Easy and Efficient Bayesian Infinite Factor Analysis

by

Evan Poworoznek

Department of Statistical Science
Duke University

Date: ______________________________

Approved:

____

David Dunson, Supervisor

____

Amy Herring

____

Colin Rundel

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Department of Statistical Science
in the Graduate School of
Duke University

2020

ABSTRACT

Easy and Efficient Bayesian Infinite Factor Analysis

by

Evan Poworoznek

Department of Statistical Science
Duke University

Date: ______________________________
Approved:

____

David Dunson, Supervisor

____
Amy Herring

____
Colin Rundel

An abstract of a thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Department of Statistical Science
in the Graduate School of
Duke University

2020

Abstract

Bayesian latent factor models are key tools for modeling linear structure in data and performing dimension reduction for correlated variables. Recent advances in prior specification allow the estimation of semi- and non-parametric infinite factor models. These models provide significant theoretical and practical advantages at the cost of computationally intensive sampling and non-identifiability of some parameters. We provide a package for the `R` programming environment that includes functions for sampling from the posterior distributions of several recent latent factor models. These computationally efficient samplers are provided for `R` with C++ source code to facilitate fast sampling of standard models and provide component sampling functions for more complex models. We also present an efficient algorithm to remove the non-identifiability that results from the included shrinkage priors. The `infinitefactor` package is available in developmental version on GitHub at https://github.com/poworoznek/infinitefactor and in release version on the CRAN package repository.

Acknowledgements

I am very grateful to my advisor, Professor David Dunson, for his guidance and support throughout my time here at Duke University, and to the other members of my committee, Professor Amy Herring and Professor Colin Rundel, who have both helped me in class and in research. I would like to thank Federico Ferrari, Dr. Kelly Moran, and Isaac Lavine for the very productive meetings. I would also like to thank all of my friends here at Duke for making this such a great environment. Finally I would like to thank my mother and father, who have given me every opportunity I could hope for.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Factor models are commonly used to characterize the dependence structure of correlated variables with a linear decomposition of the $p \times p$ covariance matrix Ω as $\Lambda\Lambda^T + \Sigma$. Here Λ is the $p \times k$ matrix of factor loadings and Σ is a diagonal $p \times p$ matrix. The columns of the matrix Λ can be interpreted as the basis vectors in a latent space. The latent dimension k, typically less than the number of covariates p, is chosen to balance the parameter efficiency of the factorization and the amount of information retained in the factorization. The above factorization can equivalently be specified with a latent representation of the covariates included in the $n \times p$ data matrix X:

$$X = \eta\Lambda^T + \epsilon, \quad \epsilon_i \sim \mathcal{N}_p(\mathbf{0}, \Sigma) \tag{1.1}$$

where $\epsilon = [\epsilon_1, \ldots, \epsilon_n]$ and η is a $n \times k$ matrix of latent factors. Data with stronger linear dependence structure lends itself to more efficient factorization and improved performance in prediction. The decomposition of the covariance Ω is not unique, and as a result Λ and η are not individually identifiable.

Factor models are applicable to a wide class of applied problems that involve dimension reduction or highly correlated data, for example in the fields of epidemiology and genomics. The covariance factorization is applicable to cases with $n << p$, and allows efficient inference and prediction in this case if k can be chosen to be small. Frequentist exploratory and confirmatory factor analysis and similar methods such as principal components analysis are well established. In comparison Bayesian methods provide uncertainty quantification and, depending on the choice of prior, can help choose number of latent factors. The choice of prior in Bayesian factor analysis is key, and recent advances in semi-parametric factor models have created multiple widely used priors including the Multiplicative Gamma shrinkage prior [BD11] and the Dirichlet-Laplace shrinkage prior [BPPD15]. These priors take different approaches to parameter shrinkage for stable estimates and are described in *Section 2.2.1*.

In this thesis we discuss an R package for Gaussian factor modeling and a new algorithm for resolving non-identifiability. The goal of the `infinitefactor` package is to make these factor models available to researchers in applied fields and provide component functions for further development of statistical methods. We provide modular samplers that make the prior-specific sampling machinery exchangeable between data models. The computational bottleneck is reduced through the use of parallelization and C++ acceleration with the `Rcpp` package. We also present the efficient MSF post-processing algorithm that resolves non-identifiability in the Λ and η parameters and allows inference on these factor loadings and latent factors, and provide an implementation of this algorithm for use with the factor model samplers.

Chapter 2

infinitefactor Package

2.1 Bayesian Sampling

In Bayesian inference, the posterior distribution of parameters combines information from the prior and likelihood. Our focus is on the prior specification for Λ that allows efficient and stable covariance factorization. As with many Bayesian applications the resulting posterior distributions are not known in closed form, and we approximate the posterior distributions by sampling. The backbone of our package is the set of samplers that produce Markov Chain Monte-Carlo (MCMC) samples of the posterior distributions. Users may specify parameters that govern the sampling, such as the prior, the total number of samples, the burn-in period length, and in some cases specification of particular hyperparameters to related shrinkage in the matrix Λ. An example of the output of an MCMC chain is shown in *Figure 2.1*.

