and real data.

2.2 Estimation of $\boldsymbol{\beta}$ by GLS when the covariance parameters are known

If the covariance parameters are known, the aim is to obtain $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ in $\mathcal{O}(N)$ time. We know that the GLS estimate of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}}_{\text{GLS}} = \arg\min_{\boldsymbol{\beta}} (Y - X\boldsymbol{\beta})^{\mathsf{T}} \mathcal{V}^{-1} (Y - X\boldsymbol{\beta}) = (X^{\mathsf{T}} \mathcal{V}^{-1} X)^{-1} X^{\mathsf{T}} \mathcal{V}^{-1} Y. \tag{2.7}$$

But, we couldn't use the above formula directly to obtain $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ as computing $\mathcal{V}^{-1}$ would take at least $\mathcal{O}(N^{3/2})$ time in the above setting (and more generally $\mathcal{O}(N^3)$). The theorem below helps us in setting up the motivation for the $\mathcal{O}(N)$) backfitting algorithm.

Theorem 1. *Consider the solutions to the following penalized least-squares problem*

$$\min_{\boldsymbol{\beta}, \boldsymbol{a}_i, \boldsymbol{b}_j} \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{z_{ij}(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \boldsymbol{a}_i + \boldsymbol{b}_j))^2}{\sigma_e^2} + \sum_{i=1}^{R} \boldsymbol{a}_i^{\mathsf{T}} \Sigma_a^{-1} \boldsymbol{a}_i + \sum_{j=1}^{C} \boldsymbol{b}_j^{\mathsf{T}} \Sigma_b^{-1} \boldsymbol{b}_j. \tag{2.8}$$

Then $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{\text{GLS}}$ and the $\{\hat{\boldsymbol{a}}_i\}_{i=1}^{R}$ and $\{\hat{\boldsymbol{b}}_j\}_{j=1}^{C}$ are the best linear unbiased prediction (BLUP) estimates of the random effects if the covariance parameters Σ_b, Σ_b and σ_e^2 are fixed and known.

Here we have introduced a binary *masking* matrix $Z \in R \times C$ for ease of notation: $z_{ij} = 1$ if the rating y_{ij} for the (i, j) client-item pair is observed, and $z_{ij} = 0$ otherwise.

Remark 2. *In practice, we may also observe multiple observations for client-item pairs, which entails additional notation. Our algorithms are able to accomodate this seamlessly, as the code mostly involves grouping the observations by items or clients. In order to keep the notation simple, we restrict our presentation to the binary annototation of z_{ij}. Another important thing to note here is that, although the first term in (2.8) appears as a summation of RC values, it is just for illustrative purpose. The summation is actually performed only on observed ratings.*

The theorem was originally given by Robinson [1991]; we provide an alternative proof in Appendix A.0.1. If the covariance parameters are known or somehow estimated, the fixed-effect parameter can be estimated by backfitting algorithms (block-coordinate descent) which minimize (2.8). We estimate $\boldsymbol{\beta}$ using backfitting with and without clubbing similar to Ghosh et al. [2022a] and Ghosh et al. [2022b].

2.2.1 Vanilla Backfitting

The objective function (2.8) treats the three groups $\{\boldsymbol{a}_i\}_{i=1}^R$, $\{\boldsymbol{b}_j\}_{j=1}^C$, and $\boldsymbol{\beta}$ as parameters to be estimated by least squares, with $\{\boldsymbol{a}_i\}_{i=1}^R$ and $\{\boldsymbol{b}_j\}_{j=1}^C$ regularized via generalized ridge penalties. The idea of backfitting is to cycle through the estimation of the parameters in each group, keeping the parameters fixed in the remaining groups, i.e., a block coordinate descent. Minimizing (2.8) is equivalent to minimizing

$$\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \boldsymbol{a}_i + \boldsymbol{b}_j))^2 + \sum_{i=1}^{R} \boldsymbol{a}_i^{\mathsf{T}}\Lambda\boldsymbol{a}_i + \sum_{j=1}^{C} \boldsymbol{b}_j^{\mathsf{T}}\Gamma\boldsymbol{b}_j, \tag{2.9}$$

where $\Lambda = \sigma_e^2\Sigma_a^{-1}$ and $\Gamma = \sigma_e^2\Sigma_b^{-1}$ are the precision matrices corresponding to the two random effects. Thus, the fit for $\boldsymbol{a}_i$ keeping $\{\boldsymbol{b}_j\}_{j=1}^C$ and $\boldsymbol{\beta}$ fixed is given by

$$\hat{\boldsymbol{a}}_i = \left(\sum_{j=1}^{C} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}} + \Lambda\right)^{-1}\left(\sum_{j=1}^{C} z_{ij}\mathbf{x}_{ij}\Big(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \boldsymbol{b}_j)\Big)\right). \tag{2.10}$$

Similarly, fit for $\boldsymbol{b}_j$ keeping $\{\boldsymbol{a}_i\}_{i=1}^R$ and $\boldsymbol{\beta}$ fixed is given by

$$\hat{\boldsymbol{b}}_j = \left(\sum_{i=1}^{R} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}} + \Gamma\right)^{-1}\left(\sum_{i=1}^{R} z_{ij}\mathbf{x}_{ij}\Big(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \boldsymbol{a}_i)\Big)\right). \tag{2.11}$$

The fit for $\boldsymbol{\beta}$ given $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ is given by

$$\hat{\boldsymbol{\beta}} = \Big(\sum_{j=1}^{C}\sum_{i=1}^{R} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}}\Big)^{-1}\Big(\sum_{j=1}^{C}\sum_{i=1}^{R}\Big(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{a}_i + \boldsymbol{b}_j)\Big)z_{ij}\mathbf{x}_{ij}\Big). \tag{2.12}$$

We cycle through these equations until the fits converge.

Remark 3.

- *For each group of parameters, we form the partial residuals holding the others fixed. Then, we optimize with respect to that group. For the random effect parameters, the problem decouples, as each step in (2.10) and (2.11) indicate.*
- *Convergence is measured in terms of the fits of the terms, $\mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta}$, $\{\mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{a}_i\}$ and $\{\mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{b}_j\}$.*
- *Each step decreases the objective, and the algorithm converges to the solution to (2.8).*
- *Notice that in (2.10), for fixed i, we only visit terms $j \in \{1, 2, \ldots, C\}$ for which $z_{ij} = 1$. Hence, we visit in total N terms for all i. Likewise, this applies to (2.11) and (3.23). Hence the computational cost of obtaining all the above fits is $\mathcal{O}(N)$.*

2.2.2 Backfitting with clubbing

One of the shortcomings of the vanilla backfitting algorithm is that it can take a long time to converge. One of the reasons is that the solution has built-in constraints that are not visible to the individual steps. The theorem below provides two such constraints.

Theorem 2. *The solutions $\{\boldsymbol{a}_i\}_{i=1}^R$ and $\{\boldsymbol{b}_j\}_{j=1}^C$ of (2.8) satisfy $\sum_{i=1}^R \hat{\boldsymbol{a}}_i = 0$ and $\sum_{j=1}^C \hat{\boldsymbol{b}}_j = 0$.*

Proof. It is enough to prove $\sum_{i=1}^R \hat{\boldsymbol{a}}_i = 0$, the proof for $\sum_{j=1}^C \hat{\boldsymbol{b}}_j = 0$ is similar. Suppose $\sum_{i=1}^R \hat{\boldsymbol{a}}_i \neq 0$, define $\bar{\boldsymbol{a}} = \sum_{i=1}^R \hat{\boldsymbol{a}}_i/R$ and $\tilde{\boldsymbol{a}}_i = \hat{\boldsymbol{a}}_i - \bar{\boldsymbol{a}}$ $\forall i$. Then it is easy to show that

$$\sum_{i=1}^{R}\tilde{\boldsymbol{a}}_i^{\mathsf{T}}\Sigma_a^{-1}\tilde{\boldsymbol{a}}_i = \sum_{i=1}^{R}(\hat{\boldsymbol{a}}_i - \bar{\boldsymbol{a}})^{\mathsf{T}}\Sigma_a^{-1}(\hat{\boldsymbol{a}}_i - \bar{\boldsymbol{a}}) < \sum_{i=1}^{R}\hat{\boldsymbol{a}}_i^{\mathsf{T}}\Sigma_a^{-1}\hat{\boldsymbol{a}}_i,$$

unless $\bar{\boldsymbol{a}} = 0$. If we choose, $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} + \bar{\boldsymbol{a}}$, i.e., $\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{a}}_i = \hat{\boldsymbol{\beta}} + \hat{\boldsymbol{a}}_i$, then

$$\sum_{i=1}^{R}\sum_{j=1}^{C}\frac{z_{ij}(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{a}}_i + \boldsymbol{b}_j))^2}{\sigma_e^2} = \sum_{i=1}^{R}\sum_{j=1}^{C}\frac{z_{ij}(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{a}}_i + \boldsymbol{b}_j))^2}{\sigma_e^2}. \tag{2.13}$$

Thus,

$$\min_{\boldsymbol{\beta},\boldsymbol{b}_j} \sum_{i=1}^{R}\sum_{j=1}^{C} \frac{z_{ij}(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \tilde{\boldsymbol{a}}_i + \boldsymbol{b}_j))^2}{\sigma_e^2} + \sum_{i=1}^{R} \tilde{\boldsymbol{a}}_i^{\mathsf{T}} \Sigma_a^{-1} \tilde{\boldsymbol{a}}_i$$
$$< \min_{\boldsymbol{\beta},\boldsymbol{b}_j} \sum_{i=1}^{R}\sum_{j=1}^{C} \frac{z_{ij}(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \hat{\boldsymbol{a}}_i + \boldsymbol{b}_j))^2}{\sigma_e^2} + \sum_{i=1}^{R} \hat{\boldsymbol{a}}_i^{\mathsf{T}} \Sigma_a^{-1} \hat{\boldsymbol{a}}_i,$$

unless $\bar{\boldsymbol{a}} = 0$ which leads to contradiction. □

The theorem above shows that the constraint $\sum_{i=1}^{R} \hat{\boldsymbol{a}}_i = 0$ is automatically imposed if we minimize (2.8) simultaneously in terms of $\{\boldsymbol{a}_i\}_{i=1}^{R}$ and $\boldsymbol{\beta}$. Similarly, the constraint $\sum_{j=1}^{C} \hat{\boldsymbol{b}}_j = 0$ is automatically imposed if we minimize (2.8) simultaneously in terms of $\{\boldsymbol{b}_j\}_{j=1}^{C}$ and $\boldsymbol{\beta}$. We refer to this as "clubbing", following Ghosh et al. [2022a]. We also combine this technique with our variational EM approach in Section 2.4.3 to achieve fast convergence.

2.2.3 An efficient way to implement clubbing

We consider the case of estimating $\boldsymbol{\beta}$ and $\boldsymbol{a}_i$ with $\{\boldsymbol{b}_j\}_{j=1}^{C}$ fixed. This would be equivalent to minimizing the loss given by (2.8) keeping $\{\boldsymbol{b}_j\}$ fixed. Thus, if $r_{ij} = y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{b}_j$, then we

$$\underset{\boldsymbol{\beta},\{\boldsymbol{a}_i\}}{\text{minimize}} \sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\big(r_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \boldsymbol{a}_i)\big)^2 + \sum_{i=1}^{R} \boldsymbol{a}_i^{\mathsf{T}} \Lambda \boldsymbol{a}_i, \tag{2.14}$$

which would be equivalent to simultaneously solving the following equations:

$$\hat{\boldsymbol{a}}_i = \Big(\sum_{j=1}^{C} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}} + \Lambda\Big)^{-1}\Big(\sum_{j=1}^{C}\Big(r_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta}\Big) z_{ij}\mathbf{x}_{ij}\Big), \tag{2.15}$$

$$\hat{\boldsymbol{\beta}} = \Big(\sum_{j=1}^{C}\sum_{i=1}^{R} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}}\Big)^{-1}\Big(\sum_{i=1}^{R}\sum_{j=1}^{C}\Big(r_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{a}_i\Big) z_{ij}\mathbf{x}_{ij}\Big), \tag{2.16}$$

where $\Lambda = \sigma_e^2 \Sigma_a^{-1}$.
Let f_a denote the fit (N vector) corresponding to the terms $\{\mathbf{x}_{ij}^{\mathsf{T}}\hat{\boldsymbol{a}}_i\}$ and $X\hat{\boldsymbol{\beta}}$ be the N-vector of fits for the fixed effects. Then from (2.15), we can write

$$f_a = S_a(r - X\hat{\boldsymbol{\beta}}), \tag{2.17}$$

and from (2.16), we have

$$X^{\mathsf{T}}X\hat{\boldsymbol{\beta}} = X^{\mathsf{T}}(r - f_a). \tag{2.18}$$

Here, S_a is the implicit $N \times N$ linear operator that computes these fits, obtained by solving (2.15) for each i. We note that if the observations were ordered by groups of i, then S_a would be block-diagonal, with the ith block a generalized ridge operator matrix for producing the fits for the observations with that value of i. This also implies that S_a is always a symmetric matrix. Plugging (2.17) into (2.18), we have

$$X^\mathsf{T} X\hat{\boldsymbol{\beta}} = X^\mathsf{T}(r - S_a(r - X\hat{\boldsymbol{\beta}})). \tag{2.19}$$

Collecting terms, we have

$$X^\mathsf{T}\left(I - S_a\right) X\hat{\boldsymbol{\beta}} = X^\mathsf{T}\left(I - S_a\right) r, \tag{2.20}$$

and hence

$$\hat{\boldsymbol{\beta}} = \left[X^\mathsf{T}\left(I - S_a\right) X\right]^{-1} X^\mathsf{T}\left(I - S_a\right) r. \tag{2.21}$$

This means that we need to apply S_a to each of the columns of X, compute the fits, and take residuals, to produce $\tilde{X} = (I - S_a)X$, then

$$\hat{\boldsymbol{\beta}} = \left(\tilde{X}^\mathsf{T} X\right)^{-1} \tilde{X}^\mathsf{T} r \text{ (as } S_a \text{ is symmetric)}. \tag{2.22}$$

We plug this $\hat{\boldsymbol{\beta}}$ in equation (2.15) to get $\hat{\boldsymbol{a}}_i$ for each i. Now we repeat this process holding the $\{\boldsymbol{a}_i\}$ fixed at their latest values $\{\hat{\boldsymbol{a}}_i\}$, and estimate $\boldsymbol{\beta}$ and $\{\boldsymbol{b}_j\}$ together. We continue the above process till fits of the terms, $\mathbf{x}_{ij}^\mathsf{T}\boldsymbol{\beta}, \{\mathbf{x}_{ij}^\mathsf{T}\boldsymbol{a}_i\}$ and $\{\mathbf{x}_{ij}^\mathsf{T}\boldsymbol{b}_j\}$ converge.

The time complexity for each step of backfitting with or without clubbing is $\mathcal{O}(Np^2)$.

2.3 Estimation of covariance parameters using the method of moments

Method of moments (MoM) is a widely used approach for estimating covariance parameters, but for the most part has been limited to cases where there are a small number of scalar parameters. See, for example, Wu et al. [2012] and Gao and Owen [2017]. In our setting, we have a novel MoM approach for crossed random effects with covariance matrices of any dimension. Our MoM approach has algorithmic complexity linear in N, and hence is scalable.

We discuss the estimation of covariance parameters under two scenarios: when Σ_a and Σ_b are assumed to be diagonal, and when nothing is assumed.

2.3.1 Σ_a and Σ_b are diagonal matrices.

If Σ_a and Σ_b are diagonal matrices, we can obtain consistent estimates of Σ_a, Σ_b, and σ_e^2 using the method of moments. Our process involves solving equations of the form

$$\mathbb{E}\Bigg[\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij} r_{ij}^2 - \sum_{i=1}^{R} \frac{\left(\sum_{j=1}^{C} z_{ij} r_{ij} x_{ij}^{(s)}\right)^2}{\sum_{j=1}^{C} z_{ij} x_{ij(s)}^2} \Bigg| X\Bigg] = \sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij} \hat{r}_{ij}^2 - \sum_{i=1}^{R} \frac{\left(\sum_{j=1}^{C} z_{ij} \hat{r}_{ij} x_{ij}^{(s)}\right)^2}{\sum_{j=1}^{C} z_{ij} x_{ij(s)}^2}, \tag{2.23}$$

$$\mathbb{E}\Bigg[\sum_{j=1}^{C}\sum_{i=1}^{R} z_{ij} r_{ij}^2 - \sum_{j=1}^{C} \frac{\left(\sum_{i=1}^{R} z_{ij} r_{ij} x_{ij}^{(s)}\right)^2}{\sum_{i=1}^{R} z_{ij} x_{ij(s)}^2} \Bigg| X\Bigg] = \sum_{j=1}^{C}\sum_{i=1}^{R} z_{ij} \hat{r}_{ij}^2 - \sum_{j=1}^{C} \frac{\left(\sum_{i=1}^{R} z_{ij} \hat{r}_{ij} x_{ij}^{(s)}\right)^2}{\sum_{i=1}^{R} z_{ij} x_{ij(s)}^2}, \tag{2.24}$$

for all s $\in \{0, \cdots, p\}$, and

$$\mathbb{E}\Bigg[\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij} \left(r_{ij} - \bar{r}_{..}\right)^2 \Bigg| X\Bigg] = \sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij} \left(\hat{r}_{ij} - \bar{\hat{r}}_{..}\right)^2. \tag{2.25}$$

Remark 4. *Here, $x_{ij}^{(s)}$ represents s-th covariate of the (i,j) client-item pair, $r = Y - X\boldsymbol{\beta}$, and $\hat{r} = Y - X\hat{\boldsymbol{\beta}}_{\text{OLS}}$.*

These equations can be seen as a generalization of the moment equations considered by Ghosh et al. [2022a]; we give more insight into these equations in Appendix A.0.3. Using the matrix formulation in the appendix, we can show that

$$\begin{aligned}
&\mathbb{E}\Bigg[\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij} r_{ij}^2 - \sum_{i=1}^{R} \frac{\left(\sum_{j=1}^{C} z_{ij} r_{ij} x_{ij}^{(s)}\right)^2}{\sum_{j=1}^{C} z_{ij} x_{ij(s)}^2} \Bigg| X\Bigg] \\
&= \sum_{s'=0}^{p}\sum_{i=1}^{R} \left[\sum_{j=1}^{C} z_{ij} x_{ij}^{(s')^2} - \frac{\left(\sum_{j=1}^{C} z_{ij} x_{ij}^{(s)} x_{ij}^{(s')}\right)^2}{\sum_{j=1}^{C} z_{ij} x_{ij}^{(s)^2}}\right] \Sigma_{a,s's'} \\
&\quad + \sum_{s'=0}^{p}\sum_{i=1}^{R} \left[\sum_{j=1}^{C} z_{ij} x_{ij}^{(s')^2} - \frac{\sum_{j=1}^{C} z_{ij} x_{ij}^{(s)^2} x_{ij}^{(s')^2}}{\sum_{j=1}^{C} z_{ij} x_{ij}^{(s)^2}}\right] \Sigma_{b,s's'} + \sigma_e^2 (n - R).
\end{aligned} \tag{2.26}$$

Remark 5. *The proof is provided in Appendix A.0.3. The complexity to compute all the quantities*

above is $\mathcal{O}(Np^2)$. Once these quantities are computed, we solve equations (2.23), (2.24) and (2.25) to get the covariance parameters. Thus we obtain $2p+3$ equations involving $2p+3$ unknown variance parameters, which we can solve to obtain MoM estimates for the covariance parameters.

Under the regularity conditions mentions in the theorems below, the covariance parameters estimated using the suggested MoM approach are consistent. We have implemented the MoM approach for the case when the covariance matrices are diagonal in Section 2.5.1. The approach is found to provide empirically consistent estimates for the covariance parameters which are then used to estimate the fixed parameter using backfitting.

Theorem 3. *Let $X_r^{(a)}(X_c^{(b)})$ be the observations arranged in row (column) blocks such that each block has a fixed level for the first (second) factor. Also assume that $\max_{1\le r\le R}\frac{\lambda_{\max}(X_r^{(a)}\Sigma_a X_r^{(a)\mathsf{T}})}{N}\to 0$, $\max_{1\le c\le C}\frac{\lambda_{\max}(X_c^{(b)}\Sigma_b X_c^{(b)\mathsf{T}})}{N}\to 0$ and $\lambda_{min}(X^\mathsf{T}X)/N\ge c$ for some $c>0$, as $N\to\infty$. Then $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is consistent.*

Theorem 4. *The method of moment estimates obtained using the above method is consistent if the fourth moments for $\mathbf{x}_{ij}, i\in\{1,\cdots,R\}$, $j\in\{1,\cdots,C\}$ are uniformly bounded and $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is consistent.*

2.3.2 Σ_a and Σ_b are unstructured

If there are no constraints on the structure of Σ_a and Σ_b, the problem of estimating covariance parameters becomes more complicated. In this case we would solve additional equations of the form

$$\hat{r}^\mathsf{T}(I-Q_r)(I-Q_s)\hat{r}=\mathbb{E}[r^\mathsf{T}(I-Q_r)(I-Q_s)r] \tag{2.27}$$

for each $r,s\in\{0,\cdots,p\}$.

However, we need Σ_a and Σ_b to be positive definite, and solving linear equations for Σ_a and Σ_b may not necessarily guarantee this without imposing additional constraints. We cannot rely on consistency either, since we would need an extremely large sample to guarantee a positive definite solution.

Rather than pursuing this approach, we developed a likelihood method using variational EM, which we discuss next.

2.4 Likelihood approaches for estimating all the parameters

2.4.1 Maximum likelihood estimation

Here we attempt to maximize the likelihood function for $(\boldsymbol{\beta}, \Sigma_a, \Sigma_b, \sigma_e^2)$ in (3.14),

$$\underset{\boldsymbol{\beta},\Sigma_a,\Sigma_b,\sigma_e^2}{\text{maximize}} \frac{1}{(\sqrt{2\pi})^N|\mathcal{V}|} \exp\left(-\frac{1}{2}(Y - X\boldsymbol{\beta})^\mathsf{T}\mathcal{V}^{-1}(Y - X\boldsymbol{\beta})\right), \tag{2.28}$$

with $\mathcal{V}$ defined as in (2.5). We showed in Section 2.2 that if the covariance parameters are known, we can use backfitting to obtain $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ in an efficient way. The problem becomes much more complicated if the covariance parameters are unknown, since the parameters Σ_a, Σ_b, σ_e^2 are buried in $\mathcal{V}$ in a complex manner. The R package "lme4" [Bates et al., 2015] implements Gaussian maximum likelihood estimation for the random effects model, but it is computationally expensive for large problems. We will compare the efficiency of our proposed algorithm to "lme4" in Section 2.5.

2.4.2 Expectation Maximization Algorithm

One potential solution to simplify the maximum likelihood problem analytically is to use the EM algorithm. In our setup, EM would treat the $\{\boldsymbol{a}_i\}_{i=1}^R$ and $\{\boldsymbol{b}_j\}_{j=1}^C$ as unobserved/missing quantities. In this case, the complete log-likelihood is (ignoring constants)

$$\begin{aligned} l(Y;\{\boldsymbol{a}_i\};\{\boldsymbol{b}_j\}) &= -\frac{1}{2}\left(N\log(\sigma_e^2) + R\log(|\Sigma_a|) + C\log(|\Sigma_b|)\right) \\ &-\frac{1}{2}\left(\sum_{i=1}^R\sum_{j=1}^C \frac{z_{ij}(y_{ij} - \mathbf{x}_{ij}^\mathsf{T}(\boldsymbol{\beta} + \boldsymbol{a}_i + \boldsymbol{b}_j))^2}{\sigma_e^2} + \sum_{i=1}^R \boldsymbol{a}_i^\mathsf{T}\Sigma_a^{-1}\boldsymbol{a}_i + \sum_{j=1}^C \boldsymbol{b}_j^\mathsf{T}\Sigma_b^{-1}\boldsymbol{b}_j\right). \end{aligned} \tag{2.29}$$

We have seen part of this expression earlier when we discussed backfitting to obtain $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ in Section 1.

EM is an iterative algorithm that involves two steps:

E-step: Compute the expected value of the complete log-likelihood conditional on Y and the current parameters $\hat{\boldsymbol{\theta}}^{(k)}$, giving the function $\ell_E(\boldsymbol{\theta})$.

M-step: Maximize $\ell_E(\boldsymbol{\theta})$ to produce new parameter estimates, $\hat{\boldsymbol{\theta}}^{(k+1)}$.