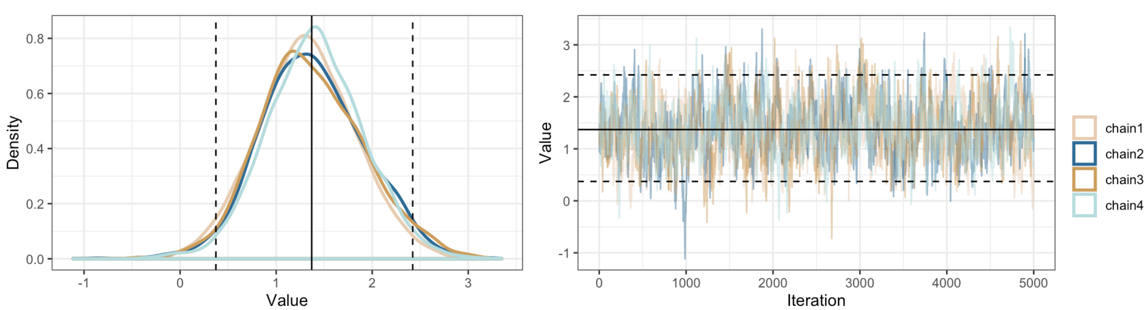

Figure 2.1: Sampling density and MCMC trace plot for 4 sampling chains of one covariance matrix entry in a linear factor model with a Dirichlet-Laplace prior. Estimated posterior mean value shown with a solid black line, and 95% posterior credible interval shown with a dashed black line.

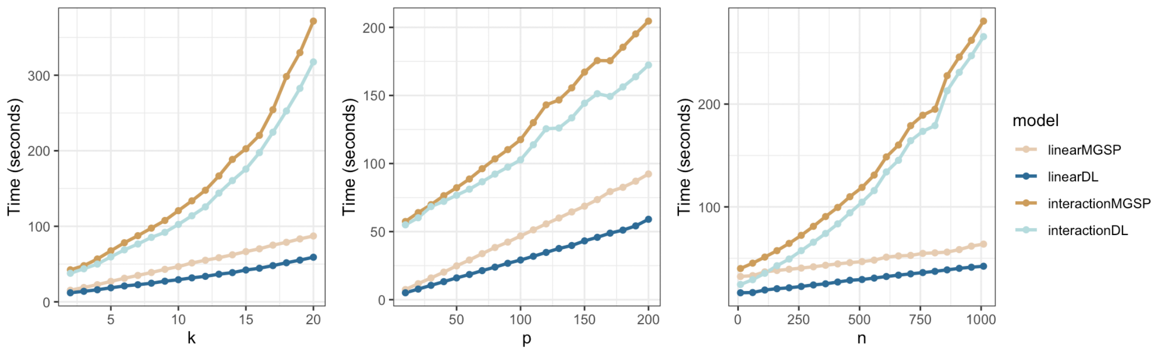

Figure 2.2: Computation time for 10000 samples (including 5000 burn in) across typical parameter ranges for the included sampling functions. The baseline parameter values are $k = 10$, $p = 100$, and $n = 500$.

Inference for parameters in the Bayesian setting is performed by ergodic averaging:

$$h(\theta) \approx \frac{1}{M} \sum_{i=1}^{M} h(\theta^{(i)})$$

for a parameter θ, a function h of interest, and MCMC posterior samples $\{\theta^{(i)}\}$ for $i = 1, ..., M$. In this way we can compute posterior expectations, quantiles, and other summary statistics using posterior samples. Consistency, posterior concentration, and related properties are discussed with the introduction of each prior [BD11] [BPPD15]. Multiple tests for convergence of MCMC samples have been developed and can be applied to output from this package, including the test of Gelman and Rubin [GR$^+$92] implemented in the `coda` package. Note that such tests should only be applied to identifiable samples. A discussion of identifiability is presented in *Section 2.3*.

2.2 Functionality

The `infinitefactor` package is intended to provide both tools for inference and building-block functions for more complex models. The four main sampling functions as of package version 1.0 are `linearMGSP()`, `linearDL()`, `interactionMGSP()`, `interactionDL()`. These functions are high level wrappers for component sampling functions that make inference accessible. These functions take in data matrices with minimal pre-processing and output posterior samples necessary for inference. Internally, these functions are wrappers for a set of independent sampling functions. Due to the formulation of the priors included in this package, and many other potential priors, these lower level sampling functions may be reused in new models.

The included priors apply shrinkage to the columns or rows of the factor loadings matrix Λ, while other parameters, such as η, are updated conditionally on Λ without the direct influence of the shrinkage prior. Due to this parameter separability we can apply the same sampling function for η across different prior specifications and models. Similarly, the linear factor model and interaction factor model differ in the model specification of the regression parameters, which affects the posterior distribution of η. Because of the same parameter separability the full conditional distribution of Λ is unaffected by this change in model. Many other likelihoods and priors can be specified with this structure to make use of some of these parameter updates.

The computational scaling of the linear factor model and FIN is shown in *Figure 2.2* across ranges of parameter values that are typically seen in factor model applications. Computational efficiencies in the package implementation result in approximately linear scaling in all parameters for the linear factor models, and linear scaling in p for the interaction model. These sampling functions are viable even for extremely large scale applications, such as data with a number of covariates on the order of $100,000$.

2.2.1 Priors

The release version of this package contains two priors applicable to different situations: the Multiplicative Gamma Shrinkage prior (MGSP) of [BD11] and the Dirichlet-Laplace (DL) prior of [BPPD15] which is been applied to factor models in [FD19].

The MGSP prior allows Bayesian covariance estimation with unknown latent dimension k, which can be an important modeling decision. The MGSP prior applies stronger shrinkage to the factor loadings matrix Λ as the column index increases. Let $\Lambda = [\lambda_{jh}]$, for $j = 1, \ldots, p$ and $h = 1, \ldots, k$,

$$\begin{gathered}\lambda_{jh} \mid \phi_{jh}, \tau_h \sim \mathcal{N}(0, \phi_{jh}^{-1}\tau_h^{-1}), \quad \phi_{jh} \sim \mathrm{Ga}\left(\frac{\nu}{2}, \frac{\nu}{2}\right) \quad \tau_h = \prod_{l=1}^{h} \theta_l \\ \theta_1 \sim \mathrm{Ga}(a_1, 1), \quad \theta_{l\geq 2} \sim \mathrm{Ga}(a_2, 1)\end{gathered} \tag{2.1}$$

where ν, a_1, and a_2 are hyperparameters that affect shrinkage. See [Dur17] for a discussion on the choice of hyperparameters. In particular, when $a_2 > 1$ the shrinkage on higher indexed columns will stochastically increase. With stocastically increasing shrinkage, columns are more and more concentrated in probability density around the zero vector as the column index increases. Hence, at each iteration of the MCMC procedure only the first k columns are sampled, where k can vary across iterations if an adaptive sampler is used. This adaptive sampling has been shown to center around the true number of factors in some simulation experiments.

We also implement the DL prior for each row of the factor loadings matrix Λ. The use of this prior requires the choice of the latent dimension k, and an adaptive

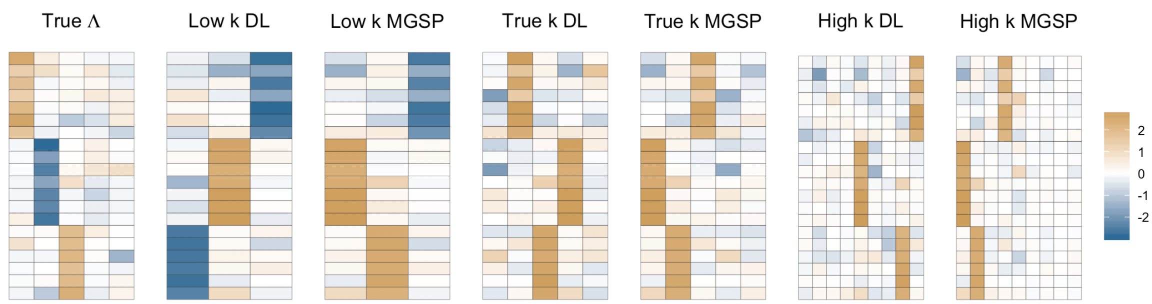

Figure 2.3: Comparison of prior behavior on the mean estimation of factor loadings for cases with misspecified k. Parameter mean estimates are shown after post-processing. Note that column sign and label switches between matrices is expected because these are summaries of different samples, and inferences within a sample are not affected.

sampler is not available. The DL prior is defined as

$$\begin{aligned} \lambda_{j,h}|\phi_{jh},\tau_j &\sim DE(\phi_{jh}\tau_j) \quad h=1,\cdots,k \\ \phi_j \sim Dir(a,\cdots,a) &\quad \tau_j \sim Gamma(ka,1/2), \end{aligned} \tag{2.2}$$

where $j = 1,\cdots,p$, $\phi_j = (\phi_{j1},\cdots,\phi_{jk})$, and DE refers to the zero mean double-exponential or Laplace distribution. Here we often choose k as an upper bound on the number of factors, as the prior allows effective deletion of redundant factor loadings through row-wise shrinkage. The DL prior provides flexible shrinkage on the factor loadings matrix, generalizing the Bayesian Lasso [PC08] to have a carefully chosen hierarchical structure on covariate-specific (τ_j) and local (ϕ_{jh}) scales. This induces a prior with concentration at zero, to strongly shrink small signals, and heavy-tails, to avoid over-shrinking large signals. The DL prior induces near sparsity row-wise in the matrix Λ, as it is reasonable to assume that each variable loads on few factors. By default, we set the hyperparameter a to the value $1/2$, see [BPPD15] for a discussion on the hyperparameter choice.

The behavior of this prior in relation to the multiplicative gamma prior under misspecification of the tuning parameter k is illustrated in *Figure 2.3* and again for larger k in *Section 2.5*. From the computational scaling results in *Figure 2.2* we can see that the Dirichlet-Laplace prior tends to result in faster sampling times in this implementation, although the parameters in the Dirichlet-Laplace sampler require draws from a generalized inverse Gaussian distribution, which may be slow to sample in certain cases due to the rejection sampling method.

2.2.2 Models

The linear latent factor model is described in the introduction, and is commonly used for linear dimension reduction and Bayesian covariance estimation. We assume an isotropic normal prior for the factors, i.e. $\eta_i \sim \mathcal{N}_k(\mathbf{0}, I_k)$, which is a common choice in the Bayesian literature and we choose independent inverse-gamma priors for the element of the diagonal of Σ. We scale the data matrix X before sampling to remove the intercept and place the covariates on the same scale, so that the factor loadings can be comparable across rows of Λ. We apply inverse scaling on the covariance samples before the function output to maintain consistency with X. Under the linear factor model the covariance of X, which we denote as Ω, is decomposed as $\Lambda\Lambda^T + \Sigma$, and regression coefficients for $X_j \mid X_{-j}$ can be efficiently computed using the Sherman-Morrison-Woodbury identity. This model makes the important assumption that the rows of X are jointly normally distributed, and violations of this assumption can lead to inaccurate inference.

An extension of (1.1) for interaction modeling was introduced in [FD19] which provides an efficient representation of a quadratic regression on the outcome. This model performs the regression of a response y onto the latent η and its quadratic form, i.e.

$$y_i = \eta_i^T \omega + \eta_i^T \Psi \eta_i + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_y^2) \tag{2.3}$$

$$X = \eta \Lambda^T + \epsilon, \quad \epsilon_i \sim \mathcal{N}_p(\mathbf{0}, \Sigma) \tag{2.4}$$

where ω is a main effect coefficient vector and Ψ is a matrix of interaction effect coefficients and all the other parameters are defined as in (1.1). In our implementation we assume independent standard normal priors for the elements of these coefficient parameters. This model specification is shown to induce a quadratic regression of y on X with the added efficiency of factor modeling in cases where a covariate set is highly correlated and contains strong linear structure. By learning this covariate structure and using it for coefficient estimation, this method does not violate the hierarchy principle in this estimation of 2-way interactions.