We have used $\boldsymbol{\theta}$ to represent all the parameters $(\beta, \Sigma_a, \Sigma_b, \sigma_e^2)$. The E-step would require computing $\mathbb{E}[\boldsymbol{a}_i\boldsymbol{a}_i^\mathsf{T}|Y;\boldsymbol{\theta}]$, $\mathbb{E}[\boldsymbol{b}_j\boldsymbol{b}_j^\mathsf{T}|Y;\boldsymbol{\theta}]$, and

$$\mathbb{E}\left[\sum_{i=1}^R\sum_{j=1}^C z_{ij}\left(y_{ij} - \mathbf{x}_{ij}^\mathsf{T}(\boldsymbol{\beta}^{(k)} + \boldsymbol{a}_i + \boldsymbol{b}_j)\right)^2 |Y;\boldsymbol{\theta}\right]. \tag{2.30}$$

Computing (2.30) involves computing $\mathbb{E}[\boldsymbol{a}_i|Y;\boldsymbol{\theta}]$, $\mathbb{E}[\boldsymbol{b}_j|Y;\boldsymbol{\theta}]$, and $\mathbb{E}[\boldsymbol{a}_i^\intercal\boldsymbol{b}_j|Y;\boldsymbol{\theta}]$ for $i \in \{0,\cdots,p\}$ and $j \in \{0,\cdots,p\}$. All is manageable except the terms $\mathbb{E}[\boldsymbol{a}_i^\intercal\boldsymbol{b}_j|Y;\boldsymbol{\theta}]$. Here, $(\boldsymbol{a}_1,\cdots,\boldsymbol{a}_R,\boldsymbol{b}_1,\cdots\boldsymbol{b}_C)$ follows a multivariate normal distribution conditional on Y and the covariance matrix for the conditional distribution would only be obtained by inverting a matrix of order $(R+C)p\times(R+C)p$. The estimate of σ_e^2 in the M-step requires the computation(2.30), and presents us with time complexity of at least $\{(R+C)p\}^3$ which is greater than or equal to $N^{3/2}p^3$.

Variational EM helps us out here. Instead of using the conditional distribution detailed above, we consider conditional distributions for $(\boldsymbol{a}_1,\cdots,\boldsymbol{a}_R,\boldsymbol{b}_1,\cdots\boldsymbol{b}_C)$ under which $\{\boldsymbol{a}_i\}_{i=1}^R$ are independent of $\{\boldsymbol{b}_j\}_{j=1}^C$.

2.4.3 Variational EM Algorithm

Variational EM was introduced by Beal and Ghahramani [2003] and is applicable when the missing data has a graphical model structure as in our case, i.e., $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ are correlated conditional on Y. Menictas et al. [2022] used streamlined variational inference for the Bayesian crossed random effects model. Our purpose in using variational EM is to estimate an approximation to the maximum likelihood estimate in the frequentist paradigm. The algorithm we develop here is a substitute for the maximum likelihood estimate for the crossed random effects model. Since we are unable to directly estimate the covariance parameters efficiently, variational EM is a practical alternative. Though variational EM does not come with the guarantee of maximization of the likelihood, the estimates are found to be empirically consistent in our simulations. Variational EM also has a close connection with backfitting which we discuss further.

Review of variational EM

Let $\boldsymbol{\alpha}$ denote the vector of unobserved quantities, $\boldsymbol{\theta}$ denote the set of parameters, and $l(\boldsymbol{\theta})$ represent the likelihood function of $\boldsymbol{\theta}$. In the E-step, we calculate the expectation of the complete likelihood given Y, i.e., sufficient statistics of $\boldsymbol{\alpha}$ get replaced by their expectation under the distribution $\mathbb{P}(\boldsymbol{\alpha}|Y)$. Under factored variational EM, instead of finding the expectation of the complete likelihood given Y, we substitute the conditional distribution of $\boldsymbol{\alpha}$ given Y with $q(\boldsymbol{\alpha})$ belonging in a restricted set Q of probability distributions.
We define

$$F(q,\boldsymbol{\theta}) := \int q(\boldsymbol{\alpha})\log\frac{\mathbb{P}(Y,\boldsymbol{\alpha}|\boldsymbol{\theta})}{q(\boldsymbol{\alpha})}d\boldsymbol{\alpha} \tag{2.31}$$

which we view as an approximation to $l(\boldsymbol{\theta})$; if $q(\boldsymbol{\alpha}) = \mathbb{P}(\boldsymbol{\alpha}|Y, \boldsymbol{\theta})$ then $F(q, \boldsymbol{\theta}) = l(\boldsymbol{\theta})$. In more detail,

$$\begin{aligned} F(q, \boldsymbol{\theta}) &= \int q(\boldsymbol{\alpha}) \log \frac{\mathbb{P}(Y, \boldsymbol{\alpha}|\boldsymbol{\theta})}{q(\boldsymbol{\alpha})} d\boldsymbol{\alpha} \\ &= \int q(\boldsymbol{\alpha}) \log \frac{\mathbb{P}(Y|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\alpha}|Y, \boldsymbol{\theta})}{q(\boldsymbol{\alpha})} d\boldsymbol{\alpha} \\ &= \int q(\boldsymbol{\alpha}) \left\{ \log \mathbb{P}(Y|\boldsymbol{\theta}) + \log \left(\frac{\mathbb{P}(\boldsymbol{\alpha}|Y, \boldsymbol{\theta})}{q(\boldsymbol{\alpha})} \right) \right\} d\boldsymbol{\alpha} \\ &= l(\boldsymbol{\theta}) - \mathrm{KL}\left(q(\boldsymbol{\alpha}) \| \mathbb{P}(\boldsymbol{\alpha}|Y, \boldsymbol{\theta})\right). \end{aligned}$$

As, $\mathrm{KL}\left(q(\boldsymbol{\alpha}) \| \mathbb{P}(\boldsymbol{\alpha}|Y, \boldsymbol{\theta})\right) \geq 0$, $F(q, \boldsymbol{\theta}) \leq l(\boldsymbol{\theta})$ with equality iff $q(\boldsymbol{\alpha}) = \mathbb{P}(\boldsymbol{\alpha}|Y, \boldsymbol{\theta})$. At each variational E-step, we maximize $F(q, \boldsymbol{\theta})$ with respect to q within the set Q. This implies that variational EM maximizes a lower bound to the likelihood. If $Q = \{q : q(\boldsymbol{\alpha}) = \prod q_u(\boldsymbol{\alpha}_u)\}$, it follows from Beal and Ghahramani [2003] that for the exponential family, optimal update for distribution of each component of $\boldsymbol{\alpha}$ is given by

$$q_u^{(k)}(\boldsymbol{\alpha}_u) \propto \mathbb{P}\left(Y, \boldsymbol{\alpha}_u | \boldsymbol{\alpha}_{-u} = \mathbb{E}_{q^{(k-1)}}(\boldsymbol{\alpha}_{-u})\right). \tag{2.32}$$

Variational EM possesses the property that

$$F(q^{(k-1)}, \boldsymbol{\theta}^{(k-1)}) \leq F(q^{(k)}, \boldsymbol{\theta}^{(k-1)}) \leq F(q^{(k)}, \boldsymbol{\theta}^{(k)}). \tag{2.33}$$

Variational EM for the crossed random effects model with random slopes

For the crossed random effects model (2.3), $\boldsymbol{\alpha}$ would have two components, $\{\boldsymbol{a}_i\}_{i=1}^R$ and $\{\boldsymbol{b}_j\}_{j=1}^C$. Here, $\boldsymbol{\theta}$ represents the set of parameters $(\boldsymbol{\beta}, \Sigma_a, \Sigma_b, \sigma_e^2)$. We choose

$$Q = \{q : q(\boldsymbol{a}_1, \cdots, \boldsymbol{a}_R, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_C) = q_a(\boldsymbol{a}_1, \cdots, \boldsymbol{a}_R) q_b(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_C)\}. \tag{2.34}$$

Lemma 1. *The optimal distributions for* $\{\boldsymbol{a}_i\}_{i=1}^R$ *and* $\{\boldsymbol{b}_j\}_{i=1}^R$ *in the variational E-step are given by*

$$q_a^{(k)}(\boldsymbol{a}_1, \cdots, \boldsymbol{a}_R) = \prod_{i=1}^R q^{(k)}(\boldsymbol{a}_i) \quad \textit{and} \quad q_b^{(k)}(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_C) = \prod_{j=1}^C q^{(k)}(\boldsymbol{b}_j)$$

with

$$q^{(k)}(\boldsymbol{a}_i) = N(\mu_{a,i}^{(k)}, \Sigma_{a,i}^{(k)}) \quad \textit{and} \quad q^{(k)}(\boldsymbol{b}_j) = N(\mu_{b,j}^{(k)}, \Sigma_{b,j}^{(k)}), \tag{2.35}$$

where

$$\mu_{a,i}^{(k)} = \Sigma_{a,i}^{(k)} \left(\frac{\sum_{j=1}^{C} z_{ij} \left(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}} \left(\boldsymbol{\beta}^{(k-1)} + \mu_{b,j}^{(k-1)} \right) \right) \mathbf{x}_{ij}}{\left(\sigma_e^{(k-1)} \right)^2} \right) \tag{2.36}$$

and

$$\Sigma_{a,i}^{(k)} = \left((\Sigma_a^{(k-1)})^{-1} + \frac{\sum_{j=1}^{C} z_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^{\mathsf{T}}}{\left(\sigma_e^{(k-1)} \right)^2} \right)^{-1} \tag{2.37}$$

with $\mu_{b,j}^{(k)}$ *and* $\Sigma_{b,j}^{(k)}$ *having similar definitions.*

Proof. The proof can be found in Appendix A.0.6. □

It is important to note that while updating the distribution of $\{\boldsymbol{a}_i\}_{i=1}^{R}$, we use the parameters of the distribution of $\{\boldsymbol{b}_j\}_{j=1}^{C}$ updated in the previous step. Likewise when updating the distribution of $\{\boldsymbol{b}_j\}_{j=1}^{C}$, we use parameters of the distribution of $\{\boldsymbol{a}_i\}_{i=1}^{R}$ recently updated. Thus, the mechanism of the update is very similar to the backfitting approach where we iterate through updates in $\{\boldsymbol{a}_i\}_{i=1}^{R}$ and $\{\boldsymbol{b}_j\}_{j=1}^{C}$. Each of the quantities in (3.27) and (3.29) can be computed in $\mathcal{O}(N)$ as computing $\mu_{a,i}^{(k)}$ and $\mu_{b,j}^{(k)}$ is equivalent to obtaining the estimates of $\{\boldsymbol{a}_i\}_{i=1}^{R}$ and $\{\boldsymbol{b}_j\}_{j=1}^{C}$ in each step of backfitting.

Variational E step: The expectation of the complete likelihood (2.29) can be computed with respect to the probability distribution $q^{(k)}$, though we can skip to the M-step and fill in the details there.

M step: The estimate $\Sigma_a^{(k)}$ is given by

$$\begin{aligned} \Sigma_a^{(k)} :&= \frac{1}{R} \mathbb{E}_{q^{(k)}} \left[\sum_{i=1}^{R} \boldsymbol{a}_i \boldsymbol{a}_i^{\mathsf{T}} \right] \\ &= \frac{1}{R} \sum_{i=1}^{R} \left(\mu_{a,i}^{(k)} {\mu_{a,i}^{(k)}}^{\mathsf{T}} + \Sigma_{a,i}^{(k)} \right) \end{aligned} \tag{2.38}$$

Similarly,

$$\Sigma_b^{(k)} = \frac{1}{C} \sum_{j=1}^{C} \left(\mu_{b,j}^{(k)} {\mu_{b,j}^{(k)}}^{\mathsf{T}} + \Sigma_{b,j}^{(k)} \right). \tag{2.39}$$

Also,

$$\boldsymbol{\beta}^{(k)} = \left(\sum_{i=1}^{R} \sum_{j=1}^{C} z_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^{\mathsf{T}} \right)^{-1} \sum_{i=1}^{R} \sum_{j=1}^{C} z_{ij} \mathbf{x}_{ij} \mathbb{E} \left(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}} \left(\mu_{a,i}^{(k)} + \mu_{b,j}^{(k)} \right) \right), \tag{2.40}$$

and finally

$$\begin{aligned}
\left(\sigma_e^{(k)}\right)^2 &= \frac{1}{N}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\mathbb{E}_{q^{(k)}}\left[\left(y_{ij}-\mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta}^{(k)}+\boldsymbol{a}_i+\boldsymbol{b}_j)\right)^2\right] \\
&= \frac{1}{N}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\left(\mathbb{E}_{q^{(k)}}\left[\left(y_{ij}-\mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta}^{(k)}+\boldsymbol{a}_i+\boldsymbol{b}_j)\right)\right]\right)^2 \\
&\qquad + \frac{1}{N}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\, Var_{q^{(k)}}\left[\left(y_{ij}-\mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta}^{(k)}+\boldsymbol{a}_i+\boldsymbol{b}_j)\right)\right] \\
&= \frac{1}{N}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\left(\mathbb{E}_{q^{(k)}}\left[\left(y_{ij}-\mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta}^{(k)}+\boldsymbol{a}_i+\boldsymbol{b}_j)\right)\right]\right)^2 \\
&\qquad + \frac{1}{N}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\, Var_{q^{(k)}}\left[\mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{a}_i+\boldsymbol{b}_j)\right] \\
&= \frac{1}{N}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\left(y_{ij}-\mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta}^{(k)}+\mu_{a,i}^{(k)}+\mu_{b,j}^{(k)})\right)^2 \\
&\qquad\qquad + \frac{1}{N}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\mathbf{x}_{ij}^{\mathsf{T}}\left(\Sigma_{a,i}^{(k)}+\Sigma_{b,j}^{(k)}\right)\mathbf{x}_{ij}.
\end{aligned}$$

The computational cost of the variational E-step and M-step is $\mathcal{O}(Np^2)$.

2.4.4 Connection of variational EM with backfitting

If the covariance parameters were known or otherwise previously estimated, the goal of variational EM would be just to estimate $\boldsymbol{\beta}$. In that case, optimal distribution for $\{\boldsymbol{a}_i\}_{i=1}^R$ and $\{\boldsymbol{b}_j\}_{i=1}^R$ in the variational E-step are given by

$$q^{(k)}(\boldsymbol{a}_i) = N(\mu_{a,i}^{(k)}, \Sigma_{a,i}) \quad \text{and} \quad q^{(k)}(\boldsymbol{b}_j) = N(\mu_{b,j}^{(k)}, \Sigma_{b,j}), \tag{2.41}$$

where

$$\mu_{a,i}^{(k)} = \Sigma_{a,i}\left(\frac{\sum_{j=1}^{C} z_{ij}\left(y_{ij}-\mathbf{x}_{ij}^{\mathsf{T}}\left(\boldsymbol{\beta}+\mu_{b,j}^{(k-1)}\right)\right)\mathbf{x}_{ij}}{\sigma_e^2}\right) \tag{2.42}$$

and

$$\Sigma_{a,i} = \left(\Sigma_a^{-1} + \frac{\sum_{j=1}^{C} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}}}{\sigma_e^2}\right)^{-1} \tag{2.43}$$

with $\mu_{b,j}^{(k)}$ and $\Sigma_{b,j}$ having similar definitions. It is important to note that the variance of $\{\boldsymbol{a}_i\}$ and $\{\boldsymbol{b}_j\}$ are now fixed over the iterations, because the covariance parameters, Σ_a, Σ_b, and σ_e^2 are

assumed to be known and fixed quantities. Equations (2.42) and (2.43) imply that

$$\mu_{a,i}^{(k)} = \left(\sigma_e^2\Sigma_a^{-1} + \sum_{j=1}^{C} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}}\right)^{-1}\left(\sum_{j=1}^{C} z_{ij}\left(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \mu_{b,j}^{(k-1)})\right)\mathbf{x}_{ij}\right), \tag{2.44}$$

and similarly,

$$\mu_{b,j}^{(k)} = \left(\sigma_e^2\Sigma_b^{-1} + \sum_{i=1}^{R} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}}\right)^{-1}\left(\sum_{i=1}^{R} z_{ij}\left(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \mu_{a,i}^{(k)})\right)\mathbf{x}_{ij}\right). \tag{2.45}$$

The estimate of $\boldsymbol{\beta}$ in E-step remains unchanged and is given by

$$\boldsymbol{\beta}^{(k)} = \left(\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}}\right)^{-1}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\mathbf{x}_{ij}\mathbb{E}\left(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}\left(\mu_{a,i}^{(k)} + \mu_{b,j}^{(k)}\right)\right). \tag{2.46}$$

It is easy to see that update of $\mu_{a,i}^{(k)}$, $\mu_{b,j}^{(k)}$, and $\hat{\boldsymbol{\beta}}$ in each step of variational EM (2.44) , (2.45) with known covariance parameters is same as update of $\hat{\boldsymbol{a}}_i$, $\hat{\boldsymbol{b}}_j$ and $\boldsymbol{\beta}$ in each step of backfitting . This draws a close connection between variational EM and backfitting. We know that $\hat{\boldsymbol{a}}_i \to \mathbb{E}[\boldsymbol{a}_i|Y]$ and $\hat{\boldsymbol{b}}_j \to \mathbb{E}[\boldsymbol{b}_j|Y]$ over the iterations in backfitting which provides a plausible justification for variational EM.

2.4.5 Estimation of covariance of $\hat{\boldsymbol{\beta}}$

Let S_G denotes the linear operator corresponding to the fit due to random components $\{\boldsymbol{a}_i\}$ and $\{\boldsymbol{b}_j\}$, i.e., $S_G(Y)_{ij} = \mathbf{x}_{ij}^{\mathsf{T}}(\hat{\boldsymbol{a}}_i + \hat{\boldsymbol{b}}_j)$, then similar to the development in Section 2.2.3, we can write

$$\hat{\boldsymbol{\beta}} = (X^{\mathsf{T}}(I - S_G)X)^{-1}(X^{\mathsf{T}}(I - S_G)Y). \tag{2.47}$$

It is important to note here that S_G is a *symmetric* operator as it is an operator matrix for a generalised linear regression. Hence we can write

$$\hat{\boldsymbol{\beta}} = \left(X^{\mathsf{T}}(X - S_G X)\right)^{-1}\left((X - S_G X)^{\mathsf{T}}Y\right). \tag{2.48}$$

Another important thing to mention here is that we never compute or store the operator S_G, instead we apply the backfitting procedure similar to Section 2.2.3 to obtain an estimate of $S_G(X_j)$ given by $\tilde{S}_G(X_j)$ for each column X_j of X.

Let $\tilde{X} = (I - \tilde{S}_G)X$, then our estimate for the covariance matrix of $\hat{\boldsymbol{\beta}}$ is

$$(X^{\mathsf{T}}\tilde{X})^{-1}\tilde{X}^{\mathsf{T}}\widehat{\mathcal{V}}\tilde{X}(\tilde{X}^{\mathsf{T}}X)^{-1}, \tag{2.49}$$

where $\widehat{\mathcal{V}}$ is an estimate for the covariance matrix of Y. Using the definition of $\widehat{\mathcal{V}}$ in (2.5), $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ is given by

$$(X^\mathsf{T}\tilde{X})^{-1}\tilde{X}^\mathsf{T}\left(X\hat{\Sigma}_a X^\mathsf{T} \bullet Z_a Z_a^\mathsf{T} + X\hat{\Sigma}_b X^\mathsf{T} \bullet Z_b Z_b^\mathsf{T} + \hat{\sigma}_e^2 I_N\right)\tilde{X}(\tilde{X}^\mathsf{T}X)^{-1}. \tag{2.50}$$

If we directly implement the matrix multiplication above, the operation cannot be performed in time complexity which is linear in N. However, exploiting the structure of the expression, a careful implementation brings the complexity down to linear in N. The details can be found in Appendix A.0.5. The covariance matrix of $\hat{\boldsymbol{\beta}}$ can be used to perform inference on $\boldsymbol{\beta}$ and z-tests.

2.5 Results

We compare the time required for maximum likelihood estimation by "lmer" Bates et al. [2015] with our methods. We also illustrate the empirical consistency of our methods using simulations. Finally, we implement our method on MovieLens data and Stitch Fix data, and compare the results with the OLS fit and the crossed random effects model with random intercepts implemented by Ghosh et al. [2022a].

2.5.1 Simulations

Error in estimating covariance matrices

To articulate the errors in estimating Σ_a and Σ_b, we are using two metrics.

- **KL divergence**: We define the distance between Σ_a and $\hat{\Sigma}_a$ in the following way. Let $\mathbb{P}_1$ denote the normal distribution with mean zero and covariance Σ_a and $\mathbb{P}_2$ denote the normal distribution with mean zero and covariance $\widehat{\Sigma}_a$, then we define the distance by

$$\mathbb{E}_{\mathbb{P}_1}\left[\log\left(\frac{d\mathbb{P}_1}{d\mathbb{P}_2}\right)\right] = \frac{1}{2}\left[\mathrm{tr}\left(\widehat{\Sigma}_a^{-1}\Sigma_a\right) - \log(|\widehat{\Sigma}_a^{-1}\Sigma_a|) - d\right]. \tag{2.51}$$

- **Frobenius norm of difference between Σ_a and $\widehat{\Sigma}_a$**:

$$\|\Sigma_a - \widehat{\Sigma}_a\|_F$$

For the simulations, we consider the model given by (2.3) with random slopes for all the covariates for clients as well as items. However, as mentioned in remark 1, in practice we typically include random slopes for only a subset of covariates for both clients and items. We present the results for the general setup in Appendix B.

Diagonal case

For our simulations, we choose $p = 3, \beta_0 = 0.1$, $\boldsymbol{\beta} = (0.2, 0.3, 0.4)$, and $\sigma_e^2 = 1$. We considered

$$\Sigma_a = \begin{bmatrix} 0.3 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix} \text{ and } \Sigma_b = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}.$$

We choose $R = C = N^{0.6}$ for our simulations which satisfies the sparsity assumption as seen in practice/e-commerce data. For each value of N, we perform 100 simulations and the plot the average values. We have proposed the following algorithms:

Backfitting: We estimate covariance parameters using the method of moments (MoM) and then perform backfitting to estimate the fixed parameter.

Clubbed Backfitting: We estimate covariance parameters using MoM and then perform backfitting with clubbing to estimate the fixed parameter.

Variational EM: We use MoM to get an initial estimate of covariance parameters and then we improve the estimates in each iteration using the variational M-step along with estimating fixed parameters.

Clubbed variational EM: Similar to variational EM, we start with MoM estimates for covariance parameters and improvise them using variational EM. We apply "clubbing" trick while estimating $\boldsymbol{\beta}$ to fasten the convergence.

Summary of estimation time: The slope of a curve in Figure 2.2 represents the exponent of N in time complexity of the associated algorithm. Therefore, the time complexity of "lmer" is $\mathcal{O}(N^{3/2})$ or worse. On the other hand, the time complexity of all of our suggested algorithms is linear in N. The reason that the algorithms with the "clubbing" trick seem to have time complexity with much smaller exponent in N is that the number of iterations required for those algorithms to converge decreases with N. Therefore, although time complexity of each step is linear in N, the total time complexity seems to be smaller than $\mathcal{O}(N)$. For $N = 10^5 = 100,000$, it takes around three days (on the Stanford Sherlock cluster even with parallel processing using "mclapply" function in R) to perform 100 simulations using "lmer" as the fitting function. Therefore, in the interest of time, we discontinue using "lmer" for values of N larger than 100, 000.

Figure 2.3 depicts empirical consistency of $\widehat{\Sigma}_a$ and $\widehat{\Sigma}_b$ estimated through MoM and variational EM as errors explained in 2.5.1 go to zero as N increases. Figure 2.4 shows that the accuracy of different estimates of $\boldsymbol{\beta}$, i.e., $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ obtained though different approaches and $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ coincide, i.e.,

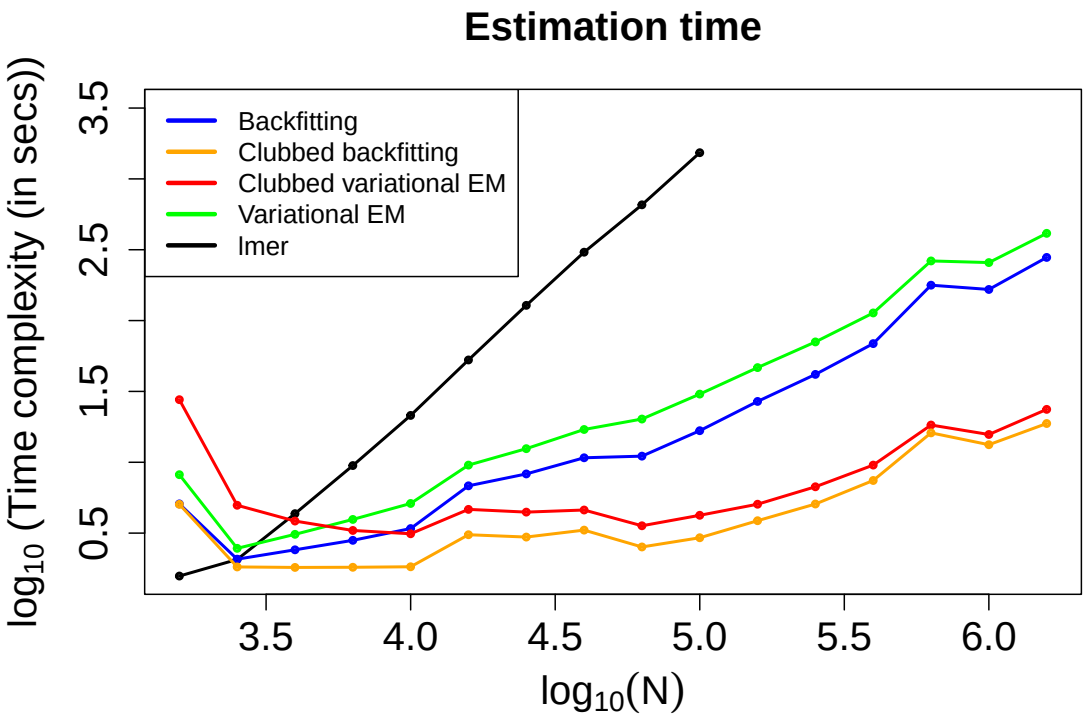


Figure 2.2: Diagonal case: Comparison of estimation time and number of iterations by backfitting, clubbed backfitting, variational EM, and clubbed variational EM (four proposed algorithms) and "lmer"(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer".

errors are very close to each other and thus $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ is consistent. Figure 2.4 also depicts the empirical consistency of the estimate of σ_e^2 obtained through the method of moments approach and variational EM approach. We see that estimates obtained through MoM approach are consistent but variational EM helps in improving those estimates. The errors made in estimating all the parameters get close to maximum likelihood estimates as N increases.

Non diagonal case

For the non-diagonal case, we also choose $p = 3, \boldsymbol{\beta} = (0.2, 0.3, 0.4), \beta_0 = 0.1, \sigma_e^2 = 1$. We considered

$$\Sigma_a = \Sigma_b = \begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 0.2 & 1 \end{bmatrix}.$$

We choose $R = C = N^{0.6}$ for our simulations and simulated 100 examples similar to the diagonal case. Figure 2.5 depicts empirical consistency of $\widehat{\Sigma}_a$ and $\widehat{\Sigma}_b$ estimated through variational EM as errors explained in 2.5.1 go to zero as N increases. Figure 2.6 shows that the accuracy of different estimates of $\boldsymbol{\beta}$, i.e., $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ obtained though different approaches and $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ coincide and thus $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ is consistent. The estimates of σ_e^2 obtained through MoM approach and variational EM depict empirical consistency in figure 2.6 although the estimate obtained through variational EM has a faster rate of convergence. We don't plot the error in estimating covariance matrices by MoM approach as method of moments is not fruitful in estimating covariances matrices of random slopes when they aren't diagonal. Figure 2.7 shows that our variational EM and backfitting approach take a great lead compared to the "lmer" approach. The average time complexity of "lmer" approach is at least $\mathcal{O}(N^{1.5})$ in our setup, while all the approach discussed by us have time complexity $\mathcal{O}(N)$. The number of iterations required to converge grows with N for "lmer" and the approaches without clubbing, while the number of iterations decreases with N for clubbing-based approaches. Here is a summary of the results:

	Without Clubbing	With Clubbing
Backfitting with MoM	👎 Slow convergence 👎 Poor estimates for Σ_a, Σ_b	👍 Fast convergence 👎 Poor estimates for Σ_a, Σ_b
Variational EM	👎 Slow convergence 👍 Good estimates for Σ_a, Σ_b	👍 Fast convergence 👍 Good estimates for Σ_a, Σ_b

Thus, clubbed variational EM provides precise estimates of all the covariance parameters with a small estimation time.

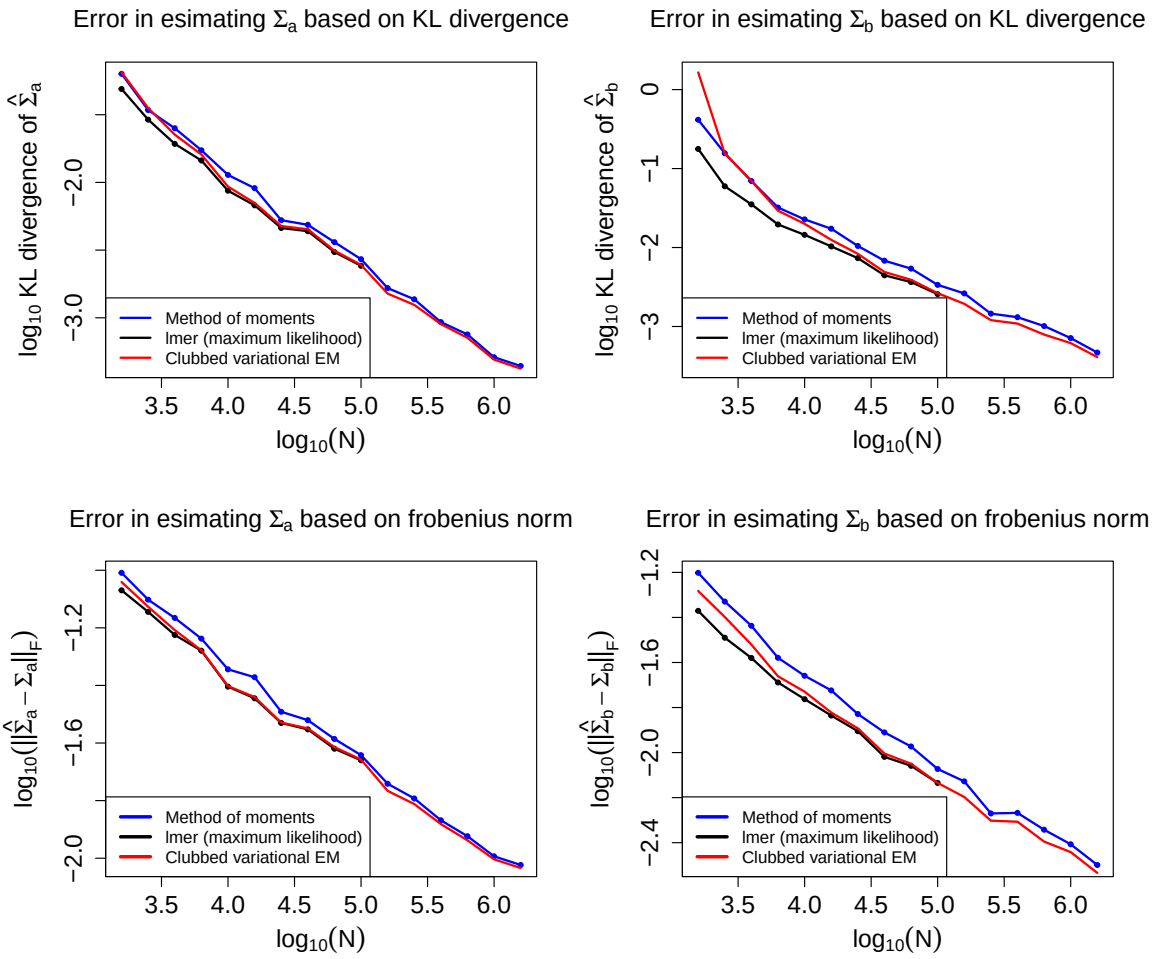


Figure 2.3: Diagonal case: Comparison of accuracy in estimating Σ_a and Σ_b based on KL divergence and Frobenius norm by the method of moments (proposed algorithm) and lmer(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer".

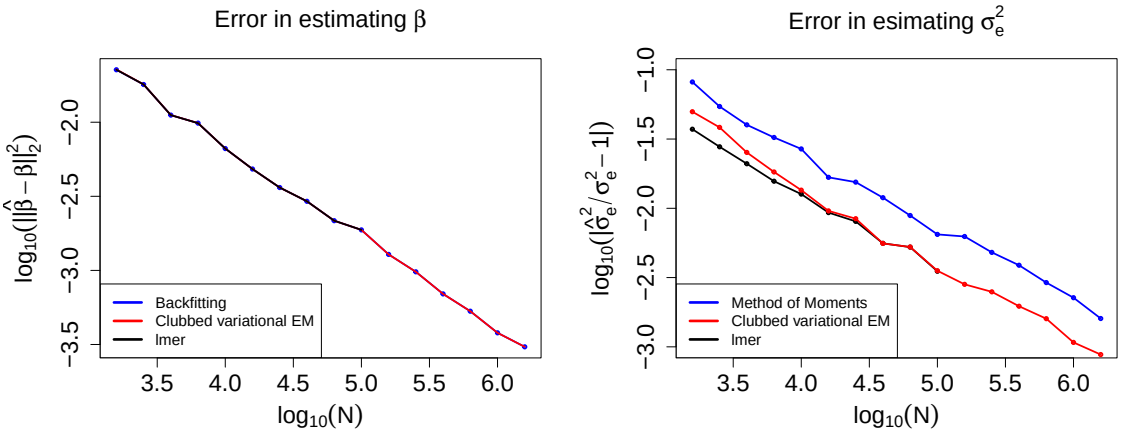


Figure 2.4: Diagonal case: Comparison of accuracy in estimating $\boldsymbol{\beta}$ and σ_e^2 by method of moments (proposed algorithm) and lmer (existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer". The errors in estimating $\boldsymbol{\beta}$ are close to each other for the illustrated algorithms, which leads the lines in the plot to coincide.

2.5.2 Application of crossed random effects model on 'Stitch Fix" data

Ghosh et al. [2022a] applied the backfitting algorithm to obtain $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ for the Stitch Fix data for the random intercepts model (3.1). Here, we demonstrate how our algorithm can be used to apply the crossed-random effects models to the Stitch Fix data using two models. The first model we implement considers a random intercept and a random slope for a variable named "match". "Match" is a continuous variable taking values in the range 0 to 1, representing the predicted probability that the client will keep the item purchased. The second model we implement considers random intercept and random slopes for the variable "match" and multiple indicator variables named "client edgy", "client boho", "item edgy", "item boho", etc. These refer to fashion styles that describe some items and some clients' preferences. Some customers and some items were given the adjective 'edgy' in the data set and another adjective was 'boho', short for 'Bohemian'. In both of the models, we assume multivariate normal distributions for the random effects for clients as well as items. It should be noted that the purpose of this section is not to choose the most appropriate random effects model. Instead, we wish to illustrate the application of our algorithm in a setup with random slopes. Along with fitting the models, we also measured the prediction accuracy of these models and compared them with ordinary least squares and the model with just random intercepts.

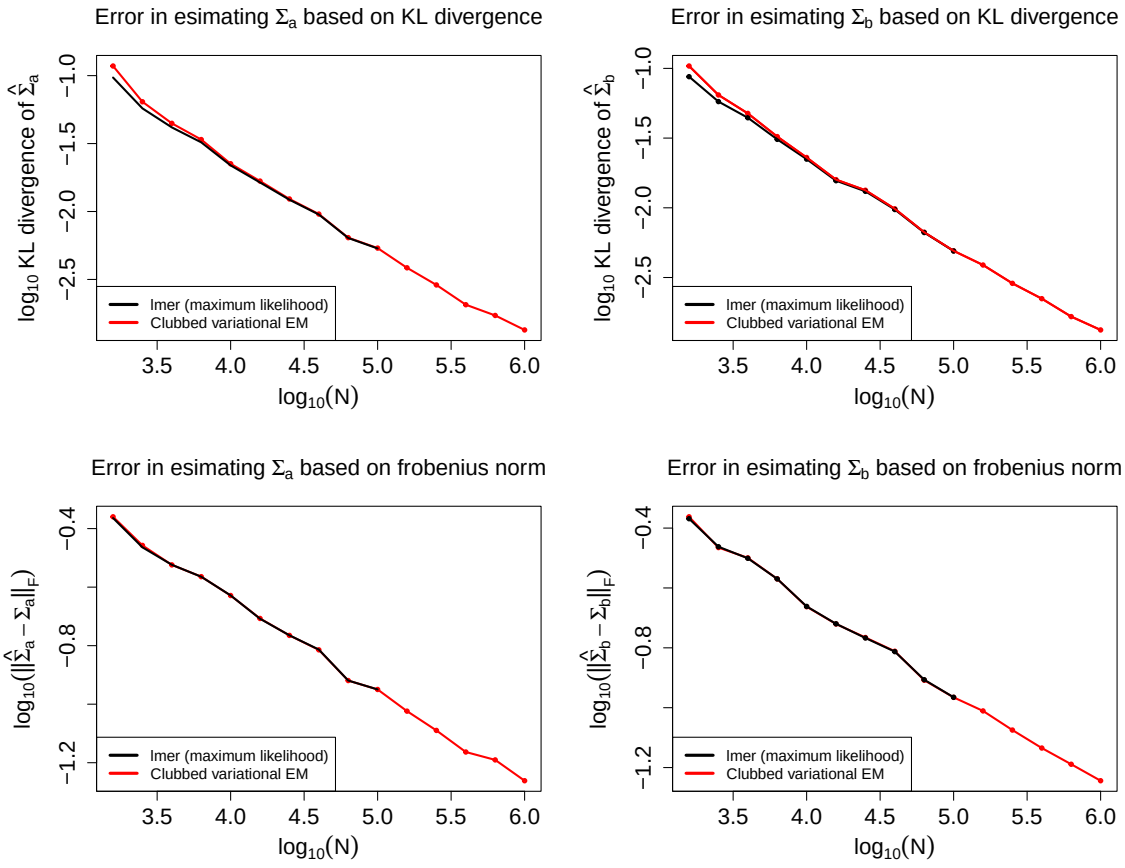


Figure 2.5: Non-diagonal case: Comparison of accuracy in estimating Σ_a and Σ_b based on KL divergence and Frobenius norm by variational EM (proposed algorithm) and lmer(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to large computational time for "lmer".

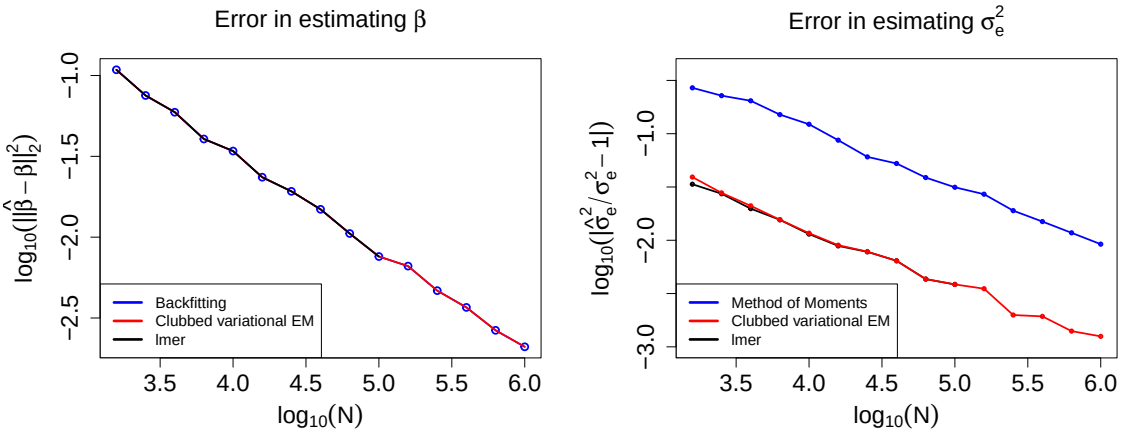


Figure 2.6: Non diagonal case: Comparison of accuracy in estimating $\boldsymbol{\beta}$ and σ_e^2 by variational EM (proposed algorithm) and lmer(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer". The errors in estimating $\boldsymbol{\beta}$ are close to each other for the illustrated algorithms, which leads the lines in the plot to coincide.

Model 1: random slope for the variable "match"

Here, we implement our devised algorithm to fit the random effects model with a random slope for the "match" variable. Thus, the model we consider here is

$$y_{ij} = \beta_0 + \tilde{\mathbf{x}}_{ij}^{\mathsf{T}}(\boldsymbol{a}_i + \boldsymbol{b}_j) + \mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + \varepsilon_{ij} \quad \forall i \in \{1, \cdots, R\} \quad j \in \{1, , \cdots, C\}, \tag{2.52}$$

where, $\boldsymbol{\beta}$ denotes the fixed effect for the entire covariate vector $\mathbf{x}_{ij}$ listed in table 3.1. We assume that $\boldsymbol{a}_i \sim \mathcal{N}_2(0, \Sigma_a)$ and $\boldsymbol{b}_j \sim \mathcal{N}_2(0, \Sigma_b)$ represent random effects for $\tilde{\mathbf{x}}_{ij} = (1, \text{match}_{ij})$. The estimate of fixed effect is tabulated in table 3.1 as $\hat{\boldsymbol{\beta}}_{\text{GLS},1}$ and the fixed effects for random intercepts model is tabulated as $\hat{\boldsymbol{\beta}}_{\text{GLS},0}$. The estimate of σ_e^2 is 4.454 and that of Σ_a and Σ_b is

$$\widehat{\Sigma_a} = \begin{pmatrix} 1.885 & -1.197 \\ -1.197 & 1.612 \end{pmatrix} \text{ and } \widehat{\Sigma_b} = \begin{pmatrix} 0.831 & -1.623 \\ -1.623 & 4.297 \end{pmatrix}.$$

Model 2: random slope for variables "match", "I(item edgy)", and "I(item boho)" for clients and random slope for variables "match", "I(client edgy)", and "I(client boho)" for items

The model considered here is

$$y_{ij} = \beta_0 + \mathbf{x}_{a,ij}^{\mathsf{T}}\boldsymbol{a}_i + \mathbf{x}_{b,ij}^{\mathsf{T}}\boldsymbol{b}_j + \mathbf{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + \varepsilon_{ij} \quad \forall i \in \{1, \cdots, R\} \quad j \in \{1, , \cdots, C\}, \tag{2.53}$$

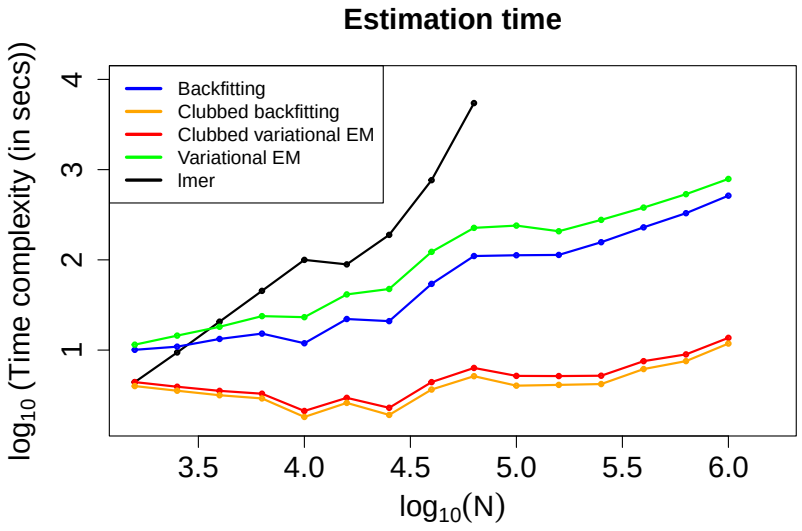


Figure 2.7: Non-diagonal case: Comparison of estimation time by backfitting, clubbed backfitting, variational EM, and clubbed variational EM (four proposed algorithms) and "lmer"(existing algorithm).

where, $\boldsymbol{\beta}$ denotes the fixed effect for the entire covariate vector $\mathbf{x}_{ij}$ similar as above. We assume that $\boldsymbol{a}_i \sim \mathcal{N}_4(0, \Sigma_a)$ represent random effects for

$$\mathbf{x}_{a,ij} = (1, \text{match}_{ij}, \text{I(item edgy)}_j, \text{I(item boho)}_j)$$

and $\boldsymbol{b}_j \sim \mathcal{N}_4(0, \Sigma_b)$ represent random effects for

$$\mathbf{x}_{b,ij} = (1, \text{match}_{ij}, \text{I(client edgy)}_i, \text{I(client boho)}_i)$$

The estimate of fixed effect is tabulated in table 3.1 as $\hat{\boldsymbol{\beta}}_{\text{GLS},2}$ and the estimate of σ_e^2 is 4.363 which is smaller than estimated variance of noise estimated for Model 1. The estimate of Σ_a and Σ_b is

$$\widehat{\Sigma_a} = \begin{pmatrix} 1.751 & -1.015 & -0.009 & 0.048 \\ -1.015 & 1.422 & -0.014 & -0.034 \\ -0.009 & -0.014 & 0.340 & 0.047 \\ 0.048 & -0.034 & 0.047 & 0.490 \end{pmatrix}$$

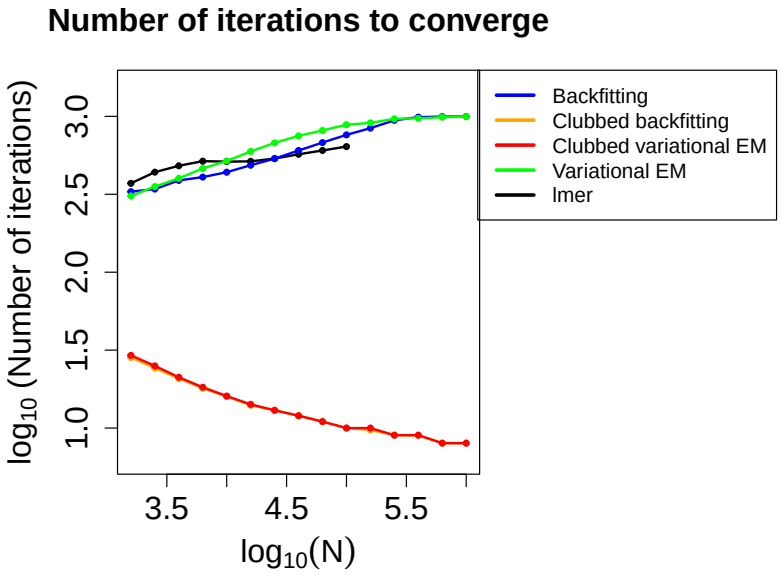


Figure 2.8: Non-diagonal case: Comparison of number of iterations by backfitting, clubbed backfitting, variational EM, and clubbed variational EM (four proposed algorithms) and "lmer"(existing algorithm). The plot for lmer is discontinued after $N = 10^5$ due to the large computational time for "lmer".

$$\widehat{\Sigma_b} = \begin{pmatrix} 0.834 & -1.574 & 0.000 & -0.022 \\ -1.574 & 4.185 & -0.032 & -0.006 \\ 0.000 & -0.032 & 0.024 & -0.001 \\ -0.022 & -0.006 & -0.001 & 0.022 \end{pmatrix}.$$

2.5.3 Naivety and inefficiencies with respect to OLS

Ghosh et al. [2022a] defined naivety and inefficiency of OLS with respect to the random intercepts model to establish the utility of fitting a random effects model. Ordinary least squares usually make two mistakes on crossed random effects data. The first is that OLS is naive about correlations in the data and this can lead it to severely underestimate the variance of $\hat{\boldsymbol{\beta}}$. The second is that OLS is inefficient compared to GLS by the Gauss-Markov theorem. Let $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ and $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ be the OLS and GLS estimates of $\boldsymbol{\beta}$, respectively. They quantified the naivety and inefficiency of OLS as follows:

- **Naivety** is defined by the ratios $\text{Var}_{\text{GLS}}(\hat{\boldsymbol{\beta}}_{\text{OLS},j})/\text{Var}_{\text{OLS}}(\hat{\boldsymbol{\beta}}_{\text{OLS},j})$ for $j \in \{1, \cdots, p\}$
- **Inefficiency** is defined by the ratios $\text{Var}_{\text{GLS}}(\hat{\boldsymbol{\beta}}_{\text{OLS},j})/\text{Var}_{\text{GLS}}(\hat{\boldsymbol{\beta}}_{\text{GLS},j})$ for $j \in \{1, \cdots, p\}$.