2.3 Identifiability

The semi-parametric shrinkage priors included in this package contain non-identifiable parameters. Bhattacharya and Dunson motivate the MGSP prior with the intuition that inference on the covariance matrix Ω does not require the matrices Λ and η to be individually identifiable. In this way they avoid introducing constraints that may induce an order dependence on the columns of X, or require a priori grouping of the covariates as noted in [CCL$^+$08]. As a consequence, inference on unprocessed samples of Λ, η, or latent-space coefficients in the factor interaction model is not possible. We see the form of this non-identifiability in the covariance decomposition and latent

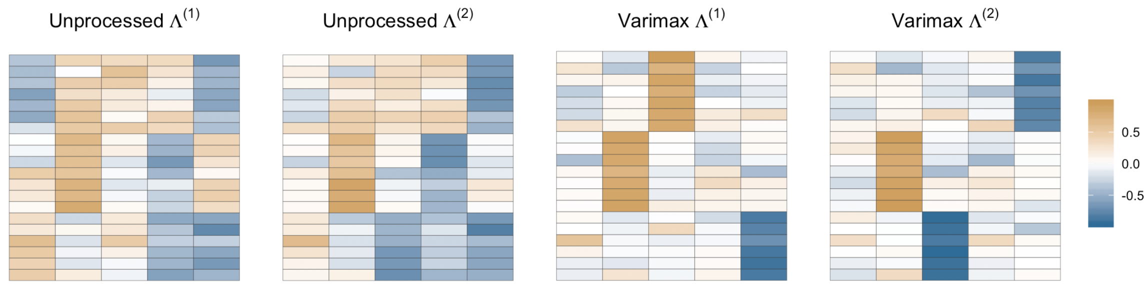

Figure 2.4: Two samples of the parameter Λ from the same chain in a simulated model before (left two) and after (right two) Varimax rotation, showing full rotational ambiguity and rotational ambiguity reduced to label and sign switching in the columns.

representation of X:

$$\Omega = \Lambda\Lambda^T + \Sigma = \Lambda P P^T \Lambda^T + \Sigma = (\Lambda P)(\Lambda P)^T + \Sigma$$

$$X = \eta\Lambda^T + \epsilon = \eta P P^T \Lambda^T + \epsilon = \eta P(\Lambda P)^T + \epsilon, \quad P \in \{P : PP^T = I_k\}$$

For any $k \times k$ rotation matrix P there is an equivalence class of factor loadings and factor pairs (Λ, η) that are given identical probability by the posterior distribution. For some member of the equivalence class $(\Lambda^{(i)}, \eta^{(i)})$ on which we wish to perform inference, the unprocessed MCMC samples are $\{\Lambda^{(i)}P^{(i)}, \eta^{(i)}P^{(i)}\}_{i=1}^{M}$. The effect of rotational ambiguity during sampling is shown in *Figure 2.4*. Determining which member of this equivalence class to choose is related to the interpretations of the loadings, which we discuss with the orthogonalization methods. This representation shows the fundamental problem with inference on unprocessed samples that contain this non-identifiability.

$$\frac{1}{M}\sum_{i=1}^{M} h(\Lambda^{(i)}P^{(i)}) \neq \frac{1}{M}\sum_{i=1}^{M} h(\Lambda^{(i)}) \approx h(\Lambda)$$

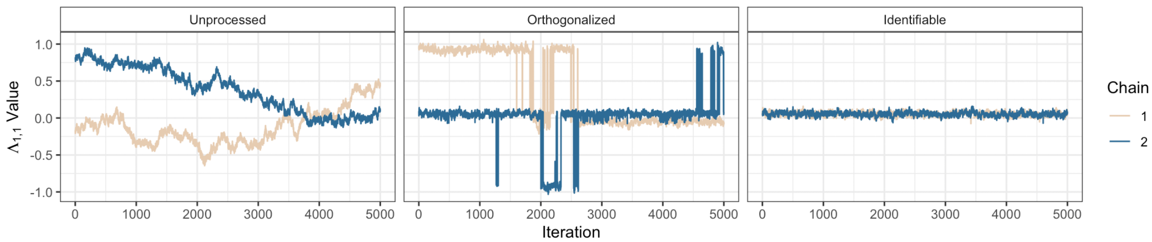

Figure 2.5: Non-identifiability in Λ shown in the trace plot of one element through different stages of processing for two sampling chains.

To solve this problem, we apply the MSF post-processing algorithm described in *Chapter 2* to determine the likely $P^{(i)}$ matrices and apply their inverses. By applying an orthogonalization procedure to each sample of Λ, we limit the $P^{(i)}$ matrices to a set of signed permutation matrices. Then by matching columns the algorithm perform a greedy search for the most likely signs and permutations. This solution is implemented in the `jointRot` function, which performs this orthogonalization and matching on the samples of Λ and applies the resulting inverse permutations to the sampled pairs $(\Lambda^{(i)}, \eta^{(i)})$ to create the approximated identifiable MCMC samples. Other methods exist to solve label switching, but focus on mixture models with component membership probabilities, which is not the case here as illustrated in *Figure 2.5*. All of the operations performed on the samples in post-processing are rotations, and therefore inference on any identifiable elements is unaffected by this post-processing.