We carry the work forward by displaying the naivety and inefficiency of OLS with respect to the random effects model discussed above. For the first model, the values of naivety range from 2.63 to

933.47 while for the second model, the range is 3.80 to 915.86. These values are larger than naivety values reported in Ghosh et al. [2022a] as the largest value reported by them was 345.28. The largest and second-largest ratios are for material indicators corresponding to 'Modal' and 'Tencel' using the models with random slopes too. We also identified the linear combination of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ for which OLS is most naive. We maximize the ratio $\mathbf{x}^\intercal \widehat{\text{Cov}}_{\text{GLS},1}(\hat{\boldsymbol{\beta}}_{\text{OLS}})x / \mathbf{x}^\intercal \widehat{\text{Cov}}_{\text{OLS}}(\hat{\boldsymbol{\beta}}_{\text{OLS}})x$ over $x \neq 0$. The resulting maximal ratio is the largest eigenvalue of $\widehat{\text{Cov}}_{\text{OLS}}(\hat{\boldsymbol{\beta}}_{\text{OLS}})^{-1}\widehat{\text{Cov}}_{\text{GLS},1}(\hat{\boldsymbol{\beta}}_{\text{OLS}})$ and it is about 1008 while the ratio provided for the random intercepts model in Ghosh et al. [2022a] was 361. The resulting maximal ratio for $\hat{\boldsymbol{\beta}}_{GLS,2}$ is about 988. We also quantified inefficiency similar to Ghosh et al. [2022a]. The figure 2.9 and 2.10 plots the naivety and inefficiency values. The efficiency values range from 1.06 to 32.60 for the first model and can be interpreted as factors by which using random slopes for the variable "match" reduces the effective sample size. The efficiency values for the second model range from 1.07 to 32.43, therefore adding the random slope for other variables didn't decrease the effective sample size. The "match" variable is found to be an outlier in terms of efficiency for both of the random slopes models similar to the model with just random intercepts. An important thing to note here is that though OLS is much more naive with respect to the models with random slopes compared to the model with just random intercepts, although the difference in efficiency is not that huge.

2.5.4 Fixed effects for Stitch Fix data

Summary

In table 3.1, we present the estimated fixed effects while fitting linear model, crossed random effects model with random intercepts (3.1) and the two crossed random effects models with random intercepts as well as random slopes discussed in Section 2.5.2 and 2.5.2. The estimate for fixed effect of the variable "match" is 5.048 for OLS and the estimate is 3.442, a much smaller value for crossed random effects model with random intercepts. When we add random slopes associated with this variable for clients as well as items, the fixed effect reduces to 3.296. However, considering the randomness, the effect associated with this variable fluctates in the range 3.296 ± 1.269 for clients and 3.296 ± 2.073 for items. The textile indicator variables like "Acrylic", "Cashmere", "Fur", "Linen", etc., are found to be significant for the OLS model but they lose their significance on fitting crossed random effects models. A possible reason could be that OLS underestimates the standard errors. There are certain textile variables like "Leather", "PVC", and "Tencel" which are significant while fitting crossed random effects model with random intercepts but they loose significant on adding randomness for slope of "match". The explaination once again is that the crossed random effects model with just random intercepts also underestimates the standard errors similar to OLS, which was also implicated in the naivety and efficiency calculations above. Upon getting a more accurate

[1]Some customers and some items were given the adjective 'edgy' in the data set and another adjective was 'boho', short for 'Bohemian'. The remaining indicator variables represent the textile used in manufacturing the item.

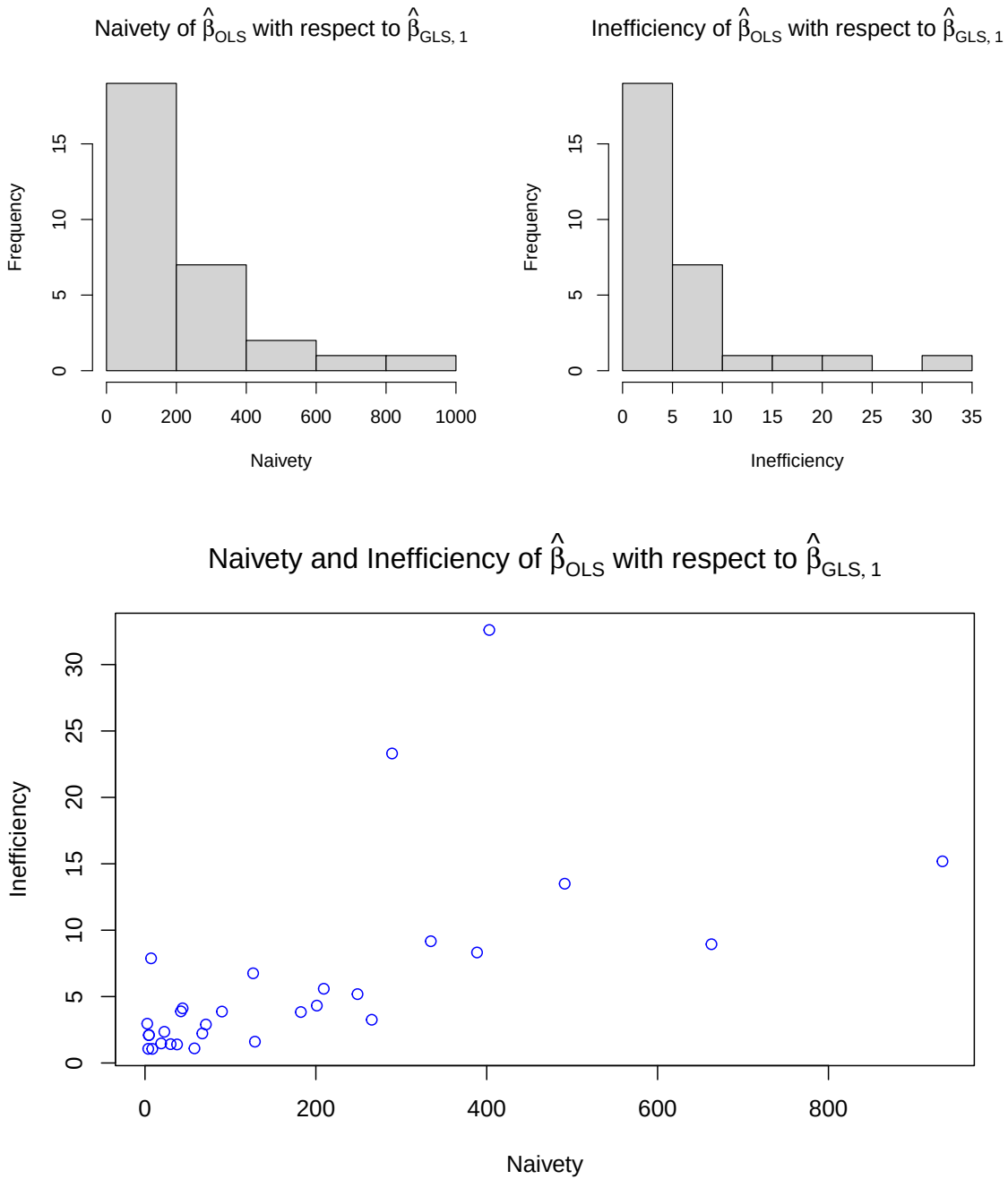


Figure 2.9: Naivety and efficiency of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ with respect to $\hat{\boldsymbol{\beta}}_{\text{GLS},1}$

picture of the standard errors, we find fewer covariates to be significant. The direction and magnitude of the fixed effects are mostly consistent between the two crossed random effects models for

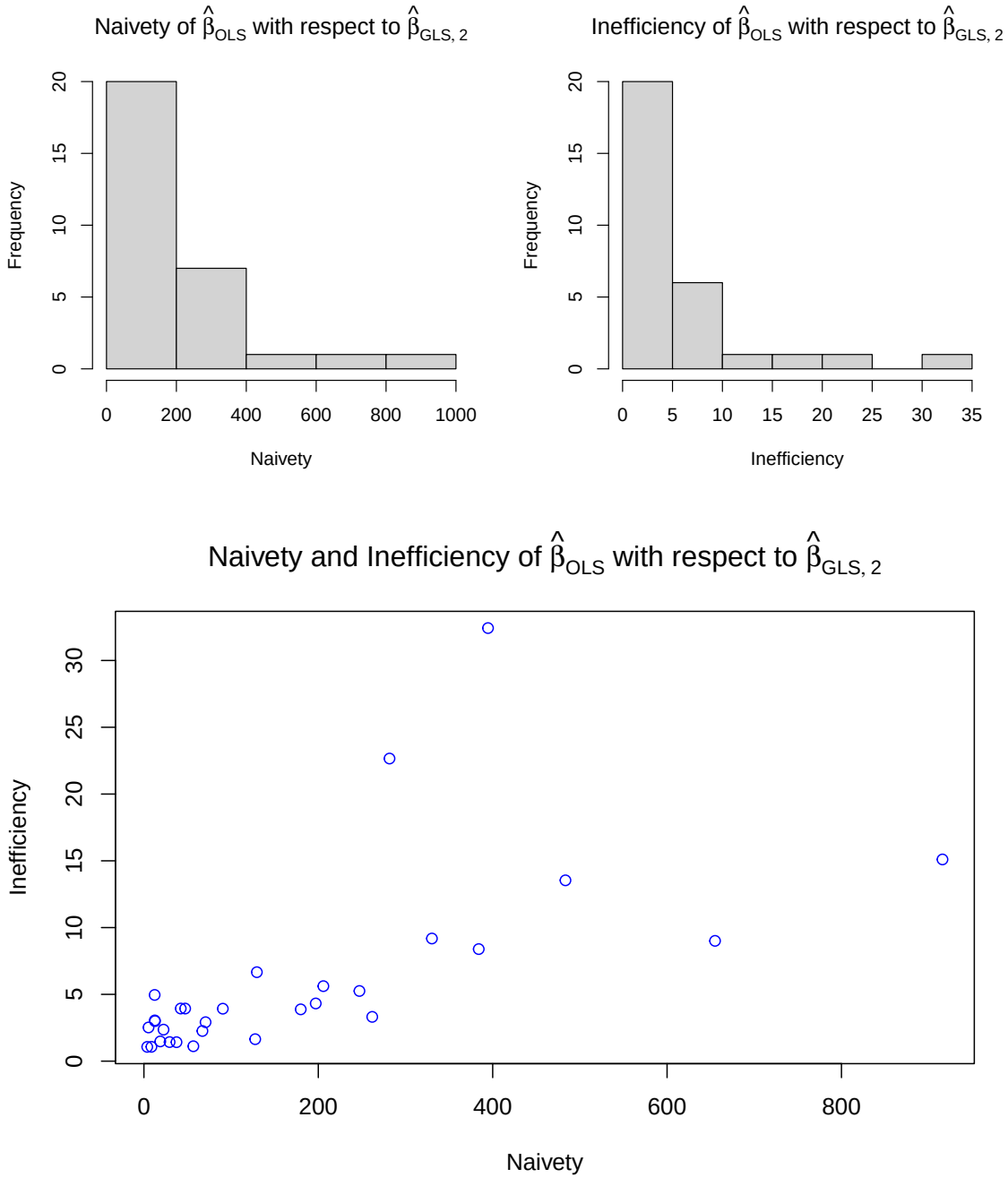


Figure 2.10: Naivety and efficiency of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ with respect to $\hat{\boldsymbol{\beta}}_{\text{GLS},2}$

the variables which are found be significant by both of them.

	$\hat{\boldsymbol{\beta}}_{\text{OLS}}$	$\hat{\boldsymbol{\beta}}_{\text{GLS.0}}$	$\hat{\boldsymbol{\beta}}_{\text{GLS.1}}$	$\hat{\boldsymbol{\beta}}_{\text{GLS.2}}$
Match	5.048***	3.442***	3.296***	3.290***
$\mathbb{I}\{$ client edgy $\}$	0.00102	0.00304	-0.0004	0.0005
$\mathbb{I}\{$ item edgy $\}$	-0.3358***	-0.3515***	-0.3667***	-0.3885***
$\mathbb{I}\{$client edgy $\}$ $\times\mathbb{I}\{$ item edgy $\}$	0.3925***	0.3793***	0.3772***	0.4063***
$\mathbb{I}\{$ client boho $\}$	0.1386***	0.1296***	0.1312***	0.1396***
$\mathbb{I}\{$ item boho $\}$	-0.5499***	-0.6266***	-0.6270***	-0.6599***
$\mathbb{I}\{$client boho $\}$ $\times\mathbb{I}\{$ item boho $\}$	0.3822***	0.3763***	0.3731***	0.4049***
Acrylic	-0.06482***	-0.00536	0.0122	0.0468
Angora	-0.0126	0.07486	0.0978	0.1178
Bamboo	-0.04593	0.03251	0.0730	0.0839
Cashmere	-0.1955***	0.00893	0.0261	0.0576
Cotton	0.1752***	0.1033***	0.1196***	0.1308***
Cupro	0.5979*	0.2089	0.0350	0.0318
Faux Fur	0.2759***	0.2749***	0.2814**	0.2779**
Fur	-0.2021***	-0.07924	-0.0776	-0.0533
Leather	0.2677***	0.1674*	0.1531	0.1557
Linen	-0.3844***	-0.08658	-0.0833	-0.1089
Modal	0.0026	0.1388*	0.1318	0.1397
Nylon	0.0335*	0.08174	0.0899	0.0933
Patent Leather	-0.2359	-0.3764	-0.5266	-0.5448
Pleather	0.4163***	0.3292***	0.3510***	0.3315***
PU	0.416***	0.4579***	0.4509***	0.4618***
PVC	0.6574***	0.9688**	0.8546	0.8812
Rayon	-0.01109***	0.05155***	0.0654***	0.0692***
Silk	-0.1422***	-0.1828***	-0.2186***	-0.2117***
Spandex	-0.3916***	0.414***	0.4080**	0.4290**
Tencel	0.4966***	0.1234*	0.1163	0.1442
Viscose	0.04066***	-0.02259	-0.0336	-0.0370
Wool	-0.0602***	-0.05883	-0.0689	-0.0521

Table 2.1: Each column represents the estimated fixed effects for the *Stitch Fix* data under different random effect regimes: OLS (no random effects), crossed random effects model with random intercepts, and two instances of the crossed random effects model with random intercepts as well as random slopes. "Match" is a continuous variable taking values in the range 0 to 1 representing the prediction of the probability that the client will keep the item purchased (based on an independent baseline model). The indicator variables named "client edgy", "client boho", "item edgy", "item boho" refer to fashion styles that describe some items and some client's preferences. The asterisks used to indicate significance follow the standard R convention, where * represents significance at the 5% level ($p \leq 0.05$), ** at the 1% level ($p \leq 0.01$), and *** at the 0.1% level ($p \leq 0.001$).

2.5.5 Prediction error

We considered a random train-test split of Stitch Fix data. We divide the test data into three categories:

- **Repeat client and repeat item:** The category includes test points with clients whose information was seen in the training data and items whose information was also available in the training data. The prediction of the rating was given by

$$\hat{Y}_{ij} = \mathbf{x}_{ij}^{\mathsf{T}}(\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{a}}_i + \hat{\boldsymbol{b}}_j)$$

  where $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{a}}_i$, and $\hat{\boldsymbol{b}}_j$ are the estimates of $\boldsymbol{\beta}$ and BLUP estimates of $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ respectively.

- **New client and repeat item:** The category includes test points with clients whose information was not seen in the training data and items whose information was available in the training data. The prediction of the rating was given by

$$\hat{Y}_{ij} = \mathbf{x}_{ij}^{\mathsf{T}}(\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{b}}_j)$$

  as the prior expectation for $\boldsymbol{a}_i$ is zero.

- **Repeat client and new item:** The category includes test points with clients whose information was seen in the training data and items whose information was not available in the training data. The prediction of the rating was given by

$$\hat{Y}_{ij} = \mathbf{x}_{ij}^{\mathsf{T}}(\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{a}}_i)$$

  as the prior expectation for $\boldsymbol{b}_j$ is zero.

We observed only two test data points in the third category, so we are only reporting prediction errors in the first two categories. Similar to table 3.1, GLS_0 represents the crossed random effects model with random intercepts, GLS_1 and GLS_2 represent the crossed random effects models with random slopes discussed in Section 2.5.2 and 2.5.2 respectively. For the first category, we observe 99,115

	OLS	GLS_0	GLS_1	GLS_2
Repeat client and repeat item	5.710125	4.875282	4.864025	4.853029
New client and repeat item	5.682390	5.606388	5.543627	5.550773

Table 2.2: Prediction error for different models among different categories

data points and for the second category, we observe 883 data points. GLS_0 provides a significant improvement over OLS in the first category but the improvement is not very significant for the second

category as the information for the clients is not available in the training data. GLS_1 provides a small improvement over GLS_0 in the first category and larger improvement in the second category. GLS_2 provides small improvement over GLS_1 in the first category, although the performance gets worse in the second category. The overall performance is found best for the second model i.e. GLS_2. The results are consistent with the naivety and inefficiency values we discussed before.

2.5.6 Application of crossed random effects model on Movie lens data

In this subsection, we compute the prediction error of crossed random effects model with random slopes on MovieLens dataset and compare it with baseline models like OLS and crossed random effects model with random intercepts. We performed the real data analysis using *MovieLens 100K* dataset available on *kaggle*. The data set describes ratings and free-text tagging activities from MovieLens, a movie recommendation service. It contains around 100,000 ratings from 942 users on 1682 movies. Each user has rated at least 20 movies. The data set also comes with certain information on users and movies, e.g., age, gender of users and genre of the movies. We use the combined information of user and movies as feature vector for linear model, i.e.,

$$\mathbf{x}_{ij} = (\text{age of user } i, \text{gender of user } i, \text{one hot encoding for genre of movie } j).$$

Following remark 1, we use subset of client-specific features as covariates with random slope for items and vice-versa. Therefore, we either use just random intercepts for clients, or we use random intercepts supplemented by random slopes for the one hot encoding vector of genre of the movie. Similarly, we either use just random intercepts for items or we use random intercepts supplemented by random slopes for age and gender of the users. We represent these models by Model_{00}, Model_{01}, Model_{10}, and Model_{11}, i.e., Model_{ij} is a model with random slopes for clients if $i = 1$ and random slopes for items if $j = 1$.

	OLS	Model_{00}	Model_{01}	Model_{10}	Model_{11}
MSE	1.224	0.885	0.878	0.881	0.875

Table 2.3: Prediction error of ratings in MovieLens data

Model_{00} is the crossed model effects model with just random intercepts discussed by Ghosh et al. [2022a] and it serves as another baseline model for us. The purpose of this section was to illustrate the utility of using crossed random effects model with random slopes. As the models with random slopes helps in bringing significant improvement in prediction accuracy, it establishes the need of scalable solution to crossed random effects model. These models could serve as efficient recommender systems and they come with an additional advantage of being interpretable.

2.6 Conclusion

In the paper, we discussed estimation under crossed random effects model with random slopes. We suggested two different algorithms based on the method of moments combined with backfitting and variational EM. The former is most appropriate when the covariance matrices for the random effects are diagonal and the latter is useful when the covariance matrices are unstructured. The algorithm based on the method of moments was shown to have theoretical as well as empirical consistency to estimate fixed effects and covariance parameters when covariance matrices for random effects were diagonal. The algorithm based on variational EM was able to accomplish a more difficult task which is to estimate the parameters when the covariance matrices are unstructured and showed great empirical results. Both of the algorithms could complete the task to estimate fixed and covariance parameters in $\mathcal{O}(N)$ and are superior to "lmer" which takes at least $\mathcal{O}(N^{3/2})$. The algorithms find huge applications in settings with large sample sizes and where the time taken to obtain the fit is crucial. So for example, for a problem with $N = 100,000$ "lmer" might take over 40 minutes, while our methods take under 10 seconds; for problems with $N = 1,000,000$ our methods take around 30 seconds (see Figure 2.2.) We demonstrated the performance of our algorithms using simulations as well as real data.

Chapter 3

Scalable recommender system based on factor analysis

Abstract

Recommender systems have become essential in the modern digital landscape, where personalized content, products, and services significantly enhance user experience. We explore statistical models for recommender systems, focusing on crossed random effects models and factor analysis. We extend the crossed random effects model to include random interactions between users and items using factor analysis. We present a scalable implementation using the Expectation Maximization (EM) and variational EM algorithms for parameter estimation in these models.

3.1 Introduction

Recommender systems play a crucial role in today's tech-driven world, helping users quickly find the right products, content, or services. As online information grows exponentially, these systems must be both efficient and scalable. This paper examines how statistical models, such as crossed random effects and factor analysis, can be applied to develop more effective and scalable recommender systems. We focus on models that handle large datasets with many missing entries, offering methods to improve prediction accuracy and user satisfaction.

We work in a setting where R users rate C items, and only a small fraction of the potential ratings, $N \ll R \times C$, are observed. Additionally, we have information about clients and online content through some covariates, e.g., age and gender of the user, and genre and duration of online content. These covariates could be used to predict ratings for the remaining (client, item) pairs, i.e.,

to suggest new items to existing clients. One statistical model commonly employed in this setting is the crossed random effects model; see, e.g., Baayen et al. [2008] for details. Previous studies, such as Ghosh et al. [2022a], Ghosh et al. [2022b], and Ghandwani et al. [2023], have focused on scalable fitting of these models. They captured user-specific and item-specific behavior through random intercepts for clients and items, offering the advantage of interpretability and enabling inference on parameters that can be used for client-specific and item-specific recommendations.

Ghandwani et al. [2023] expanded on the crossed random effects model with random intercepts by incorporating random slopes into the model, accounting for variability in the effect size of covariates across different users and items. For example, certain clients may love horror movies much more compared to other clients and therefore rate horror movies differently. Ghandwani et al. [2023] captured this information by considering random slopes for clients in each genre category. We refer the readers to section 3.2.1 for further details.

Factor analysis can be viewed as an extension of random slopes, in that it learns additional latent features from the data and their corresponding random slopes. We investigate the application of factor analysis in recommender systems, particularly for matrices with incomplete data, and discuss scalable approaches for parameter estimation using EM algorithms. By addressing these challenges, we aim to improve the predictive accuracy and scalability of recommender systems in real-world applications.

3.2 Related work

Matrix factorization has been a key technique in the development of recommender systems. Koren et al. [2009] introduced matrix factorization methods that significantly improved recommendation accuracy by effectively capturing latent factors in user-item interactions. Mnih and Salakhutdinov [2007] proposed probabilistic matrix factorization (PMF) offering better performance, especially in scenarios with sparse data. Further advancements were made by Mazumder et al. [2010], who developed spectral regularization algorithms for learning from large, incomplete matrices. Their work focused on enhancing the scalability and efficiency of matrix factorization methods, making them more practical for large-scale applications. Candes and Recht [2008] showed that it is possible to recover a low-rank matrix from a small number of observed entries using convex optimization techniques.

These foundational studies have informed the development of more sophisticated and scalable approaches to matrix factorization, which continue to play a crucial role in modern recommender systems.

3.2.1 Crossed random effects model

A number of statistical models can be used for recommender systems, and the crossed random effects model is one of them. Ghosh et al. [2022a] considered the crossed random-effects model incorporating covariate information for the (client, item) pair through the following model:

$$y_{ij} = \beta_0 + a_i + b_j + \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + \varepsilon_{ij}, \tag{3.1}$$

where y_{ij} is the rating given by user i to item j, and $\boldsymbol{x}_{ij}$ is the covariate vector containing information about user i as well as item j. The term $\boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta}$ allows for prediction for existing users/items as well as new users or items. Here a_i is the random intercept for user i, b_j is the random intercept for item j and ε_{ij} is the error term. When there are multiple ratings from the same user, the random intercepts for users help in capturing user specific behaviour. Similarly, when there are multiple ratings for the same item, the random intercepts for items help in capturing item specific behaviour. The reasoning for using random intercepts instead of fixed is that randomness provides a rational way to shrink intercepts towards 0 in cases with sparse data. They made the assumptions that

$$a_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma_a^2), \quad b_j \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma_b^2), \quad \text{and} \quad \varepsilon_{ij} \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma_e^2)$$

where a_i, b_j and ε_{ij} are mutually independent.

Ghandwani et al. [2023] provided a scalable solution for the crossed random effects model, incorporating both random intercepts and random slopes. Typically, random slopes for clients are used for item-specific features, such as genre or duration of online content, while random slopes for items are used for client-specific features, such as age or gender. If $\boldsymbol{x}_{ij}^a$ represents the covariate vector for item-specific features, $\boldsymbol{x}_{ij}^b$ represents the covariate vector for client-specific features, and $\boldsymbol{x}_{ij}$ represents the full covariate vector for client as well as item specific features, the model is given by:

$$y_{ij} = \beta_0 + a_i + b_j + \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + {\boldsymbol{x}_{ij}^a}^{\mathsf{T}}\boldsymbol{u}_i + {\boldsymbol{x}_{ij}^b}^{\mathsf{T}}\boldsymbol{v}_j + \varepsilon_{ij}, \tag{3.2}$$

where $\boldsymbol{u}_i$ is the vector of random effects for clients, and $\boldsymbol{v}_j$ is the vector of random effects for items, with other terms having the same meaning. If the covariate vectors $\boldsymbol{x}_{ij}, \boldsymbol{x}_{ij}^a$ and $\boldsymbol{x}_{ij}^b$ include a constant term, the intercept terms get absorbed within the slope terms and we can simplify the model (3.2) as follows:

$$y_{ij} = \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + {\boldsymbol{x}_{ij}^a}^{\mathsf{T}}\boldsymbol{u}_i + {\boldsymbol{x}_{ij}^b}^{\mathsf{T}}\boldsymbol{v}_j + \varepsilon_{ij}. \tag{3.3}$$

If a feature changes with both the client and the item (e.g., the number of times the client has watched the content or the duration they actually watched) then one could include the feature in both client-specific and item-specific covariate vectors.