Orthogonalization procedures are well studied in the factor analysis literature and multiple popular approaches have been developed including Varimax [Kai58], Quartimax [Jac05], and Promax [HW64]. Each of these methods maximizes a different objective related to the orthogonality of the columns of a factor loadings matrix. These result in slightly different choices of the member of the (Λ, η) equivalence class, but those methods listed and many more result in reducing $\{P^{(i)}\}_{i=1}^{M}$ to a set of signed permutation matrices, making them viable for this method of post-processing. Our package implements the Varimax rotation procedure by default due to popularity and wide implementation.

2.4 Extensions

There are a number of features not included in the core samplers that can be added without major code changes due to the modular parameter update design. The RCPP components output data objects that are readable and editable by R functions, thus these components can be used in combination with additional code in R or RCPP. The existence of missing data is common in real world applications, and the included factor models provide a simple full conditional distribution of the complete data, $X_i = \Lambda\eta_i + \epsilon_i$. For missing at random (MAR) entries of X, a sample of the complete data from this posterior predictive distribution can be imputed in every MCMC iteration for accurate inclusion of uncertainty due to missingness while making use of all available data. This extension modifies none of the parameter updates, and can be included in all of the core samplers. Example code to perform this sampler augmentation is provided on GitHub, defining a function to provide a complete X given the incomplete X and the current values of Λ, η, and Σ. Similar code is provided for the case that X displays missingness due to a limit of detection.

Priors and data models involving semi-parametric factor models are developed regularly, and those that are of interest can be coded for inclusion in this package. The upcoming Cumulative Shrinkage Prior of [LDD19] applies shrinkage to the factor loadings, and can be included in this package framework. This new prior is available for use with both the linear and interaction models, and compatible with the missing data extensions.

2.5 Supplementary Figures

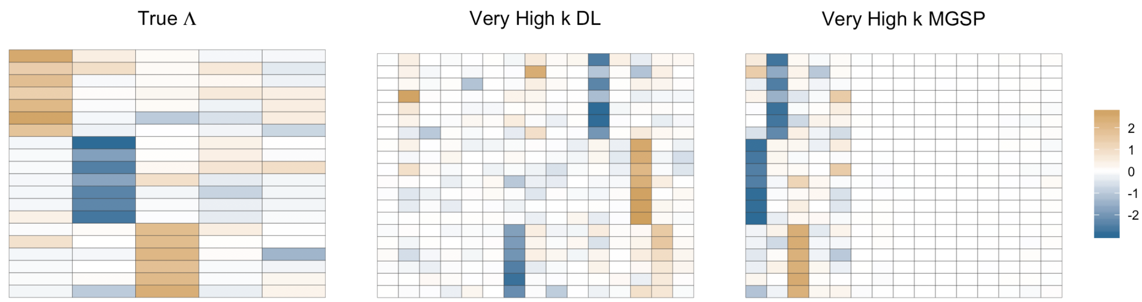

Figure 2.6: True parameter value used in simulation and posterior mean estimates of post-processed factor loadings samples showing the difference in prior specification in the case where k is largely misspecified in sampling. Note the splitting of the lower group and the number of loadings that have a large value for one predictor under the Dirichlet-Laplace prior.

Chapter 3

MSF Algorithm

3.1 Background

In the introduction we presented a typical formulation of a Gaussian factor model as in [Wes03]:

$$X = \eta\Lambda^T + \epsilon, \quad \epsilon_i \sim \mathcal{N}_p(\mathbf{0}, \Sigma)$$

where $\epsilon = [\epsilon_1, \ldots, \epsilon_n]$ and η is a $n \times k$ matrix of latent factors. We note the decomposition of the $p \times p$ covariance matrix $\text{cov}(X_i) = \Lambda\Lambda^T + \Sigma$. In this formulation X and Ω are identifiable matrices, because changes in the values of their elements result in changes in the likelihood. As stated in *Section 2.3*, the likelihood is invariant to rotations in Λ and η:

$$\Omega = \Lambda\Lambda^T + \Sigma = \Lambda PP^T\Lambda^T + \Sigma = (\Lambda P)(\Lambda P)^T + \Sigma$$

$$X = \eta\Lambda^T + \epsilon = \eta PP^T\Lambda^T + \epsilon = \eta P(\Lambda P)^T + \epsilon, \quad P \in \{P : PP^T = I_k\}$$

We refer to this form of non-identifiability as *rotational ambiguity* because this likelihood invariance can result in posterior samples that do not share rotational alignment. We will continue to use the latent factor model as an example but our method can be applied to a much larger class of models involving rotational ambiguity in matrix valued parameters.

A possibility to impose identifiability in Λ is specifying significant constraints a priori [GD09], [EC11]. A typical choice requires $\lambda_{i,j} = 0$ for $j > i$ and $\lambda_{j,j} > 0$ for $j = 1, \cdots, k$ [GZ96]. This structure has been used routinely in numerous settings [LCW+06], to perform inference on Λ, at the cost of introducing order dependence among the factors in the case $k < p$ [CCL+08]. In this chapter we focus on methods of post-processing to recover identifiable samples and perform inference on Λ and η without order dependence..

The effect of rotational ambiguity on inference in frequentist settings can be solved via orthogonalization, using for example Varimax [Kai58], Quartimax [NW54], Promin [LS99], the Orthogonal Procrustes algorithm in [ABHP16] etc. Orthogonalization methods are well studied and commonly used in factor analysis, and the objective functions to enforce maximal orthogonality result in interpretable parameter estimates. The application of these methods reduce the possible rotation matrices P to a set of signed permutation matrices: $P \in \{P : P^T P = I_k;\ \forall i, j\ P_{ij} \in \{-1, 0, 1\}\}$. The rotational ambiguity between samples is then limited to switching in the column labels and column signs, which we will refer to as *label switching* and *sign switching* respectively.