In this paper, we aim to learn additional latent features from the data using a factor model. For

instance, we could extend the random slopes model (3.3) to the following model:

$$y_{ij} = \boldsymbol{x}_{ij}^{\mathsf{T}}\beta + \boldsymbol{x}_{ij}^{a\,\mathsf{T}}\boldsymbol{u}_i + \boldsymbol{x}_{ij}^{b\,\mathsf{T}}\boldsymbol{b}_j + \boldsymbol{l}_j^{\mathsf{T}}\boldsymbol{f}_i + \varepsilon_{ij}, \tag{3.4}$$

where $\boldsymbol{l}_j$ is the vector of learned features (a.k.a. factor loadings) for item j and $\boldsymbol{f}_i$ is the vector of random slopes (a.k.a. hidden factor) associated with client i. Before exploring this application of factor analysis in recommender systems, we provide an overview of factor analysis in the following section.

3.3 Factor analysis

Factor analysis (Gorsuch [2014]) is a powerful statistical technique used to simplify complex data sets by identifying underlying factors that explain the relationships among observed variables. By reducing a large number of interrelated variables to a smaller set of unobserved variables which we call factors, this method helps to reveal the hidden structure within the data.

3.3.1 Introduction to factor analysis

To introduce factor analysis, we assume we have complete data; later we relax this restriction. The model aims to explain a set of C variables measured on each of R users using a vector of K common factors ($\boldsymbol{f}_i$), where there are fewer factors per unit than observations per unit ($K < C$). The idea is that the factors $\{\boldsymbol{f}_i\}$ generate the C variables via a factor loading matrix ($\boldsymbol{L} \in \mathbb{R}^{C\times K}$). For a single individual, the model can be represented as:

$$\boldsymbol{y}_i = \boldsymbol{\mu} + \boldsymbol{L}\boldsymbol{f}_i + \boldsymbol{\varepsilon}_i \quad \forall i \in \{1, \cdots, R\}. \tag{3.5}$$

Here $\boldsymbol{y}_i$ is the set of ratings by user i, $\boldsymbol{f}_i$ is the vector of random factors for user i, and $\boldsymbol{\varepsilon}_i$ represents the error made by factor. We usually assume that $\mathrm{Cov}(\boldsymbol{f}_i) = \boldsymbol{I}$ since $\boldsymbol{L}$ can absorb any such scaling. We further assume that $\mathrm{Cov}(\boldsymbol{\varepsilon}_i) = \boldsymbol{\Psi}$ (typically diagonal) and $\boldsymbol{f}_i$ and $\boldsymbol{\varepsilon}$ are uncorrelated. If in addition we assume that $\boldsymbol{f}_i$ and $\boldsymbol{\varepsilon}$ are normally distributed, this implies that $\boldsymbol{y}_i$ has a Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Psi} + \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}}$. The likelihood of the observed data is given by

$$\prod_{i=1}^{R} \frac{1}{\sqrt{|\boldsymbol{\Psi} + \boldsymbol{L}\boldsymbol{L}^T|}} \exp\left(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu})^{\mathsf{T}}\left(\boldsymbol{\Psi} + \boldsymbol{L}\boldsymbol{L}^T\right)^{-1}(\boldsymbol{y}_i - \boldsymbol{\mu})\right). \tag{3.6}$$

There is no closed form expression for the maximum likelihood estimates in general for factor analysis. The Expectation Maximization (EM) algorithm is an efficient approach to obtain the maximum likelihood estimates.

Expectation Maximization approach

The EM algorithm is an iterative approach to find maximum likelihood estimates of parameters θ in models where we have observed data X and unobserved (latent) variables Z. Let $p(X, Z|\theta)$ be the complete data likelihood. Each iteration consists of:

E-step: Compute the expectation of the complete-data log-likelihood:

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{Z|X,\theta^{(t)}} \left[\log p(X, Z \mid \theta)\right].$$

M-step: Maximize this expected log-likelihood with respect to θ:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta \mid \theta^{(t)}).$$

We repeat the process until convergence is reached, providing an estimate of θ that maximizes the likelihood of the observed data X.

With factor models, we treat the factors $\boldsymbol{f}_i$ as missing data. If they were observed, the complete likelihood for the model in (3.5) is given by

$$\begin{aligned}
&\prod_{i=1}^{R} \pi(\boldsymbol{y}_i|\boldsymbol{f}_i) \prod_{i=1}^{R} \pi(\boldsymbol{f}_i) \\
&\quad \propto \prod_{i=1}^{R} \frac{1}{\sqrt{|\boldsymbol{\Psi}|}} \exp\left(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\mu} - \boldsymbol{L}\boldsymbol{f}_i)^{\mathsf{T}} \boldsymbol{\Psi}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu} - \boldsymbol{L}\boldsymbol{f}_i)\right) \exp\left(-\boldsymbol{f}_i^{\mathsf{T}} \boldsymbol{f}_i/2\right).
\end{aligned}$$

Therefore, the complete log-likelihood is given by

$$-\frac{R}{2}\log\left(|\boldsymbol{\Psi}|\right) - \frac{1}{2}\sum_{i=1}^{R}\left[(\boldsymbol{y}_i - \boldsymbol{\mu} - \boldsymbol{L}\boldsymbol{f}_i)^{\mathsf{T}} \boldsymbol{\Psi}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu} - \boldsymbol{L}\boldsymbol{f}_i) + \boldsymbol{f}_i^{\mathsf{T}} \boldsymbol{f}_i\right]. \tag{3.7}$$

The E-step requires computing the expectation of the complete log-likelihood conditional on $\{\boldsymbol{y}_i\}_{i=1}^{R}$, which entails computing $\mathbb{E}[\boldsymbol{f}_i|\boldsymbol{y}_i]$ and $\mathrm{Cov}(\boldsymbol{f}_i \mid \boldsymbol{y}_i)$. Since the joint distribution of $(\boldsymbol{f}_i, \boldsymbol{y}_i)$ is given by

$$\begin{pmatrix} \boldsymbol{f}_i \\ \boldsymbol{y}_i \end{pmatrix} \overset{\text{i.i.d}}{\sim} \mathcal{N}_{K+C}\left(\begin{pmatrix} 0 \\ \boldsymbol{\mu} \end{pmatrix}, \begin{pmatrix} \boldsymbol{I}_K & \boldsymbol{L}^{\mathsf{T}} \\ \boldsymbol{L} & \boldsymbol{\Psi} + \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}} \end{pmatrix}\right), \tag{3.8}$$

standard results give us the conditional distribution of $\boldsymbol{f}_i$ given $\boldsymbol{y}_i$:

$$\boldsymbol{f}_i \mid \boldsymbol{y}_i \sim \mathcal{N}_K\left(\left(\boldsymbol{L}^T \boldsymbol{\Psi}^{-1} \mathbf{L} + \boldsymbol{I}_K\right)^{-1} \mathbf{L}^T \boldsymbol{\Psi}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}), \left(\boldsymbol{L}^T \Psi^{-1} \boldsymbol{L} + \boldsymbol{I}_K\right)^{-1}\right). \tag{3.9}$$

The M-step entails maximizing the conditional expectation of the complete log-likelihood in E-step. The biggest utility comes from the fact that we have closed form expressions for the M-step.

We will provide more details in the next section.

3.3.2 Use of the factor model in recommender systems

The factor model could be used as an effective tool for building recommender systems. Though most of the tools are built for complete matrices, we are here to explore the idea of factor analysis for incomplete matrices. The model we use here is

$$y_{ij} = \boldsymbol{f}_i^\mathsf{T} \boldsymbol{l}_j + \varepsilon_{ij}, \tag{3.10}$$

where $\boldsymbol{f}_i, \boldsymbol{l}_j \in \mathbb{R}^K$. Here, $\boldsymbol{f}_i$ represents the vector of K latent factors for user i, and $\boldsymbol{l}_j$ denotes the loadings (or latent representation) of item j on these same factors. We assume that $\boldsymbol{f}_i \sim \mathcal{N}_K(0, \boldsymbol{I}_K)$ and that the errors are independently distributed as $\varepsilon_{ij} \overset{i.i.d}{\sim} \mathcal{N}(0, \boldsymbol{\Psi}_{jj})$. The primary focus of the model is to predict the rating for user-item pair which is not observed. If we have information on features of the user-item pair i.e., $\boldsymbol{x}_{ij}$, we can include them in the model by considering

$$y_{ij} = \boldsymbol{x}_{ij}^\mathsf{T} \boldsymbol{\beta} + \boldsymbol{f}_i^\mathsf{T} \boldsymbol{l}_j + \varepsilon_{ij}, \tag{3.11}$$

The goal is to provide maximum likelihood estimates of $\boldsymbol{\beta}, \{\boldsymbol{l}_j\}_{j=1}^C$ and $\boldsymbol{\Psi}$. In addition to estimating these parameters, we can also estimate $\mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}]$ as in the E-step for use in prediction. We attempt to provide a scalable solution for the estimation task. As already mentioned $N \ll (R \times C)$, therefore, we want our algorithm to scale with the number of ratings observed.

Let $\boldsymbol{y}_i$ represent the vector of ratings provided by client i (length of $\boldsymbol{y}_i$ is less than or equal to C), $\boldsymbol{X}_{i.}$ represent the covariate matrix corresponding to those ratings and $\boldsymbol{L}_{i.}$ represent the factor loadings concerning the items rated by client i. Rewriting (3.11) in the matrix notation, we have

$$\boldsymbol{y}_i = \boldsymbol{X}_{i.} \boldsymbol{\beta} + \boldsymbol{L}_{i.} \boldsymbol{f}_i + \boldsymbol{\varepsilon}_i. \tag{3.12}$$

Under model (3.12), $\mathbb{E}[\boldsymbol{y}_i] = \boldsymbol{X}_{i.}\boldsymbol{\beta}$ and $\mathrm{Cov}(\boldsymbol{y}_i) = \boldsymbol{L}_{i.}\boldsymbol{L}_{i.}^\mathsf{T} + \boldsymbol{\Psi}_{i.}$ where $\boldsymbol{\Psi}_{i.} = \mathrm{Cov}(\boldsymbol{\varepsilon}_i)$. Then, the marginal distribution of $\boldsymbol{y}_i$ is given by

$$f(\boldsymbol{y}_i) \propto \frac{1}{\sqrt{|\boldsymbol{L}_{i.}\boldsymbol{L}_{i.}^\mathsf{T} + \boldsymbol{\Psi}_{i.}|}} \exp\left(-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{X}_{i.}\boldsymbol{\beta})^\mathsf{T}(\boldsymbol{L}_{i.}\boldsymbol{L}_{i.}^\mathsf{T} + \boldsymbol{\Psi}_{i.})^{-1}(\boldsymbol{y}_i - \boldsymbol{X}_{i.}\boldsymbol{\beta})\right) \tag{3.13}$$

If $\boldsymbol{y}$ denotes the vector of concatenated ratings of all users, the marginal distribution of $\boldsymbol{y}$ is given by the product of the marginal likelihood of $\{f(\boldsymbol{y}_i)\}_{i=1}^R$ in (3.13). The complete likelihood represents the product of the probability density function of unobserved data and probability density function of conditional distribution of observed data given unobserved data. The complete likelihood under

the model (3.11) is given by

$$L := \pi(\boldsymbol{y} \mid \{\boldsymbol{f}_i\}_{i=1}^R) \prod_{i=1}^R \pi(\boldsymbol{f}_i) \tag{3.14}$$

$$= \exp\left(-\frac{1}{2}\sum_{i=1}^R\sum_{j=1}^C z_{ij}\left[\log(\boldsymbol{\Psi}_{jj}) + \frac{(y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T}\boldsymbol{\beta} - \boldsymbol{f}_i^\mathsf{T}\boldsymbol{l}_j)^2}{\boldsymbol{\Psi}_{jj}}\right]\right) \prod_{i=1}^R \exp\left(-\boldsymbol{f}_i^\mathsf{T}\boldsymbol{f}_i/2\right), \tag{3.15}$$

where z_{ij} represents the sparsity pattern of the data with $z_{ij} = 1$ if y_{ij} is observed and zero otherwise. Notice we have used the binary z_{ij} in (3.14) so as to avoid notation for partially observed observation vectors as in (3.13).

Maximizing the complete likelihood above could be seen as equivalent to minimizing a loss, which is very similar to matrix factorization with an additional penalty on $\{\boldsymbol{f}_i\}_{i=1}^R$. For matrix factorization, most of the algorithms alternate between updating model parameters and providing a fit for missing entries. On the other hand, we use the algorithm to estimate the model parameters based only on the observed data. The model parameters could be then used to predict future ratings. For a user-item pairing, we predict the rating by

- $\boldsymbol{x}_{ij}^\mathsf{T}\boldsymbol{\beta} + \boldsymbol{l}_j^\mathsf{T}\mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}]$ if some ratings for both client and item have been observed, and by
- $\boldsymbol{x}_{ij}^\mathsf{T}\boldsymbol{\beta}$ if there is no information for either client or item.

Following Ghosh et al. [2022a], we could also include random intercepts in the model. If we add random intercepts for users, the model can be represented as

$$y_{ij} = \boldsymbol{x}_{ij}^\mathsf{T}\boldsymbol{\beta} + \boldsymbol{f}_i^\mathsf{T}\boldsymbol{l}_j + a_i + \varepsilon_{ij}, \tag{3.16}$$

where $a_i \in \mathbb{R}$, $a_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_a^2)$. The random intercept term a_i can be absorbed into $\boldsymbol{f}_i^\mathsf{T}\boldsymbol{l}_j$ by considering its respective factor loading term to be σ_a and converted into the formulation (3.11).

For the models (3.10), (3.11), and (3.16), we could use the traditional EM algorithm to estimate the parameters. In section 3.4, we discuss the details of the algorithm based on EM to recover maximum likelihood estimates.

In the usual scenarios, there are larger number of users compared to items and therefore we modelled random intercepts for users. In the alternate scenario, when there are larger number of items compared to the number of users; it could be useful to consider random intercepts for items as well. Hence, we consider the model given by

$$y_{ij} = \boldsymbol{x}_{ij}^\mathsf{T}\boldsymbol{\beta} + \boldsymbol{f}_i^\mathsf{T}\boldsymbol{l}_j + a_i + b_j + \epsilon_{ij}, \tag{3.17}$$

where $b_j \in \mathbb{R}$, $b_j \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_b^2)$, and the other terms retain their previous meanings. In the same

scenario, we can also consider factors for items with the model given by

$$y_{ij} = \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + \boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{l}_j + \boldsymbol{g}_i^{\mathsf{T}}\boldsymbol{m}_j + \epsilon_{ij}, \tag{3.18}$$

where $\boldsymbol{m}_j$ are the random factors for items and $\boldsymbol{g}_i^{\mathsf{T}}$ are the corresponding factor loadings for users. For the models (3.17) and (3.18), we shall need to use variational EM to estimate the respective parameters. The reason is that the model would involve crossed random effects given by $\{\boldsymbol{f}_i\}$ and $\{b_j\}$ or $\{\boldsymbol{f}_i\}$ and $\{\boldsymbol{m}_j\}$ and traditional EM would not be scalable. The detailed reasoning for this can be found in Ghandwani et al. [2023]. The variational EM has an advantage of being scalable and it is seen to be empirically consistent. We discuss the details of algorithm based on variational EM in section 3.6.

3.4 EM algorithm

Various algorithms can be used to fit factor models, and the Expectation Maximization (EM) algorithm is one of them. Ghahramani and Hinton [1996] illustrated the use of the EM algorithm to fit the mixture of factor models. In most of the scenarios, factor analysis has been applied to complete matrices. As the target here is to use factor model in a recommender system, we discuss the use of the EM algorithm to fit the factor model on incomplete data matrices. In the following subsection, we illustrate the application of the EM algorithm in the estimation of parameters for the model (3.11). For the EM algorithm, $\{\boldsymbol{f}_i\}_{i=1}^{R}$ is treated as missing data. We can also simplify the algorithm under the assumption $\boldsymbol{\Psi}_{jj} = \sigma_e^2,\ \forall j \in \{1, \cdots, C\}$ if we can assume that the unobserved noise have an equal variance for all items.

3.4.1 E-step

In the E-step, we calculate the expectation of the log-likelihood given the observed data, i.e., $Q := \mathbb{E}[\log L \mid Y]$. The complete-data log-likelihood assumes a Gaussian prior for each factor $\boldsymbol{f}_i$ and a Gaussian likelihood for each observed entry, with observation-specific variance $\boldsymbol{\Psi}_{jj}$. The indicator z_{ij} denotes whether (i, j) is observed. The resulting complete-data log-likelihood is

$$\log L = \left(-\frac{1}{2} \sum_{i=1}^{R} \sum_{j=1}^{C} z_{ij} \left[\log(\boldsymbol{\Psi}_{jj}) + \frac{(y_{ij} - \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} - \boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{l}_j)^2}{\boldsymbol{\Psi}_{jj}} \right] - \sum_{i=1}^{R} \frac{1}{2} \boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{f}_i \right). \tag{3.19}$$

To compute Q, we need to compute the conditional distribution of missing data given the observed data, i.e.,the posterior of the latent factors:

$$\pi(\boldsymbol{f}_1, \cdots, \boldsymbol{f}_R \mid \boldsymbol{y}) \propto \mathbb{P}(\boldsymbol{y}, \{\boldsymbol{f}_i\}_{i=1}^{R}).$$

Because both the prior and likelihood are Gaussian, the conditional posterior for each $\boldsymbol{f}_i$ is also multivariate normal.

$$\begin{aligned}
\pi(\boldsymbol{f}_1, \cdots, \boldsymbol{f}_R \mid \boldsymbol{y}) &\propto \mathbb{P}(\boldsymbol{y}, \{\boldsymbol{f}_i\}_{i=1}^R) \\
&\propto \exp\left\{ -\frac{1}{2} \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{z_{ij}(y_{ij} - \boldsymbol{x}_{ij}^\top \boldsymbol{\beta} - \boldsymbol{f}_i^\top \boldsymbol{l}_j)^2}{\boldsymbol{\Psi}_{jj}} - \frac{1}{2} \sum_{i=1}^{R} \boldsymbol{f}_i^\top \boldsymbol{f}_i \right\} \\
&\propto \exp\left\{ -\frac{1}{2} \sum_{i=1}^{R} \boldsymbol{f}_i^\top \left(\boldsymbol{I}_k + \sum_{j=1}^{C} \frac{z_{ij} \boldsymbol{l}_j \boldsymbol{l}_j^\top}{\boldsymbol{\Psi}_{jj}} \right) \boldsymbol{f}_i \right. \\
&\qquad \left. + \boldsymbol{f}_i^\top \sum_{j=1}^{C} \frac{z_{ij}(y_{ij} - \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}) \boldsymbol{l}_j}{\boldsymbol{\Psi}_{jj}} \right\}.
\end{aligned}$$

Therefore, $\boldsymbol{f}_i$ follow a normal distribution conditional on $\boldsymbol{y}$ with

$$\mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}] = \left(\boldsymbol{I}_k + \sum_{j=1}^{C} \frac{z_{ij} \boldsymbol{l}_j \boldsymbol{l}_j^\top}{\boldsymbol{\Psi}_{jj}} \right)^{-1} \left(\sum_{j=1}^{C} \frac{z_{ij}(y_{ij} - \boldsymbol{x}_{ij}^\top \boldsymbol{\beta}) \boldsymbol{l}_j}{\boldsymbol{\Psi}_{jj}} \right) \tag{3.20}$$

and

$$\mathrm{Var}[\boldsymbol{f}_i \mid \boldsymbol{y}] = \left(\boldsymbol{I}_k + \sum_{j=1}^{C} \frac{z_{ij} \boldsymbol{l}_j \boldsymbol{l}_j^\top}{\boldsymbol{\Psi}_{jj}} \right)^{-1}. \tag{3.21}$$

The expectation of the complete log-likelihood is given by

$$\begin{aligned}
Q = -\frac{1}{2} \sum_{i=1}^{R} \sum_{j=1}^{C} z_{ij} \mathbb{E}\left[\log(\boldsymbol{\Psi}_{jj}) + \frac{(y_{ij} - \boldsymbol{x}_{ij}^\top \boldsymbol{\beta} - \boldsymbol{f}_i^\top \boldsymbol{l}_j)^2}{\boldsymbol{\Psi}_{jj}} \mid \boldsymbol{y} \right] \\
- \frac{1}{2} \sum_{i=1}^{R} \mathbb{E}\left[\boldsymbol{f}_i^\top \boldsymbol{f}_i \mid \boldsymbol{y} \right] + \text{const.}
\end{aligned}$$

3.4.2 M-step

In the M-step, we update the estimates of $\boldsymbol{\beta}, \{\boldsymbol{l}_j\}_{j=1}^C$ and $\boldsymbol{\Psi}$ by maximizing Q. Setting the partial derivates to zero,

$$\frac{\partial}{\partial \boldsymbol{\beta}} Q = 0, \; \frac{\partial}{\partial \boldsymbol{l}_j} Q = 0, \text{ and } \frac{\partial}{\partial \boldsymbol{\Psi}_{jj}} Q = 0 \, \forall j \in \{1, \cdots, C\}. \tag{3.22}$$

This yields closed-form updates:

$$\boldsymbol{\beta} = \left(\sum_{j=1}^{C} \sum_{i=1}^{R} z_{ij} \boldsymbol{x}_{ij} \boldsymbol{x}_{ij}^\top \right)^{-1} \left(\sum_{j=1}^{C} \sum_{i=1}^{R} z_{ij} \mathbb{E}[y_{ij} - \boldsymbol{f}_i^\top \boldsymbol{l}_j \mid \boldsymbol{y}] \boldsymbol{x}_{ij} \right), \tag{3.23}$$

$$\boldsymbol{l}_j = \left(\sum_{i=1}^{R} z_{ij} \mathbb{E}[\boldsymbol{f}_i \boldsymbol{f}_i^\mathsf{T} \mid \boldsymbol{y}] \right)^{-1} \left(\sum_{i=1}^{R} z_{ij} (y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T} \boldsymbol{\beta}) \mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}] \right), \tag{3.24}$$

and

$$\begin{aligned} \boldsymbol{\Psi}_{jj} &= \sum_{i=1}^{R} \frac{z_{ij} \mathbb{E}\left[(y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T} \boldsymbol{\beta} - \boldsymbol{f}_i^\mathsf{T} \boldsymbol{l}_j)^2 \mid \boldsymbol{y} \right]}{\sum_{i=1}^{R} z_{ij}} \\ &= \sum_{i=1}^{R} \frac{z_{ij} (y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T} \boldsymbol{\beta})^2 + z_{ij} \boldsymbol{l}_j^\mathsf{T} \mathbb{E}[\boldsymbol{f}_i \boldsymbol{f}_i^\mathsf{T} \mid \boldsymbol{y}] \boldsymbol{l}_j - 2 z_{ij} (y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T} \boldsymbol{\beta}) \boldsymbol{l}_j^\mathsf{T} \mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}]}{\sum_{i=1}^{R} z_{ij}}. \end{aligned}$$

Equation 3.24 gives us,

$$\left(\sum_{i=1}^{R} z_{ij} \mathbb{E}[\boldsymbol{f}_i \boldsymbol{f}_i^\mathsf{T} \mid \boldsymbol{y}] \right) \boldsymbol{l}_j = \sum_{i=1}^{R} z_{ij} (y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T} \boldsymbol{\beta}) \mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}].$$

Therefore,

$$\boldsymbol{\Psi}_{jj} = \sum_{i=1}^{R} \frac{z_{ij} (y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T} \boldsymbol{\beta})^2 - z_{ij} \boldsymbol{l}_j^\mathsf{T} \mathbb{E}[\boldsymbol{f}_i \boldsymbol{f}_i^\mathsf{T} \mid \boldsymbol{y}] \boldsymbol{l}_j}{\sum_{i=1}^{R} z_{ij}}. \tag{3.25}$$

Both the E-step and M-step can be computed in $\mathcal{O}(N)$ time per iteration. This follows because all required expectations and parameter updates reduce to sums over the observed entries (i, j), each involving only local quantities for the corresponding user or item. Specifically, the E-step computes a conditional mean and variance for each $\boldsymbol{f}_i$ by summing over the items j for which $z_{ij} = 1$, followed by inversion of a fixed-size $K \times K$ matrix. Likewise, the M-step updates for $\boldsymbol{\beta}$, $\{\boldsymbol{l}_j\}$, and $\{\Psi_{jj}\}$ consist of sums over the same (i, j) pairs and small matrix inversions for each item or coefficient block. By updating each parameter block conditionally (backfitting), no global matrix operations scale worse than linear in the total number of observations N.

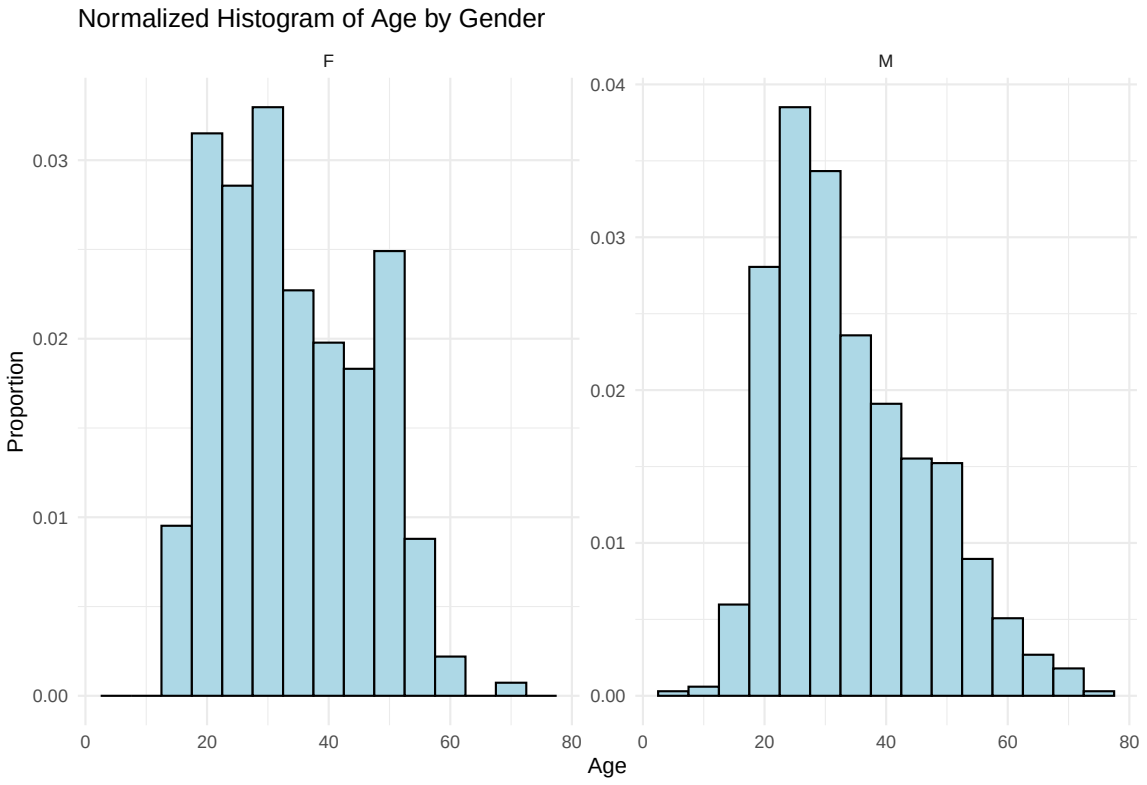
Algorithm 1 EM algorithm for factor analysis model (3.11)

Require: $\boldsymbol{y} = \{y_{ij}\}$, $\boldsymbol{X} = \{\boldsymbol{x}_{ij}\}$ and dimension of factor features k.
Initialize $\boldsymbol{\beta} \sim \mathcal{N}_p(0, \boldsymbol{I})$ and $\boldsymbol{l}_j \sim \mathcal{N}_k(0, \boldsymbol{I})$, $\forall j \in \{1, \cdots, C\}$
Initialise $\boldsymbol{\Psi}_{jj} \sim \chi_1^2$, $\forall\, j \in \{1, \cdots, C\}$.
while $\boldsymbol{\beta}$, $\{\boldsymbol{l}_j\}_{j=1}^C$ and $\{\boldsymbol{\Psi}_{jj}\}_{j=1}^C$ have not converged **do**
  Compute $\mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}]$ and $\mathrm{Var}[\boldsymbol{f}_i \mid \boldsymbol{y}]$ by (3.20) and (eq. (3.21)).
  Update $\boldsymbol{\beta}$ based on current estimates of $\{\boldsymbol{l}_j\}_{j=1}^C$ and $\{\mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}]\}_{i=1}^R$ using (3.23).
  Update $\{\boldsymbol{l}_j\}_{j=1}^C$ based on current estimates of $\boldsymbol{\beta}$, $\{\mathbb{E}[\boldsymbol{f}_i \mid \boldsymbol{y}]\}_{i=1}^R$ and $\{\mathbb{E}[\boldsymbol{f}_i \boldsymbol{f}_i^\mathsf{T} \mid \boldsymbol{y}]\}_{i=1}^R$ using (3.24).
  Update $\{\boldsymbol{\Psi}_{jj}\}_{j=1}^C$ based on current estimates of $\boldsymbol{\beta}$, $\{\boldsymbol{l}_j\}_{j=1}^C$ and $\{E[\boldsymbol{f}_i \boldsymbol{f}_i^\mathsf{T} \mid \boldsymbol{y}]\}_{i=1}^R$ using (3.25).
end while

3.5 Recommendation on Movie Lens

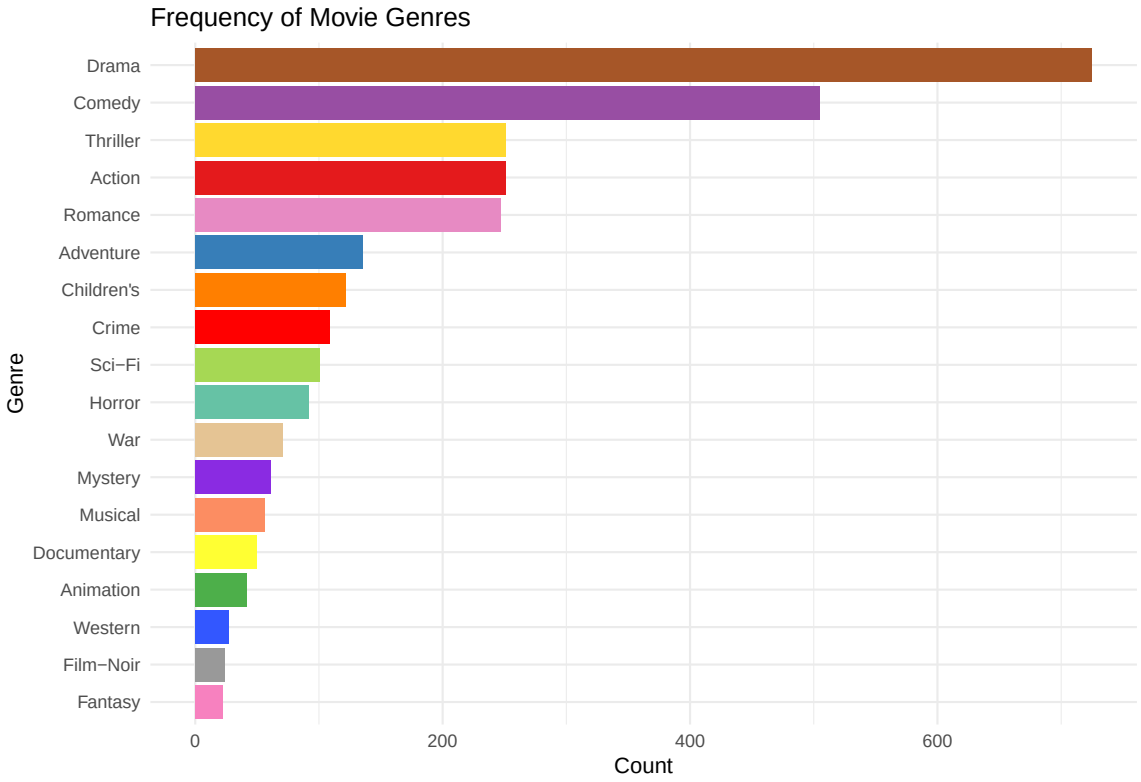
We conducted a real data analysis using "Movie lens 100K dataset" which available on *kaggle*. The dataset describe ratings and free-text tagging activities from MovieLens, a movie recommendation service. It contains approximately 100,000 ratings from 942 users on 1,682 movies. Each user has rated at least 20 movies. The dataset also provides user and movie-related information, such as users' age, gender, occupation, and the genre of the movies. Figures 3.1 and 3.2 helps in visualization of the covariates of the data.

Figure 3.1: Histogram of age for females and males



We used the combined user and movie information as features for the linear model, i.e., $x_{ij} = (\text{age}_i, \text{gender}_i, \text{one hot vector for genre}_j)$. The dataset comes with five splits of the dataset into training and test datasets. We use the same splits to compare our algorithm on factor analysis with and without random intercepts to the softImpute algorithm developed by Hastie et al. [2015] and the baseline intercept model (3.1).

Figure 3.2: Frequency of genre of movies



The **softImpute** algorithm minimizes the following loss:

$$\sum_{i=1}^{R}\sum_{j=1}^{C}\left(y_{ij}-a_i-b_j-\boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{l}_j\right)^2+\lambda\left(\boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{f}_i+\boldsymbol{l}_j^{\mathsf{T}}\boldsymbol{l}_j\right), \tag{3.26}$$

where $a_i, b_j \in \mathbb{R}$ and $\boldsymbol{f}_i, \boldsymbol{l}_j \in \mathbb{R}^K$. The **hardImpute** algorithm minimizes the same loss with λ fixed to zero. See Hastie et al. [2015] for details.

Minimizing the softImpute loss in Equation (3.26) is equivalent to modeling the outcome variable as Gaussian conditional on $\boldsymbol{f}_i$ and $\boldsymbol{l}_j$, with Gaussian priors on the marginal distributions of $\{\boldsymbol{f}_i\}$ and $\{\boldsymbol{l}_j\}$, as follows:

$$\text{Given } \boldsymbol{f}_i \text{ and } \boldsymbol{l}_j, \quad y_{ij} \overset{i.i.d}{\sim} \mathcal{N}(a_i+b_j+\boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{l}_j,\, \lambda\sigma^2), \quad \text{and} \quad \boldsymbol{f}_i, \boldsymbol{l}_j \overset{i.i.d}{\sim} \mathcal{N}(0,\, \sigma^2 I_K).$$

Under these assumptions, the marginal distribution of y_{ij} is no longer Gaussian. However, we prefer models where the outcome variable remains marginally Gaussian. To this end, we consider models in which the loading vectors $\{\boldsymbol{l}_j\}$ are treated as fixed but unknown, while the latent factors

$\{\boldsymbol{f}_i\}$ follow Gaussian distributions, ensuring that the outcome is marginally Gaussian. Applying the concepts developed in this paper, and utilizing the EM and variational EM algorithms, we fit the following models on the training data:

1. **Model with only random client intercepts:**

$$y_{ij} = \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + a_i + \varepsilon_{ij}, \quad i = 1, \ldots, R, \quad j = 1, \ldots, C.$$

2. **Model with random client and item intercepts:**

$$y_{ij} = \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + a_i + b_j + \varepsilon_{ij}, \quad i = 1, \ldots, R, \quad j = 1, \ldots, C.$$

3. **Factor model without random intercepts:**

$$y_{ij} = \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + \boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{l}_j + \varepsilon_{ij}, \quad i = 1, \ldots, R, \quad j = 1, \ldots, C.$$

4. **Factor model with random client intercepts:**

$$y_{ij} = \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + \boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{l}_j + a_i + \varepsilon_{ij}, \quad i = 1, \ldots, R, \quad j = 1, \ldots, C.$$

5. **Factor model with random client and item intercepts:**

$$y_{ij} = \boldsymbol{x}_{ij}^{\mathsf{T}}\boldsymbol{\beta} + \boldsymbol{f}_i^{\mathsf{T}}\boldsymbol{l}_j + a_i + b_j + \varepsilon_{ij}, \quad i = 1, \ldots, R, \quad j = 1, \ldots, C.$$

In all of the above models, we assume that client-specific intercepts $a_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_a^2)$, item-specific intercepts $b_j \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_b^2)$, hidden factors $\boldsymbol{f}_i \overset{i.i.d}{\sim} \mathcal{N}(0, I_K)$, and stochastic errors $\varepsilon_{ij} \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_e^2)$. Model parameters thus include $\sigma_a^2, \sigma_b^2, \sigma_e^2$, fixed effects $\boldsymbol{\beta}$, and factor loadings $\{\boldsymbol{l}_j\}_{j=1}^{C}$.

Since the softImpute and hardImpute algorithms do not use covariate information, we performed a linear regression of ratings on the covariates and applied the hardImpute algorithm to the residuals to ensure valid comparisons.

For fitting the model (5), we use Variational EM algorithm which we haven't discussed so far and we discuss in detail in the next section.

We use the mean squared error (MSE) of the predicted ratings as the metric for model comparison. The experiments are fully replicable using the provided random seed values. For the HardImpute method ($\lambda = 0$), we vary the number of latent factors K from 1 to 3. Similarly, for Models 3, 4, and 5, we explore latent dimensions in the same range. For each model, we report the results corresponding to the configuration that achieves the lowest MSE in Table 3.1. For Model 3, the lowest MSE is attained with $K = 3$ latent factors, while for Model 4, the minimum typically

Split	HardImpute	Model 1	Model 2	Model 3	Model 4	Model 5
First	0.8900	1.0930	0.9064	0.8930	0.8786	0.8639
Second	0.8794	1.0640	0.8890	0.8825	0.8693	0.8641
Third	0.8832	1.0347	0.8759	0.8660	0.8426	0.8422
Fourth	0.8688	1.0411	0.8748	0.8526	0.8469	0.8437
Fifth	0.8598	1.0439	0.8747	0.8497	0.8477	0.8505
Average	0.8762	1.0553	0.8842	0.8708	0.8570	0.8529

Table 3.1: Prediction accuracy (MSE) of different models on the MovieLens dataset. The red value highlights the model achieving the best prediction accuracy for each split. Each split represents a different training and test split of the MovieLens dataset.

occurs at $K = 2$ for most data splits. The results for Model 5 are consistent with Model 4, suggesting that the inclusion of random intercepts may compensate for one of the latent factors. Figures 3.3 and 3.4 further illustrate the prediction accuracy and the distribution of predictions from Model 5 across the true rating levels. These plots visually confirm the good calibration and slightly improved fit of Model 5 relative to HardImpute.

3.6 Variational EM algorithm

Variational EM, following Beal and Ghahramani [2003], extends the classical EM framework to settings where the missing data have a nontrivial graphical structure, such as in models (3.16) and (3.17). In these models, the latent factors $\boldsymbol{f}_i$ and the random intercepts a_i and b_j are correlated conditional on the observations $\boldsymbol{y}$, which makes the exact posterior intractable.

Since model (3.17) generalizes (3.16), we illustrate the algorithm for (3.17). For simplicity, we assume homoscedastic noise, i.e., $\boldsymbol{\Psi}_{jj} = \sigma_e^2$ for all j, but the same approach extends directly to the heteroscedastic case.

The complete-data log-likelihood can be written as:

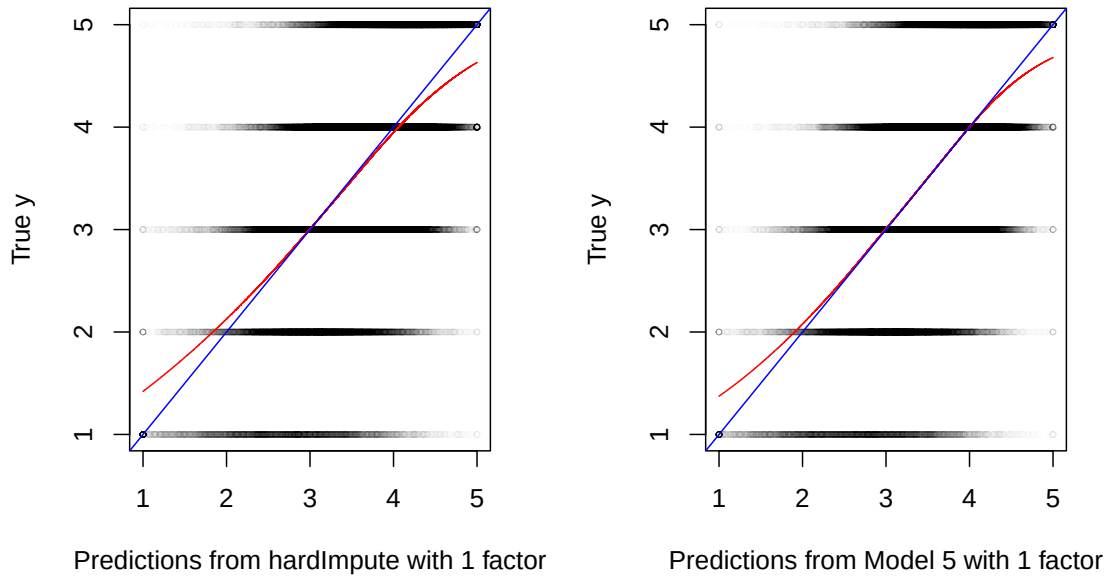
$$\begin{aligned} L(\cdot) = & -\frac{1}{2}\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}\Big[\log(\sigma_e^2) + \frac{(y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T}\boldsymbol{\beta} - \boldsymbol{f}_i^\mathsf{T}\boldsymbol{l}_j - a_i - b_j)^2}{\sigma_e^2}\Big] \\ & -\frac{1}{2}\sum_{i=1}^{R}\boldsymbol{f}_i^\mathsf{T}\boldsymbol{f}_i - \frac{1}{2\sigma_a^2}\sum_{i=1}^{R}a_i^2 - \frac{1}{2\sigma_b^2}\sum_{j=1}^{C}b_j^2 + \text{const.} \end{aligned}$$

The marginal log-likelihood of the observed data is then:

$$l(\boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) = \int L(\cdot)\, d\boldsymbol{f}_i\, da_i\, db_j.$$

Since this integral is intractable due to the dependence structure, we introduce a variational distribution to approximate the posterior $\mathbb{P}(\{\boldsymbol{f}_i\}, \{a_i\}, \{b_j\} \mid \boldsymbol{y})$ with a tractable family $q(\{\boldsymbol{f}_i\}, \{a_i\}, \{b_j\})$.

Figure 3.3: Visualisation of prediction accuracy of HardImpute algorithm and predictions by Model 5. The red curve shows a smoothed fit of the predicted ratings against the true ratings, while the blue line represents the ideal reference line $x = y$. The prediction accuracy of the two models is visually very similar, with Model 5 having a slight edge.



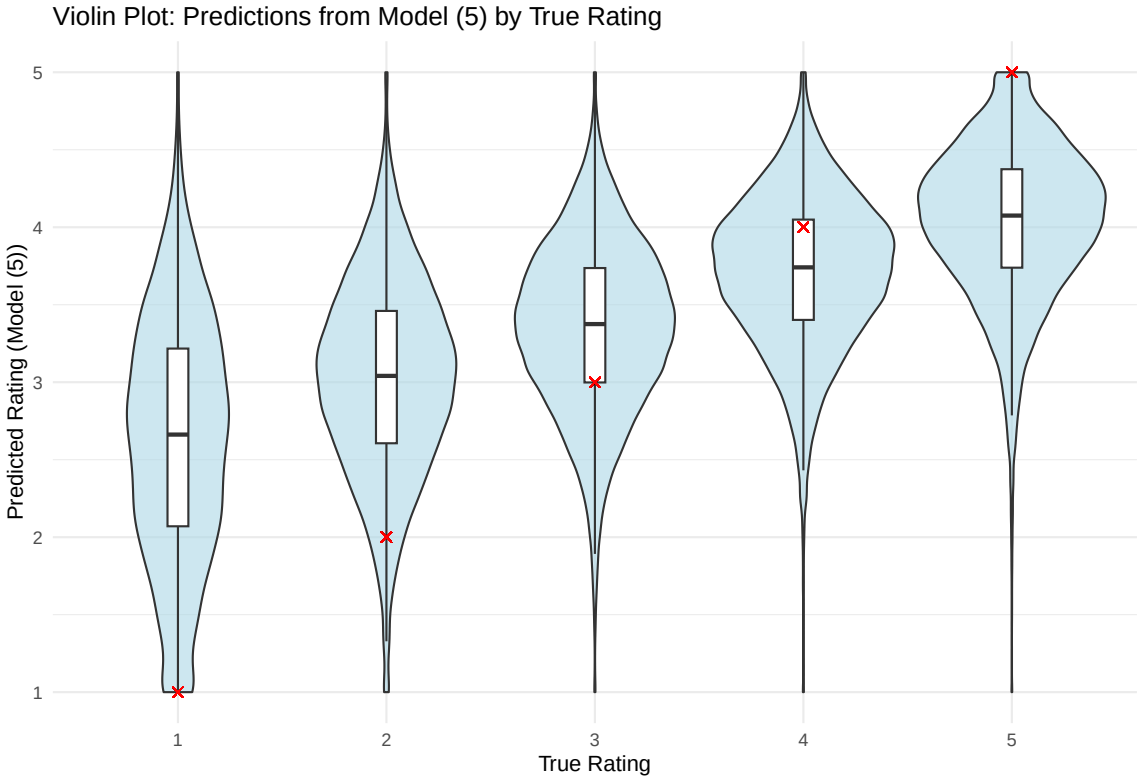
The variational objective is then

$$F(q, \boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) = \int q(\boldsymbol{f}_i, a_i, b_j) \log \frac{\mathbb{P}(\boldsymbol{y}, \boldsymbol{f}_i, a_i, b_j \mid \boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2)}{q(\boldsymbol{f}_i, a_i, b_j)} \, d\boldsymbol{f}_i \, da_i \, db_j.$$

This quantity defines a lower bound on the marginal log-likelihood:

$$\begin{aligned} F(q, \boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) &= l(\boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) - \mathrm{KL}\big(q \,\|\, \mathbb{P}(\boldsymbol{f}_i, a_i, b_j \mid \boldsymbol{y}, \ldots)\big) \\ &\leq l(\boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2). \end{aligned}$$

Figure 3.4: Violin plot showing the distribution of predictions from Model 5 at each level of the outcome variable (True Rating). The plot shows that the mean predicted rating conditional on the true rating steadily increases with the true rating, and that predictions are approximately normally distributed around the true rating.



To make this tractable, we assume the factorized mean-field variational family:

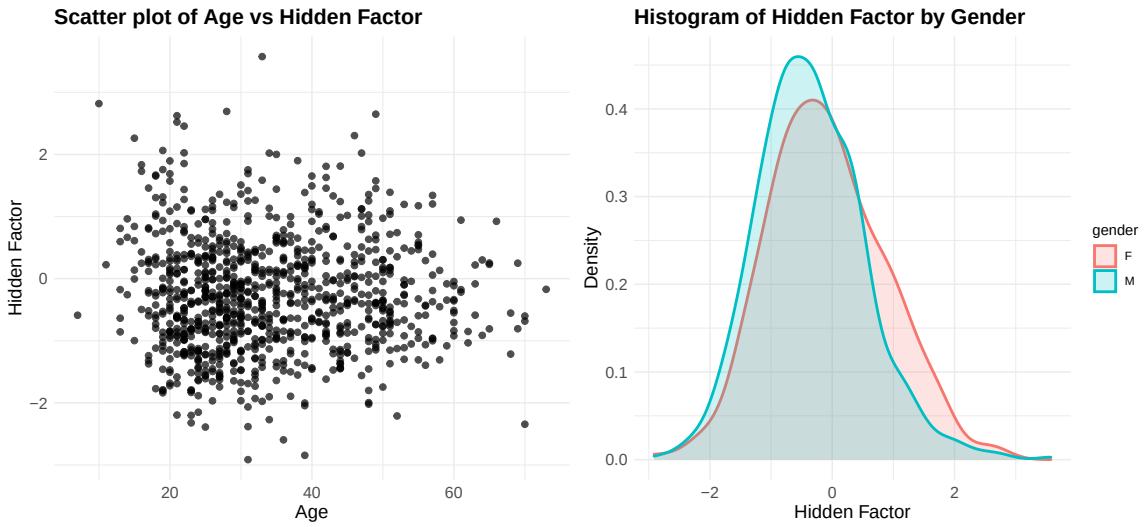
$$Q = \Big\{ q : q(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_R, a_1, \ldots, a_R, b_1, \ldots, b_C) \\ = q_f(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_R)\; q_a(a_1, \ldots, a_R)\; q_b(b_1, \ldots, b_C) \Big\}.$$

(One could assume only partial factorization, e.g., that $\{\boldsymbol{f}_i, a_i\}$ are independent of $\{b_j\}$; however, the fully factorized structure illustrates how the method extends to multiple independent components.)

In the exponential family setting, the optimal variational update for each block follows directly from Beal and Ghahramani [2003]:

$$q_u^{(k)}(\alpha_u) \propto \mathbb{P}\big(Y, \alpha_u \mid \alpha_{-u} = \mathbb{E}_{q^{(k-1)}}(\alpha_{-u})\big).$$

Figure 3.5: This figure examines how the latent factor estimated by Model 5 relates to the covariates Age and Gender. The scatter plot displays the joint relationship between the hidden factor and Age, showing no clear dependence. The histogram shows the conditional distribution of the factor by gender, revealing nearly identical distributions for both groups. Together, this suggests that the hidden factor captures latent user characteristics distinct from Age and Gender.