Our goal is to estimate the set of rotation matrices $\{P^{(i)}\}_{i=1}^M$ so that we may perform inference on our sampling chain $\{\Lambda^{(i)} P^{(i)}, \eta^{(i)} P^{(i)}\}_{i=1}^M$ for a chosen rotational alignment $(\Lambda^{(i)}, \eta^{(i)})$. Several algorithms try to resolve label switching: the Pivotal Reordering Algorithm (PRA) [MLMR05], the Equivalence Classes Representative (ERC) algorithm [PI10], and the SEM-based algorithm in [SJW10]. The ECR and SEM algorithms are specific to mixture models with class assignment probabilities, and they are not generalizable to our setting where switching is a product of orthogonalization, not distribution component membership. PRA can be applied to solve label switching in Λ, but the computational cost is very high. For the PRA, a loss function is computed for all the possible permutations of the posterior samples, so applied to the case with label and sign switching scales computationally in k by $k!2^kM$ for a number of posterior samples M. Even for small sample sets the inverse permutations for $k > 5$ can be infeasible to compute in this manner, for instance at $k = 5$ and $p = 50$ we expect 5000 samples to take about 30 hours to align using PRA.

In order to proceed with inference on large matrices that display rotational ambiguity, we develop a greedy algorithm to approximate the solution to the loss function in PRA. We take advantage of the large parameter space to match columns of the orthogonalized matrices with a representative of an alignment choice $(\Lambda^{(i)}, \eta^{(i)})$. In orthogonalized samples column label and sign switches bias ergodic summaries, and in this algorithm we use such differences to inform our alignment, directly reducing the source of bias. We define this algorithm in *Section 3.2*, and cover methods for choosing a representative pivot matrix in *Section 3.2.2*, and technicalities of the matching in *Section 3.2.3*. In *Section 3.3* we apply our algorithm to a simulated dataset, and a dataset of gene expressions. This algorithm is implemented in `jointRot()` function of the `infinitefactor` package.

3.2 Algorithm

In this section we describe our MSF (match-sign-factor) algorithm that solves rotational ambiguity in the posterior samples of non identifiable matrix value parameters. The MSF procedure is summarized in *Algorithm 1.* The three key steps are 1) apply an orthogonalization procedure to each posterior sample, 2) choose a reference matrix (the pivot) and 3) match the columns of each posterior sample to the pivot. For use in the algorithm we define $\Lambda^{(1)}, \cdots, \Lambda^{(M)}$ to be M posterior samples of Λ and let $c_j^{(s)}$ be the j^{th} column of $\Lambda^{(s)}$. Steps 1) and 2) above are "embarassingly parallel" operations, for which samples can be easily processed on multiple cores of a computer. Additionally the orthogonalization step can be completed for each sample completely independently with no shared memory access in a distributed environment. This is currently implemented with parallel computing using the `parallel` package in R and the efficient forking found in Mac and Linux based systems. The alignment `for` loop can be easily parallelized by assigning groups of samples to different processor cores, but each thread must have access to the pivot sample.

Result: Resolve rotational ambiguity in samples of a matrix valued random variable
Input $\{\Lambda^{(s)} : s = 1...M\}$;
for s *in* $1 : M$ **do**
  1. Orthogonalize $\Lambda^{(s)}$;
end
2. Choose a pivot Λ^{P} from $\{\Lambda^{(s)} : s = 1...M\}$;
for s *in 1...M* **do**
  for i *in 1...k* **do**
    4. Compute the norms of the differences between column i of $\Lambda^{(s)}$ and each column of $(\Lambda^{P}, -\Lambda^{P})$;
    5. Record in R the index of the minimum norm value;
    6. Drop the matched column from the pivot;
  end
  7. Reorder and re-sign $\Lambda^{(s)}$ and $\eta^{(s)}$ according to R
end

Algorithm 1: MSF

3.2.1 Orthogonalization

The orthogonalization step is necessary to reduce the problem to label and sign switching and has the benefit of representing the samples in a more interpretable form. In the factor analysis case, orthogonalization of significant factor loadings helps to cluster the covariates, often inducing some sparsity, and methods such as Varimax allow widely understood interpretation of the factor loadings. Oblique rotations