The Variational EM algorithm alternates between updating the variational distribution and the model parameters in a way that guarantees monotonic improvement of the bound:

$$F(q^{(k-1)}, \theta^{(k-1)}) \leq F(q^{(k)}, \theta^{(k-1)}) \leq F(q^{(k)}, \theta^{(k)}).$$

Thus, Variational EM maximizes a lower bound to the marginal likelihood.

3.6.1 Variational E-step

In the Variational E-step, we compute the expectation of the complete log likelihood under the optimal factorized distributions for $\{\boldsymbol{f}_i\}$, $\{a_i\}$, and $\{b_j\}$.

The posterior factorizes as:

$$q_f^{(k)}(\boldsymbol{f}_1, \cdots, \boldsymbol{f}_R) = \prod_{i=1}^{R} q^{(k)}(\boldsymbol{f}_i),$$

$$q_a^{(k)}(a_1, \cdots, a_R) = \prod_{i=1}^{R} q^{(k)}(a_i), \quad \text{and} \quad q_b^{(k)}(b_1, \cdots, b_C) = \prod_{j=1}^{C} q^{(k)}(b_j)$$

with each component Gaussian:

$$q^{(k)}(\boldsymbol{f}_i) = N(\boldsymbol{\mu}_{\boldsymbol{f},\boldsymbol{i}}^{(\boldsymbol{k})}, \boldsymbol{\Sigma}_{\boldsymbol{f},\boldsymbol{i}}^{(\boldsymbol{k})}),$$

$$q^{(k)}(a_i) = N(\mu_{a,i}^{(k)}, {\sigma^2}_{a,i}^{(k)}), \quad \text{and} \quad q^{(k)}(b_j) = N(\mu_{b,j}^{(k)}, {\sigma^2}_{b,j}^{(k)}),$$

where

$$\mu_{a,i}^{(k)} = {\sigma^2}_{a,i}^{(k)} \left(\frac{\sum_{j=1}^{C} z_{ij} \big(y_{ij} - \boldsymbol{x}_{ij}^{\mathsf{T}} \boldsymbol{\beta}^{(k-1)} - \boldsymbol{\mu}_{\boldsymbol{f},\boldsymbol{i}}^{(\boldsymbol{k-1})^{\mathsf{T}}} \boldsymbol{l}_j^{(k-1)} - \mu_{b,j}^{(k-1)} \big)}{\left(\sigma_e^{(k-1)} \right)^2} \right), \tag{3.27}$$

$$\mu_{b,j}^{(k)} = {\sigma^2}_{b,j}^{(k)} \left(\frac{\sum_{i=1}^{R} z_{ij} \big(y_{ij} - \boldsymbol{x}_{ij}^{\mathsf{T}} \boldsymbol{\beta}^{(k-1)} - \boldsymbol{\mu}_{\boldsymbol{f},\boldsymbol{i}}^{(\boldsymbol{k-1})^{\mathsf{T}}} \boldsymbol{l}_j^{(k-1)} - \mu_{a,i}^{(k)} \big)}{\left(\sigma_e^{(k-1)} \right)^2} \right) \tag{3.28}$$

and

$${\sigma^2}_{a,i}^{(k)} = \left(\frac{\sum_{j=1}^{C} z_{ij}}{\left(\sigma_e^{(k-1)} \right)^2} + \frac{1}{\left(\sigma_a^{(k-1)} \right)^2} \right)^{-1}, \tag{3.29}$$

$${\sigma^2}_{b,j}^{(k)} = \left(\frac{\sum_{i=1}^{R} z_{ij}}{\left(\sigma_e^{(k-1)} \right)^2} + \frac{1}{\left(\sigma_b^{(k-1)} \right)^2} \right)^{-1}. \tag{3.30}$$

Also,

$$\boldsymbol{\mu}_{\boldsymbol{f},\boldsymbol{i}}^{(\boldsymbol{k})} = \boldsymbol{\Sigma}_{\boldsymbol{f},\boldsymbol{i}}^{(\boldsymbol{k})} \left(\frac{\sum_{j=1}^{C} z_{ij} \big(y_{ij} - \boldsymbol{x}_{ij}^{\mathsf{T}} \boldsymbol{\beta}^{(k-1)} - \mu_{a,i}^{(k)} - \mu_{b,j}^{(k)} \big) \boldsymbol{l}_j^{(k-1)}}{\left(\sigma_e^{(k-1)} \right)^2} \right) \tag{3.31}$$

and

$$\boldsymbol{\Sigma}_{\boldsymbol{f},\boldsymbol{i}}^{(\boldsymbol{k})} = \left(\frac{\sum_{j=1}^{C} z_{ij} \boldsymbol{l}_j^{(k-1)} \boldsymbol{l}_j^{(k-1)}}{\left(\sigma_e^{(k-1)} \right)^2} + \boldsymbol{I}_{K+1} \right)^{-1}. \tag{3.32}$$

This shows that each posterior mean and variance depends only on local sums over the observed pairs (i, j).

3.6.2 M-step

In the M-step, we maximize the expected log-likelihood computed under the current variational posterior to update the fixed effects $\boldsymbol{\beta}$, the item loadings $\{\boldsymbol{l}_j\}$, and the variance components σ_a^2, σ_b^2, and σ_e^2. Since the objective is convex in these blocks, this reduces to equating the derivative with respect to each parameter to zero, i.e.,

$$\frac{\partial}{\partial \boldsymbol{\beta}} F(q, \boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) = 0, \quad \frac{\partial}{\partial \boldsymbol{l}_j} F(q, \boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) = 0$$

$$\frac{\partial}{\partial \sigma_a^2} F(q, \boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) = 0$$

$$\frac{\partial}{\partial \sigma_b^2} F(q, \boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) = 0, \quad \frac{\partial}{\partial \sigma_e^2} F(q, \boldsymbol{\beta}, \{\boldsymbol{l}_j\}, \sigma_a^2, \sigma_b^2, \sigma_e^2) = 0$$

Therefore, fresh estimates of the parameters in the M-step are given by

$$\hat{\boldsymbol{\beta}} = \left(\sum_{j=1}^{C} \sum_{i=1}^{R} z_{ij} \boldsymbol{x}_{ij} \boldsymbol{x}_{ij}^\mathsf{T} \right)^{-1} \left(\sum_{j=1}^{C} \sum_{i=1}^{R} z_{ij} \left(y_{ij} - \boldsymbol{\mu}_{\boldsymbol{f},\boldsymbol{i}}^{(\boldsymbol{k})\mathsf{T}} \boldsymbol{l}_j^{(k)} - \mu_{a,i}^{(k)} - \mu_{b,j}^{(k)} \right) \boldsymbol{x}_{ij} \right) \tag{3.33}$$

$$\hat{\boldsymbol{l}}_j = \left(\sum_{i=1}^{R} z_{ij} \mathbb{E}_q[\boldsymbol{f}_i \boldsymbol{f}_i^\mathsf{T} \mid \boldsymbol{y}] \right)^{-1} \left(\sum_{i=1}^{R} z_{ij} \left(y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T} \hat{\boldsymbol{\beta}}^{(k)} - \mu_{a,i}^{(k)} - \mu_{b,j}^{(k)} \right) \mathbb{E}_q \left[\boldsymbol{f}_i \mid \boldsymbol{y} \right] \right) \tag{3.34}$$

$$\hat{\sigma}_a^2 = \frac{1}{R} \sum_{i=1}^{R} \mathbb{E}_q[a_i a_i^\mathsf{T}] = \frac{1}{R} \sum_{i=1}^{R} \left(\mu_{a,i}^{(k)} \mu_{a,i}^{(k)\mathsf{T}} + \Sigma_{a,i}^{(k)} \right) \tag{3.35}$$

$$\hat{\sigma}_b^2 = \frac{1}{C} \sum_{j=1}^{C} \mathbb{E}_q[b_j b_j^\mathsf{T}] = \frac{1}{C} \sum_{j=1}^{C} \left(\mu_{b,j}^{(k)} \mu_{b,j}^{(k)\mathsf{T}} + \Sigma_{b,j}^{(k)} \right) \tag{3.36}$$

$$\hat{\sigma}_e^2 = \frac{1}{N} \sum_{i=1}^{R} \sum_{j=1}^{C} z_{ij} \mathbb{E}_q \left[(y_{ij} - \boldsymbol{x}_{ij}^\mathsf{T} \beta - \boldsymbol{f}_i^\mathsf{T} \boldsymbol{l}_j - a_i - b_j)^2 \right] \tag{3.37}$$

Similar to the standard EM case, all updates involve only sums over the observed entries and inversions of small matrices, so each full iteration remains linear in the number of nonzero observations N. In practice, we use block coordinate updates (backfitting) to keep the computation tractable for large datasets.

Algorithm 2 Factored Variational EM algorithm for factor analysis model

Require: $\boldsymbol{y} = \{y_{ij}\}$, $\boldsymbol{X} = \{\boldsymbol{x}_{ij}\}$, and factor dimension K.
  Initialize $\boldsymbol{\beta} \sim N_p(0, I)$ and $\boldsymbol{l}_j \sim N_K(0, I)$ for each j.
  Initialize $\sigma_a^2, \sigma_b^2, \sigma_e^2 \sim \chi_1^2$.
  while parameters have not converged **do**
    Update the variational distributions by (3.27)–(3.32).
    Update the model parameters by (3.33)–(3.37).
  end while

3.7 Conclusion

In conclusion, we demonstrate the application of crossed random effects models and factor analysis in enhancing recommender systems. By integrating random intercepts and slopes, the proposed models effectively captured user-item interactions, even with sparse covariate information. Factor analysis

was instrumental in uncovering latent factors, allowing for accurate predictions in cases of incomplete data. The use of Expectation Maximization and variational EM algorithms provided scalable approaches to parameter estimation, proving advantageous in handling large datasets. Through empirical evaluation on the MovieLens dataset, the models presented here showed significant improvement in predictive accuracy, offering valuable insights for advancing personalized recommendation methodologies in real-world scenarios.

Appendix A

Some proofs

A.0.1 Proof of Theorem 1

Theorem 1. *Consider the solutions to the following penalized least-squares problem*

$$\min_{\boldsymbol{\beta},\boldsymbol{a}_i,\boldsymbol{b}_j} \sum_{i=1}^{R}\sum_{j=1}^{C} \frac{z_{ij}(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{\beta} + \boldsymbol{a}_i + \boldsymbol{b}_j))^2}{\sigma_e^2} + \sum_{i=1}^{R} \boldsymbol{a}_i^{\mathsf{T}}\Sigma_a^{-1}\boldsymbol{a}_i + \sum_{j=1}^{C} \boldsymbol{b}_j^{\mathsf{T}}\Sigma_b^{-1}\boldsymbol{b}_j. \tag{2.8}$$

Then $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{\text{GLS}}$ *and the* $\{\hat{\boldsymbol{a}}_i\}_{i=1}^{R}$ *and* $\{\hat{\boldsymbol{b}}_j\}_{j=1}^{C}$ *are the best linear unbiased prediction (BLUP) estimates of the random effects if the covariance parameters* Σ_b, Σ_b *and* σ_e^2 *are fixed and known.*

Here,

$$\hat{\boldsymbol{\beta}}_{\text{GLS}} = \arg\min_{\boldsymbol{\beta}} (Y - X\boldsymbol{\beta})\mathcal{V}^{-1}(Y - X\boldsymbol{\beta}) = (X^{\mathsf{T}}\mathcal{V}^{-1}X)^{-1}X^{\mathsf{T}}\mathcal{V}^{-1}Y \tag{2.7}$$

Proof. We rewrite the model as

$$Y = X\boldsymbol{\beta} + ZU + \varepsilon \quad \text{where, } U \sim \mathcal{N}(0, \Sigma_u), \text{ and } \epsilon \sim \mathcal{N}(0, \sigma_e^2 I)$$

Then,

$$P_r(Y = y, U = u) = P_r(Y = y)P_r(U = u|Y = y) = P_r(Y = y|U = u)P_r(U = u)$$

$P_r(Y = y)$ corresponds to the likelihood in (3.14) and $\hat{\boldsymbol{\beta}}_{\text{GLS}}$ in (2.7), and $P_r(Y = y|U = u)P_r(U = u)$ corresponds to the exponent of (2.8). Now,

$$\begin{aligned} &P_r(U = u|Y = y) \\ &\quad = \frac{1}{\sqrt{|Var(U|y)|}} \exp\left(-\frac{(u - \mathbb{E}[U|y])^{\mathsf{T}}\left[Var(U|y)\right]^{-1}(u - \mathbb{E}[U|y])}{2}\right) \end{aligned}$$

which is maximized for $u = \mathbb{E}[U|Y = y]$, and

$$\max_u P_r(U = u|Y = y) = \frac{1}{(2\pi)^{(p+1)/2}\sqrt{|Var(U|y)|}} = \text{a constant}$$

$$\max_u P_r(y, U = u) = \max_u P_r(Y = y)P_r(U = u|Y = y) = \text{ constant } \times p(y)$$

Therefore,

$$\arg\max_{\boldsymbol{\beta}} \max_u P_r(Y = y, U = u) = \arg\max_{\boldsymbol{\beta}} P_r(Y = y)\frac{1}{\sqrt{|Var(U|y)|}}$$

It is understood that $Var(U|y)$ is constant in terms of $\boldsymbol{\beta}$, which implies,

$$\arg\max_{\boldsymbol{\beta}} \max_u P_r(Y = y, U = u) = \arg\max_{\boldsymbol{\beta}} P_r(Y = y)$$

which is equivalent to saying that

$$\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{\text{GLS}}$$

□

A.0.2 Consistency of $\hat{\boldsymbol{\beta}}_{\text{OLS}}$

Theorem 3. *Let $X_r^{(a)}(X_c^{(b)})$ be the observations arranged in row (column) blocks such that each block has a fixed level for the first (second) factor. Also assume that $\max_{1\le r\le R} \frac{\lambda_{\max}(X_r^{(a)}\Sigma_a X_r^{(a)\mathsf{T}})}{N} \to 0$, $\max_{1\le c\le C} \frac{\lambda_{\max}(X_c^{(b)}\Sigma_b X_c^{(b)\mathsf{T}})}{N} \to 0$ and $\lambda_{min}(X^{\mathsf{T}}X)/N \ge c$ for some $c > 0$, as $N \to \infty$. Then $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is consistent.*

Proof. To begin with, we assume the observations are arranged by the level of the first factor. $Y = X\boldsymbol{\beta} + \varepsilon$ where $\varepsilon \sim N(0, \mathcal{V})$ for

$$\mathcal{V} = \underbrace{diag(X_1^{(a)}\Sigma_a X_1^{(a)\mathsf{T}}, \cdots, X_R^{(a)}\Sigma_a X_R^{(a)\mathsf{T}})}_{B_R} + P_C \underbrace{diag(X_1^{(b)}\Sigma_b X_1^{(b)\mathsf{T}}, \cdots, X_C^{(b)}\Sigma_b X_C^{(b)\mathsf{T}})}_{B_C} P_C^{\mathsf{T}} + \sigma_e^2 \mathbf{I_N}$$

with P_C being an appropriate permutation matrix. Note that $\hat{\boldsymbol{\beta}}_{\text{OLS}} = \boldsymbol{\beta} + (X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}\varepsilon$.

$\mathbb{E}((X^\mathsf{T}X)^{-1}X^\mathsf{T}\varepsilon) = \mathbf{0}$. Let w be any $p \times 1$ vector with finite norm.

$$\begin{aligned}
\mathrm{Var}(w^\mathsf{T}(X^\mathsf{T}X)^{-1}X^\mathsf{T}\varepsilon) &= w^\mathsf{T}(X^\mathsf{T}X)^{-1}X^\mathsf{T}\mathcal{V}X(X^\mathsf{T}X)^{-1}w \\
&\leq \lambda_{\max}(B_R)w^\mathsf{T}(X^\mathsf{T}X)^{-1}w \\
&\quad + \lambda_{\max}(P_C B_C P_C^\mathsf{T})w^\mathsf{T}(X^\mathsf{T}X)^{-1}w + \sigma_e^2 w^\mathsf{T}(X^\mathsf{T}X)^{-1}w \\
&\leq \lambda_{\max}(B_R)\frac{\|w\|^2}{cN} + \lambda_{\max}(P_C B_C P_C^\mathsf{T})\frac{\|w\|^2}{cN} + \sigma_e^2\frac{\|w\|^2}{cN} \\
&\leq \lambda_{\max}(B_R)\frac{\|w\|^2}{cN} + \lambda_{\max}(B_C)\frac{\|w\|^2}{cN} + \sigma_e^2\frac{\|w\|^2}{cN} \to 0
\end{aligned}$$

So $\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}$ is consistent estimator for $\boldsymbol{\beta}$. □

A.0.3 Method of Moments

Theorem 4. *The method of moment estimates obtained using the above method is consistent if the fourth moments for* $\mathbf{x}_{ij}, i \in \{1, \cdots, R\}$, $j \in \{1, \cdots, C\}$ *are uniformly bounded and* $\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}$ *is consistent.*

Proof. Each of the equations (2.23), (2.24), and (2.25) can be represented as

$$\mathbb{E}\Big[r^\mathsf{T}(I-Q)r\Big] = \hat{r}^\mathsf{T}(I-Q)\hat{r} \text{ for some } Q \in \mathbb{R}^{N\times N}. \tag{A.1}$$

For example, Q for (2.23) is given by

$$Q = \begin{bmatrix} \frac{x_{1.}^{(s)}x_{1.}^{(s)^\mathsf{T}}}{\|x_{1.}^{(s)}\|^2} & 0 & \cdots & 0 \\ 0 & \frac{x_{2.}^{(s)}x_{2.}^{(s)^\mathsf{T}}}{\|x_{2.}^{(s)}\|^2} & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \frac{x_{R.}^{(s)}x_{R.}^{(s)^\mathsf{T}}}{\|x_{R.}^{(s)}\|^2} \end{bmatrix} \tag{A.2}$$

It is sufficient to show that $(\mathbb{E}[r^\mathsf{T}(I-Q)r|X] - \hat{r}^\mathsf{T}(I-Q)\hat{r})/N$ converges to zero $\forall Q \in \{P_1, \cdots, P_p\}$,

$\{P'_1, \cdots, P'_p\}, J_n$ and $\forall\ \Sigma_a, \Sigma_b, \sigma_e^2$ where $J_n = 1_n 1_n^\mathsf{T}/n$.

$$\begin{aligned}\frac{\hat{r}^\mathsf{T}(I-Q)\hat{r}}{N} &= \frac{(\hat{r}-r+r)^\mathsf{T}(I-Q)(\hat{r}-r+r)}{N} \\ &= \frac{r^\mathsf{T}(I-Q)r + (\hat{r}-r)^\mathsf{T}(I-Q)(\hat{r}-r) + 2r^\mathsf{T}(I-Q)(\hat{r}-r)}{N} \\ &= \frac{r^\mathsf{T}(I-Q)r}{N} + \frac{(\hat{\boldsymbol{\beta}}_{\text{OLS}}-\boldsymbol{\beta})^\mathsf{T}X^\mathsf{T}(I-Q)X(\hat{\boldsymbol{\beta}}_{\text{OLS}}-\boldsymbol{\beta})}{N} \\ &\qquad - \frac{2r^\mathsf{T}(I-Q)X(\hat{\boldsymbol{\beta}}_{\text{OLS}}-\boldsymbol{\beta})}{N}\end{aligned}$$

As $\mathbf{x}_{ij}$ have finite fourth moment $\forall i \in \{1, \cdots, R\}\, j \in \{1, \cdots, C\}$, $X^\mathsf{T}X/N = O_p(1)$, and as $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ is consistent, second and third terms converge to zero in probability. Now,

$$\frac{(\hat{r}^\mathsf{T}(I-Q)\hat{r} - \mathbb{E}[r^\mathsf{T}(I-Q)r|X])}{N} = \frac{r^\mathsf{T}(I-Q)r}{N} - \frac{\mathbb{E}[r^\mathsf{T}(I-Q)r|X])}{N} + o_p(1)$$

Now,

$$\frac{r^\mathsf{T}(I-Q)r - \mathbb{E}[r^\mathsf{T}(I-Q)r]}{N} \to 0 \tag{A.3}$$

and

$$\frac{\mathbb{E}[r^\mathsf{T}(I-Q)r|X] - \mathbb{E}[r^\mathsf{T}(I-Q)r]}{N} \to 0 \tag{A.4}$$

by the strong law of large numbers assuming finite fourth moment of $\mathbf{x}_{ij}$ $\forall i \in \{1, \cdots, R\}$ and $j \in \{1, \cdots, C\}$. Let, $\boldsymbol{\theta} = (diag(\Sigma_a), diag(\Sigma_b), \sigma_e^2)$, then method of moment estimate, $\boldsymbol{\theta}_n$ is solution of equations in the form $A_n\boldsymbol{\theta} = B_n$, i.e. , $\boldsymbol{\theta}_n = A_n^{-1}B_n$. The above proof guarantees that $A_n \to A$ and $B_n \to A\boldsymbol{\theta}$, $\forall\boldsymbol{\theta}$. Therefore, $\boldsymbol{\theta}_n \to \boldsymbol{\theta}$. □

A.0.4 Proof of (2.26)

We know from (2.3) that

$$r_{ij} = \mathbf{x}_{ij}^\mathsf{T}(\boldsymbol{a}_i + \boldsymbol{b}_j) + \varepsilon_{ij}$$

It could also be represented as

$$r = \sum_{s'=0}^{p} X_A^{(s')} A_{s'} + \sum_{s'=0}^{p} X_B^{(s')} B_{s'} + \varepsilon.$$

Here, $A_{s'} \in \mathbb{R}^R$ and $B_{s'} \in \mathbb{R}^C$ represent the stacked random coefficients of the s'-th coordinate for clients and items, respectively, and $X_A^{(s')}$ and $X_B^{(s')}$ denote the corresponding design matrices. Then,

we have

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{i=1}^{R}\sum_{j=1}^{C} z_{ij}r_{ij}^2 - \sum_{i=1}^{R}\frac{\left(\sum_{j=1}^{C} z_{ij}r_{ij}x_{ij}^{(s)}\right)^2}{\sum_{j=1}^{C} z_{ij}x_{ij(s)}^2}\Big|X\Big] = \mathbb{E}[r^{\mathsf{T}}(I-Q)r] \\
&= \mathbb{E}\left(\sum_{s'=0}^{p} X_A^{(s')}A_{s'} + \sum_{s'=0}^{p} X_B^{(s')}B_{s'} + \varepsilon\right)^{\mathsf{T}}(I-Q) \\
&\qquad\qquad \cdot\left(\sum_{s'=0}^{p} X_A^{(s')}A_{s'} + \sum_{s'=0}^{p} X_B^{(s')}B_{s'} + \varepsilon\right) \\
&= \sum_{s'=0}^{p} E\Big[(X_A^{(s')}A_{s'})^{\mathsf{T}}(I-Q)(X_A^{(s')}A_{s'})\Big] \\
&\qquad + \sum_{s'=0}^{p} E\Big[(X_B^{(s')}B_s')^{\mathsf{T}}(I-Q)(X_B^{(s')}B_{s'})\Big] + \sigma_e^2(n-R) \\
&= \sum_{s'=0}^{p}\operatorname{tr}\left[X_A^{(s')^{\mathsf{T}}}(I-Q)X_A^{(s')}\right]\Sigma_{a,s's'} + \sum_{s'=0}^{p}\operatorname{tr}\left[X_B^{(s')^{\mathsf{T}}}(I-Q)X_B^{(s')}\right]\Sigma_{b,s's'} \\
&\qquad\qquad + \sigma_e^2(n-R) \\
&= \sum_{s'=0}^{p}\sum_{i=1}^{R}\left[\sum_{j=1}^{C} z_{ij}x_{ij}^{(s')^2} - \frac{\left(\sum_{j=1}^{C} z_{ij}x_{ij}^{(s)}x_{ij}^{(s')}\right)^2}{\sum_{j=1}^{C} z_{ij}x_{ij}^{(s)^2}}\right]\Sigma_{a,s's'} \\
&\qquad + \sum_{s'=0}^{p}\sum_{i=1}^{R}\left[\sum_{j=1}^{C} z_{ij}x_{ij}^{(s')^2} - \frac{\sum_{j=1}^{C} z_{ij}x_{ij}^{(s)^2}x_{ij}^{(s')^2}}{\sum_{j=1}^{C} z_{ij}x_{ij}^{(s)^2}}\right]\Sigma_{b,s's'} + \sigma_e^2(n-R)
\end{aligned}$$