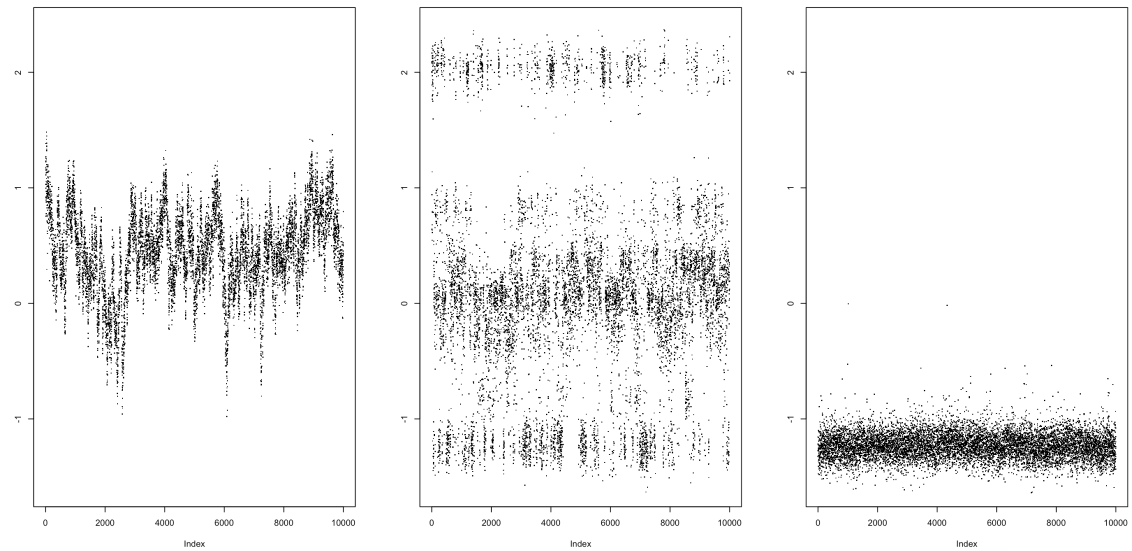

Figure 3.1: Detailed view of samples of one element of Λ with full rotational ambiguity (Left), sign and label switching after the application of Varimax (Center), and complete alignment after using our algorithm (Right)

[Thu47] offer better representations of correlated factors, or more generally linear association in the columns of the non-identifiable matrix. However, some oblique rotation methods show invariance in the minimization objective to a wider class of rotations than just signed permutation matrices. Such rotation methods cannot be used with the MSF

In our approach we focus on orthogonal rotations that are only objective-invariant under signed permutation matrices. Many orthogonalization methods internally perform row normalization. This temporary normalization for algorithmic preconditioning does not affect inference if the inverse scaling is performed before output of the orthogonalized samples. *Figure 3.2* shows an example of the application of Varimax to the posterior samples of Λ, which is the orthogonalization method implemented by default in the `jointRot()` function. The sample shown in that figure remains misaligned with the true value of Λ used in simulation, but contains the same information about group structure. After MCMC samples undergo alignment with the

MSF algorithm, any signed permutation can be applied to all samples in the chain without affecting inference, as there is no "true" alignment. Care should be taken to combine multiple sampling chains before MSF is applied, or ensure the same pivot is used in their separate alignment so that the chains share rotational alignment.

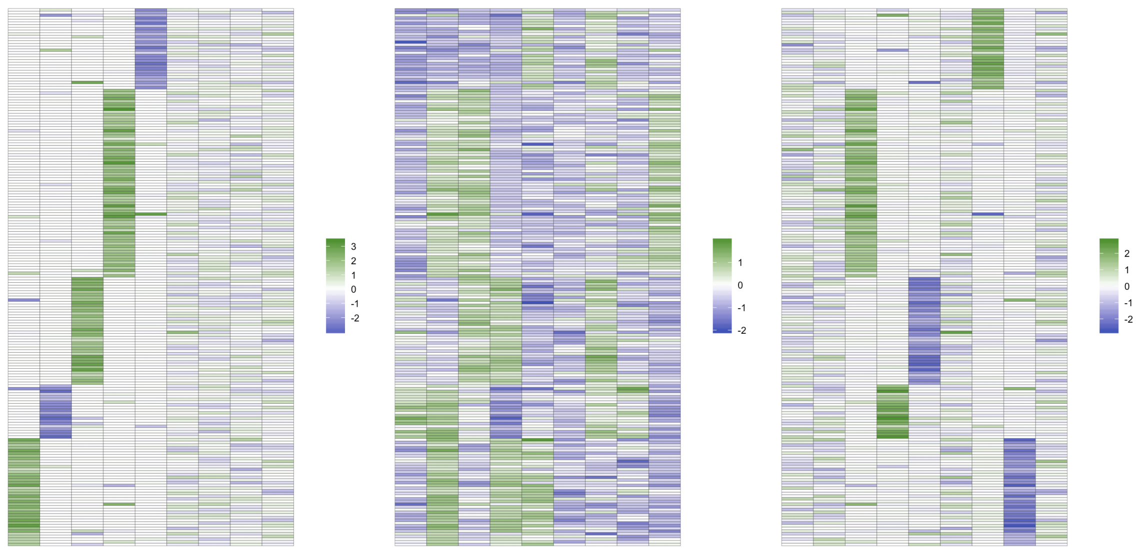

Figure 3.2: (Left) true factor loadings matrix Λ used in simulation, (Center) a posterior sample of Λ and (Right) the same sample of Λ after applying orthogonalization. Note the column switching and sign switching in the samples. In this example $k = 9$, $p = 200$, and $n = 500$.

3.2.2 Choosing a Pivot

We define a pivot $\Lambda^P = [c_1^P \cdots c_k^P]$ as an exemplar Λ matrix that we use to align the posterior samples, as in [MLMR05]. The pivot matrix is used in the algorithm as a proxy of a 'true' Λ. Every column $c_j^{(s)}$ of $\Lambda^{(s)}$ is matched to either the negative or positive of c_h^P, for some $h \in 1, \ldots, k$. In order to correctly match similarly distributed columns, we must choose a pivot that is not a poor representative of any column. This choice is a tuning parameter in the algorithm, and results should not change drastically for different choices of pivot. To choose a pivot without knowing the