The first part of the last equality follows directly from the definition of Q, the last part could be shown using the following argument:
Let,

$$X_B^{(s')} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_R \end{bmatrix}$$

$$\begin{aligned}
\operatorname{tr}\left[\left(X_B^{(s')}\right)^{\mathsf{T}}(I-Q)\left(X_B^{(s')}\right)\right] &= \sum_{i=1}^{R}\operatorname{tr}\left[W_i^{\mathsf{T}}\left(I - \frac{x_{i.}^{(s)}x_{i.}^{(s)^{\mathsf{T}}}}{\|x_{i.}^{(s)}\|^2}\right)W_i\right] \\
&= \sum_{i=1}^{R}\operatorname{tr}(W_i^{\mathsf{T}}W_i) - \frac{\operatorname{tr}\left(x_{i.}^{(s)^{\mathsf{T}}}W_iW_i^{\mathsf{T}}x_{i.}^{(s)}\right)}{\|x_{i.}^{(s)}\|^2}
\end{aligned}$$

Now,

$$W_i W_i^{\mathsf{T}} = \begin{bmatrix} {x_{i1}^{(s')}}^2 & 0 & \cdots & 0 \\ 0 & {x_{i2}^{(s')}}^2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & {x_{in_i}^{(s')}}^2 \end{bmatrix}$$

Thus,

$$\operatorname{tr}\left[\left(X_B^{(s')}\right)^{\mathsf{T}} (I - Q) \left(X_B^{(s')}\right) \right] = \sum_{i=1}^{R} \left[\sum_{j=1}^{C} z_{ij} {x_{ij}^{(s')}}^2 - \frac{\sum_{j=1}^{C} z_{ij} x_{ij}^{(s)^2} x_{ij}^{(s')^2}}{\sum_{j=1}^{C} z_{ij} x_{ij}^{(s)^2}} \right]$$

The additional equation we added was given by

$$\hat{r}^{\mathsf{T}} \left(I - \frac{1}{n} 11^{\mathsf{T}} \right) \hat{r} = E\left[r^{\mathsf{T}} \left(I - \frac{1}{n} 11^{\mathsf{T}} \right) r \right]$$

L.H.S. of the above equation will be given by

$$\sum_{i=1}^{R} \sum_{j=1}^{C} z_{ij} \hat{r}_{ij}^2 - \frac{1}{n} \left(\sum_{i=1}^{R} \sum_{j=1}^{C} z_{ij} \hat{r}_{ij} \right)^2$$

For R.H.S., similar to above, we will need the following expressions:

$$\operatorname{tr}\left[\left(X_A^{(s)}\right)^{\mathsf{T}} \left(I - \frac{1}{n} 11^{\mathsf{T}} \right) \left(X_A^{(s')}\right) \right] = \sum_{i=1}^{R} \left(\sum_{j=1}^{C} z_{ij} x_{ij}^{(s)} x_{ij}^{(s')} - \frac{\sum_{j=1}^{C} z_{ij} x_{ij}^{(s)} \sum_{j=1}^{C} z_{ij} x_{ij}^{(s')}}{n} \right)$$

and

$$\operatorname{tr}\left[\left(X_B^{(s)}\right)^{\mathsf{T}} \left(I - \frac{1}{n} 11^{\mathsf{T}} \right) \left(X_B^{(s')}\right) \right] = \sum_{j=1}^{C} \left(\sum_{i=1}^{R} z_{ij} x_{ij}^{(s)} x_{ij}^{(s')} - \frac{\sum_{i=1}^{R} z_{ij} x_{ij}^{(s)} \sum_{i=1}^{R} z_{ij} x_{ij}^{(s')}}{n} \right)$$

A.0.5 Simplication of covariance of $\hat{\boldsymbol{\beta}}$

$$
\begin{aligned}
&\text{Cov}\hat{\boldsymbol{\beta}} \\
&= (X^\intercal \tilde{X})^{-1} \tilde{X}^\intercal \left(X\hat{\Sigma}_a X^\intercal \bullet Z_a Z_a^\intercal + X\hat{\Sigma}_b X^\intercal \bullet Z_b Z_b^\intercal + \hat{\sigma}_e^2 I_N \right) \tilde{X} (\tilde{X}^\intercal X)^{-1} \\
&= (X^\intercal \tilde{X})^{-1} \left[\tilde{X}^\intercal \left(X\hat{\Sigma}_a X^\intercal \bullet Z_a Z_a^\intercal \right) \tilde{X} + \tilde{X}^\intercal \left(X\hat{\Sigma}_b X^\intercal \bullet Z_b Z_b^\intercal \right) \tilde{X} \right] (X^\intercal X)^{-1} \\
&\qquad\qquad + \hat{\sigma}_e^2 (\tilde{X}^\intercal X)^{-1}
\end{aligned}
$$

Let $P_a \in \mathbb{R}^{N\times N}$ denotes the permutation matrix that rearranges the data points so that they are grouped by clients and $P_b \in \mathbb{R}^{N\times N}$ denotes the permutation matrix that rearranges the data points so that they are grouped by items.

$$
\begin{aligned}
\tilde{X}^\intercal \left(X\hat{\Sigma}_a X^\intercal \bullet Z_a Z_a^\intercal \right) \tilde{X} &= \tilde{X}^\intercal P_a^\intercal P_a \left(X\hat{\Sigma}_a X^\intercal \bullet Z_a Z_a^\intercal \right) P_a^\intercal P_a \tilde{X} && \text{(A.5)} \\
&= \tilde{X}^\intercal P_a^\intercal \left(P_a X\hat{\Sigma}_a X^\intercal P_a^\intercal \bullet P_a Z_a Z_a^\intercal P_a^\intercal \right) P_a \tilde{X} && \text{(A.6)}
\end{aligned}
$$

In order to keep the notation simplified, we shall move ahead assuming that data is grouped by clients, i.e., we shall be denoting $P_a X$ by X, $P_a \tilde{X}$ by $\tilde{X}$ and $P_a Z_a$ by Z_a. The assumption simplifies the structure of $Z_a Z_a^\intercal$ to the block diagonal matrix given by

$$
Z_a Z_a^\intercal = \begin{bmatrix}
1_{n_1} 1_{n_1}^\intercal & 0 & \cdots & 0 \\
0 & 1_{n_2} 1_{n_2}^\intercal & \cdots & 0 \\
0 & 0 & \ddots & 0 \\
0 & 0 & 0 & 1_{n_R} 1_{n_R}^\intercal
\end{bmatrix}
$$

Now, let's define $U = X\hat{\Sigma_a}^{1/2}$ and represent column i of $\tilde{X}$ by s and column j of $\tilde{X}$ by t. Therefore, $X\hat{\Sigma}_a X^\intercal = \sum_{q=0}^{p} U^{(q)} U^{(q)^\intercal}$ where $\{U^{(q)}\}$ represents columns of U.

Therefore, (i, j) entry in equation A.5 is given by

$$\sum_{q=0}^{p}(s_{1.}^{\mathsf{T}},\cdots,s_{R.}^{\mathsf{T}})\begin{bmatrix}{U^{(q)}}_{1.}{U^{(q)}}_{1.}^{\mathsf{T}} & \cdots & 0\\ 0 & \ddots & 0\\ 0 & \cdots & {U^{(q)}}_{R.}{U^{(q)}}_{R.}^{\mathsf{T}}\end{bmatrix}\begin{pmatrix}t_1\\ \vdots\\ t_R\end{pmatrix}$$
$$=\sum_{q=0}^{p}\sum_{i=1}^{R}s_{i.}^{\mathsf{T}}{U^{(q)}}_{i.}{U^{(q)}}_{i.}^{\mathsf{T}}t_{i.}$$
$$=\sum_{q=0}^{p}\sum_{i=1}^{R}s_{i.}^{\mathsf{T}}{U^{(q)}}_{i.}{U^{(q)}}_{i.}^{\mathsf{T}}t_{i.}$$
$$=\sum_{q=0}^{p}\sum_{i=1}^{R}\sum_{j=1}^{C}z_{ij}s_{ij}^{\mathsf{T}}{U^{(q)}}_{ij}{U^{(q)}}_{ij}^{\mathsf{T}}t_{ij}$$

Therefore, the quantity $\left[\tilde{X}^{\mathsf{T}}\left(X\hat{\Sigma}_aX^{\mathsf{T}}\bullet Z_aZ_a^{\mathsf{T}}\right)\tilde{X}+\tilde{X}^{\mathsf{T}}\left(X\hat{\Sigma}_bX^{\mathsf{T}}\bullet Z_bZ_b^{\mathsf{T}}\right)\tilde{X}\right]\in\mathbb{R}^{(p+1)\times(p+1)}$ can be computed in $\mathcal{O}(Np^3)$ time which makes the computation of $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ possible in time linear in N.

A.0.6 Proof of Lemma 1

$$q_a^k(a_1,\cdots,a_R)\propto\mathbb{P}(y,\{\boldsymbol{a}_i\}_{i=1}^R|\boldsymbol{b}_j=E_{Q^{(k-1)}}[\boldsymbol{b}_j])$$
$$\propto\exp\left\{-\frac{1}{2}\sum_{i=1}^{R}\sum_{j=1}^{C}\frac{z_{ij}(y_{ij}-\mathbf{x}_{ij}^{\mathsf{T}}(\boldsymbol{a}_i+\mu_{b,j}^{(k-1)}+\boldsymbol{\beta}))^2}{\left(\sigma_e^{(k-1)}\right)^2}-\frac{1}{2}\sum_{i=1}^{R}\boldsymbol{a}_i^{\mathsf{T}}(\Sigma_a^{(k-1)})^{-1}\boldsymbol{a}_i\right\}$$
$$\propto\exp\left\{-\frac{1}{2}\sum_{i=1}^{R}\boldsymbol{a}_i^{\mathsf{T}}\left((\Sigma_a^{(k-1)})^{-1}+\frac{\sum_{j=1}^{C}z_{ij}\mathbf{x}_{ij}\mathbf{x}_{ij}^{\mathsf{T}}}{\left(\sigma_e^{(k-1)}\right)^2}\right)\boldsymbol{a}_i\right.$$
$$\left.+\boldsymbol{a}_i^{\mathsf{T}}\frac{\sum_{j=1}^{C}z_{ij}\left(y_{ij}-\mathbf{x}_{ij}^{\mathsf{T}}\left(\boldsymbol{\beta}^{(k-1)}+\mu_{b,j}^{(k-1)}\right)\right)\mathbf{x}_{ij}}{\left(\sigma_e^{(k-1)}\right)^2}\right\}$$

If $q_a^k(a_1,\cdots,a_R)$ represents the distribution $\mathcal{N}(\mu_{a,i}^{(k)},\Sigma_{a,i}^{(k)})$, then

$$q_a^k(a_1,\cdots,a_R)\propto\exp\left\{-\frac{1}{2}\boldsymbol{a}_i^{\mathsf{T}}\left(\Sigma_{a,i}^{(k)}\right)^{-1}\boldsymbol{a}_i+\boldsymbol{a}_i^{\mathsf{T}}\left(\Sigma_{a,i}^{(k)}\right)^{-1}\mu_{a,i}^{(k)}\right\}\tag{A.7}$$

Thus,

$$\mu_{a,i}^{(k)} = \Sigma_{a,i}^{(k)} \left(\frac{\sum_{j=1}^{C} z_{ij} \big(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}} \big(\boldsymbol{\beta}^{(k-1)} + \mu_{b,j}^{(k-1)} \big) \big) \mathbf{x}_{ij}}{\left(\sigma_e^{(k-1)} \right)^2} \right)$$

and

$$\Sigma_{a,i}^{(k)} = \left((\Sigma_a^{(k-1)})^{-1} + \frac{\sum_{j=1}^{C} z_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^{\mathsf{T}}}{\left(\sigma_e^{(k-1)} \right)^2} \right)^{-1}.$$

Similarly,

$$\begin{aligned}
q_b^k(b_1, \cdots, b_C) &\propto \mathbb{P}(y, \{\boldsymbol{b}_j\}_{j=1}^{C} | \boldsymbol{a}_i = E_{Q^{(k)}}[\boldsymbol{a}_i]) \\
&\propto \exp \left\{ -\frac{1}{2} \sum_{j=1}^{C} \sum_{i=1}^{R} \frac{z_{ij} (y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}} (\boldsymbol{b}_j + \mu_{a,i}^{(k)} + \boldsymbol{\beta}))^2}{\left(\sigma_e^{(k-1)} \right)^2} - \frac{1}{2} \sum_{j=1}^{C} \boldsymbol{b}_j^{\mathsf{T}} (\Sigma_b^{(k-1)})^{-1} \boldsymbol{b}_j \right\} \\
&\propto \exp \left\{ -\frac{1}{2} \sum_{j=1}^{C} \boldsymbol{b}_j^{\mathsf{T}} \left((\Sigma_b^{(k-1)})^{-1} + \frac{\sum_{i=1}^{R} z_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^{\mathsf{T}}}{\left(\sigma_e^{(k-1)} \right)^2} \right) \boldsymbol{b}_j \right. \\
&\qquad \left. + \boldsymbol{b}_j^{\mathsf{T}} \frac{\sum_{j=1}^{C} z_{ij} \big(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}} \big(\boldsymbol{\beta}^{(k-1)} + \mu_{a,i}^{(k)} \big) \big) \mathbf{x}_{ij}}{\left(\sigma_e^{(k-1)} \right)^2} \right\}
\end{aligned}$$

For the same reasoning,

$$\mu_{b,j}^{(k)} = \Sigma_{b,j}^{(k)} \left(\frac{\sum_{i=1}^{R} z_{ij} \big(y_{ij} - \mathbf{x}_{ij}^{\mathsf{T}} \big(\boldsymbol{\beta}^{(k-1)} + \mu_{a,i}^{(k-1)} \big) \big) \mathbf{x}_{ij}}{\left(\sigma_e^{(k-1)} \right)^2} \right)$$

and

$$\Sigma_{b,j}^{(k)} = \left((\Sigma_b^{(k-1)})^{-1} + \frac{\sum_{i=1}^{R} z_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^{\mathsf{T}}}{\left(\sigma_e^{(k-1)} \right)^2} \right)^{-1}$$

Appendix B

Results under general setup

For the general setup, we consider the covariates to be the union of item-based features and client-based features, i.e., $x_{ij} = (1, v_j, u_i)$. We include random effects for clients on item-based features and vice versa. Therefore, the model is given by

$$y_{ij} = x_{ij}^\top \beta + v_j^\top a_i + u_i^\top b_j + \varepsilon_{ij},$$

where $a_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}_3(0, \Sigma_a)$, $b_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}_3(0, \Sigma_b)$, and $\varepsilon_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_e^2)$, with

$$\Sigma_a = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}, \quad \Sigma_b = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}, \quad \text{and} \quad \sigma_e^2 = 1.$$

We chose $\beta = (0.1, 0.2, 0.3, 0.4, 0.5)$, and generated $(v_j, u_i) \overset{\text{i.i.d.}}{\sim} \mathcal{N}_4(0, I)$.

In Figures B.1 and B.2, we present the consistency of our algorithm in estimating the model parameters under the general setup. The results are similar to those presented earlier, i.e., the accuracy of our proposed algorithm approaches that of the maximum likelihood estimates as N increases. In Figure B.3, we compare the estimation time of our proposed algorithm with existing approach in the general setup; once again, the conclusions are consistent with those reported previously.

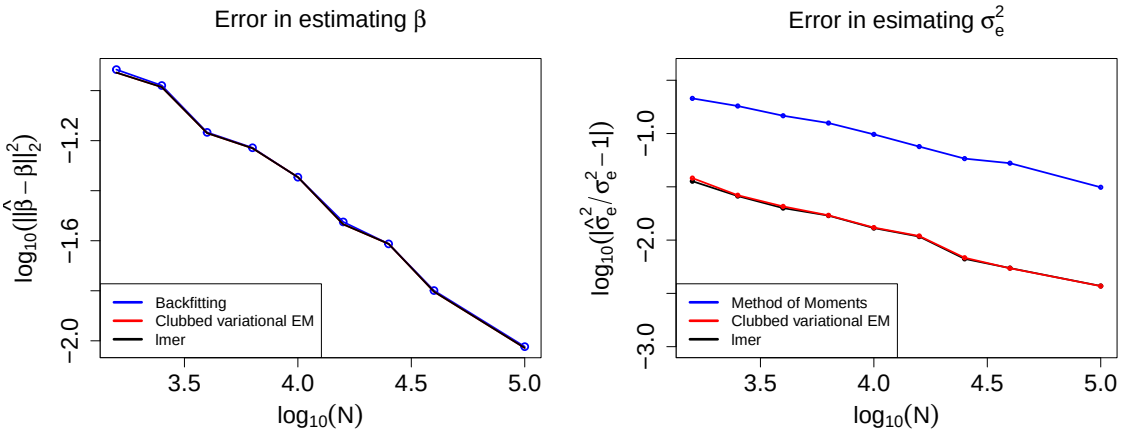


Figure B.1: General setup: Comparison of accuracy in estimating $\boldsymbol{\beta}$ and σ_e^2 by variational EM (proposed algorithm) and lmer(existing algorithm). The errors in estimating $\boldsymbol{\beta}$ are close to each other for the illustrated algorithms, which leads the lines in the plot to coincide.

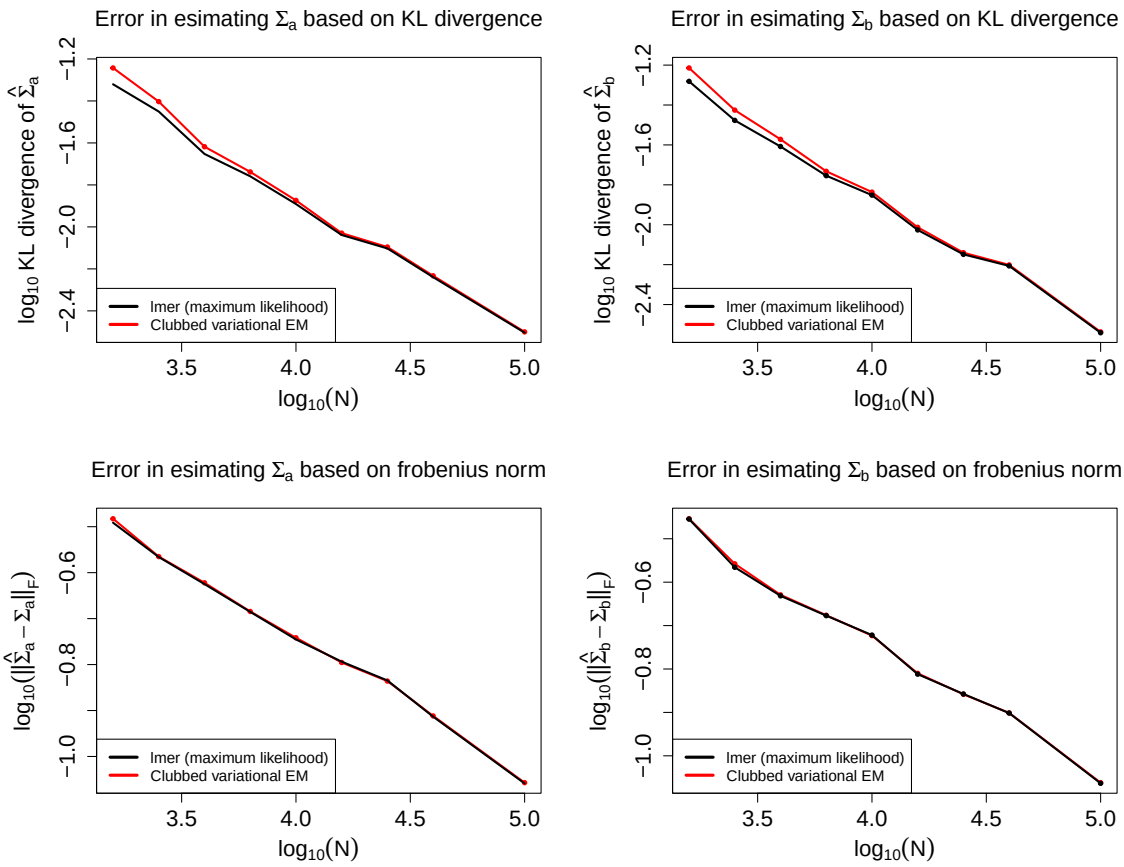


Figure B.2: General setup: Comparison of accuracy in estimating Σ_a and Σ_b based on KL divergence and Frobenius norm by variational EM (proposed algorithm) and lmer(existing algorithm).

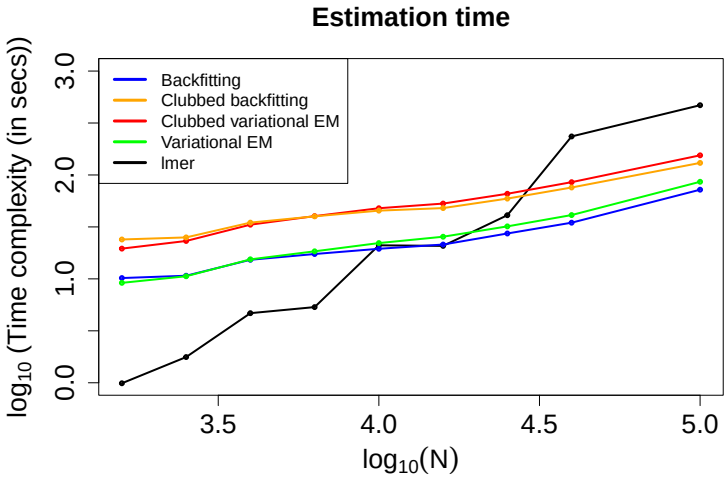


Figure B.3: General setup: Comparison of estimation time by backfitting, clubbed backfitting, variational EM, and clubbed variational EM (four proposed algorithms) and "lmer"(existing algorithm).

Bibliography

R.H. Baayen, D.J. Davidson, and D.M. Bates. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412, 2008. ISSN 0749-596X. doi: https://doi.org/10.1016/j.jml.2007.12.005. URL https://www.sciencedirect.com/science/article/pii/S0749596X07001398. Special Issue: Emerging Data Analysis.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. doi: 10.18637/jss.v067.i01.

Matthew Beal and Zoubin Ghahramani. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian Statistics*, 7(2):000–000, 2003.

A. Buja, T. Hastie, and R. Tibshirani. Linear smoothers and additive models (with discussion). *The Annals of Statistics*, 17(2):453–510, 1989.

Emmanuel J. Candes and Benjamin Recht. Exact matrix completion via convex optimization, 2008. URL https://arxiv.org/abs/0805.4471.

K. Gao and A. B. Owen. Efficient moment calculations for variance components in large unbalanced crossed random effects designs. *Electronic Journal of Statistics*, 11(1):1235–1296, 2017.

K. Gao and A. B. Owen. Estimation and inference for very large linear mixed effects models. *Statistica Sinica*, 30:1741–1771, 2020.

Zoubin Ghahramani and Geoffrey E. Hinton. The em algorithm for mixtures of factor analyzers. 1996. URL https://api.semanticscholar.org/CorpusID:18270595.

Disha Ghandwani, Swarnadip Ghosh, Trevor Hastie, and Art B. Owen. Scalable solution to crossed random effects model with random slopes, 2023.

S. Ghosh and C. Zhong. Convergence rate of a collapsed gibbs sampler for crossed random effects models. Technical report, arxiv:2109.02849, 2021.

Swarnadip Ghosh. *Scalable inference for crossed random effects models.* PhD thesis, Stanford University, 2022.

Swarnadip Ghosh, Trevor Hastie, and Art B Owen. Backfitting for large scale crossed random effects regressions. *Annals of Statistics*, 50(1):560–583, 2022a.

Swarnadip Ghosh, Trevor Hastie, and Art B Owen. Scalable logistic regression with crossed random effects. *Electronic Journal of Statistics*, 16(2):4604–4635, 2022b.

Max Goplerud. Fast and accurate estimation of non-nested binomial hierarchical models using variational inference. *Bayesian Analysis*, 17:623–650, 2022. doi: 10.1214/21-BA1266.

Richard L Gorsuch. *Factor analysis: Classic edition*. Routledge, 2014.

Trevor Hastie, Rahul Mazumder, Jason Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *Journal of Machine Learning Research*, 16(104):3367–3402, 2015. URL http://jmlr.org/papers/v16/hastie15a.html.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(80):2287–2322, 2010. URL http://jmlr.org/papers/v11/mazumder10a.html.

Marianne Menictas, Gioia Di Credico, and Matt P. Wand. Streamlined variational inference for linear mixed models with crossed random effects, 2022.

Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf.

O. Papaspiliopoulos, G. O. Roberts, and G. Zanella. Scalable inference for crossed random effects models. *Biometrika*, 107(1):25–40, 2020.

G.K. Robinson. That BLUP is a good thing: the estimation of random effects. *Statistical Science*, 6(1):15–51, 1991.

Ping Wu, Winfried Stute, and Li-Xing Zhu. Efficient estimation of moments in linear mixed models. *Bernoulli*, 18(1):206–228, 2012.