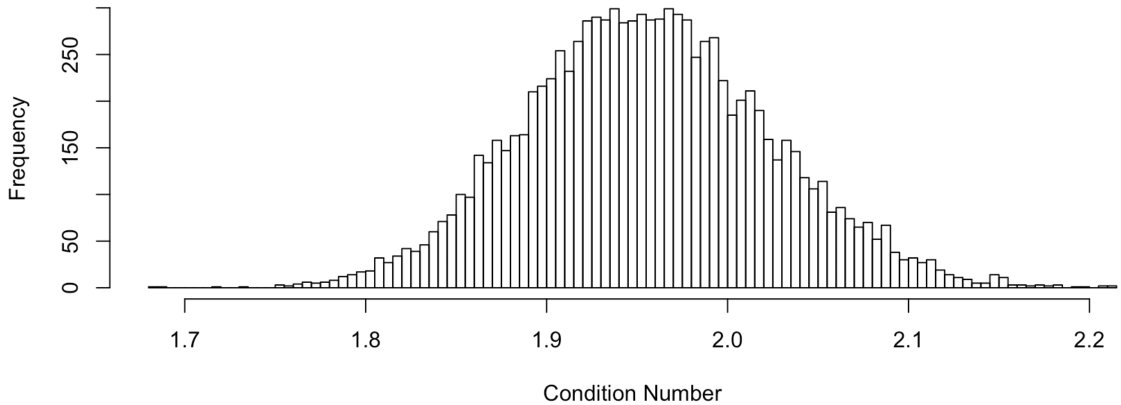

Figure 3.3: Empirical distribution of the condition numbers of samples from the sample set generated in *Section 3.3.1*

posterior column distributions we translate our goal of centrality in each column distribution to centrality in the distribution of information across columns. For this reason, we consider the condition number:

$$\kappa^{(i)} = \kappa(\Lambda^{(i)}) = \frac{\sigma_{max}(\Lambda^{(i)})}{\sigma_{min}(\Lambda^{(i)})}$$

where $\sigma_{max}(\Lambda^{(i)})$ and $\sigma_{min}(\Lambda^{(i)})$ are respectively the largest and smallest singular values of $\Lambda^{(s)}$. An example of the distribution of $\kappa^{(1)}, \ldots, \kappa^{(M)}$ can be seen in *Figure 3.3*. In our implementation of the MSF algorithm we choose as a pivot the matrix with the median condition number. In practice when many samples have very similar condition numbers, the choice of pivot is more arbitrary, and inference should not be affected by choosing any as a pivot. If the distribution of condition numbers is irregular, sensitivity analysis to the choice of pivot should be performed. Note that the value of the condition numbers may approach infinity in the case Λ was sampled with overestimated k. In this case, we can use the largest singular value in place of the condition number. We find in practice that these two choices provide similar performance.

3.2.3 Matching

In order to resolve sign and label switching simultaneously, we "match" each sample column $c_j^{(i)}$, for $j = 1, \dots k$, $i = 1, \dots, M$ with the closest column of $(\Lambda^P, -\Lambda^P)$. Due to sampling noise one c_h^P may be closest to more than one $c_j^{(s)}$. We avoid matching multiple $c_j^{(s)}$s to one c_h^P, which we refer as *duplication*, in order to ensure the application of MSF performs a rotation on each sample. Alignment that involves duplicating columns prevents the accurate reconstruction of the identifiable parameters after post-processing. Duplication of columns also introduces numerical instability to many operations performed on posterior samples.

To find the inverse permutations to align the samples we attempt to minimize a similar global loss function as in the Pivotal Reordering Algorithm [MLMR05]:

$$P^{(i)} = \arg\min_{P \in \mathcal{P}} ||\Lambda^{(i)} P - \Lambda^P||_2,$$

where $\mathcal{P} = \{P : P^T P = I_k;\ \forall i, j\ P_{ij} \in \{-1, 0, 1\}\}$ is the set of all signed permutations of size k. The loss function penalizes differences between the pivot and the posterior samples, and to minimise it we must find the column label and sign choices $c_1^{(s)}, \dots, c_k^{(s)}$ that best match the columns of pivot.

This problem is NP hard, and we can only find the true minimizer by computing the loss for every possible $P \in \mathcal{P}$. Instead, to approximate the true minimizer the MSF algorithm minimizes the loss iteratively by column. The solution to the MSF algorithm may not minimize the global loss $||(\Lambda^{(i)})P - \Lambda^P||_2$ if that global optimum requires a suboptimal match between columns of the sample and pivot. In such a case MSF would fail to find the true minimizer and PRA would succeed. The number of vector norms computed for one sample in MSF is $k(2k+1)$, which is much less than the $k!2^k$ computations of the global loss required for the complete search.

In the `infinitefactor` package implementation of MSF we use the vector L_2 norm to compare the columns of $\Lambda^{(s)}$ and Λ^P. We choose this norm because it matches the objective function of the Varimax algorithm. Other rotation schemes such as Quartimax may work better with an implementation of MSF that uses the one norm or the sup norm. Simulations show that the outcome of MSF is not sensitive to the choice of the norm, and this is supported by the approximate equivalence of norms in finite dimensional space.

In practice columns with larger norms should be matched first to prevent accidental mismatches. Whenever using a prior that introduces increasing shrinkage as the column order increases, as in [BD11] or [LDD19], we can start matching the columns starting from $c_1^{(s)}$ and proceed in order. In practice we should only consider other orders of matching if the prior choice on Λ introduces decreasing shrinkage as the column order increases. Several methods for factor models take an "over-fitted" approach, related to [BD11] and [RM11], and choose k to correspond to an upper bound on the number of factors. In this scenario, after orthogonization, we might have multiple columns centered near $\mathbf{0}$. The MSF algorithm may not detect when these columns switch labels or signs between samples, as a result of the lack of unique information contained in each column. However, a trailing set of columns near to zero are not of interest for inference on group structure in the covariates.

This method of matching is consistent with respect to n so long as the original model used to generate the sample is consistent in estimating this (Λ, η) pair. For convenience we assume that orthogonalization has already been performed using whatever method is preferred, and the problem has been reduced to finding the set of M signed permutation matrices $\{P^{(i)}\}_{i=1}^M$.

Theorem 1. *For a consistent posterior distribution, and for posterior samples of* Λ *that have undergone orthogonalization, the result of the MSF algorithm converges in probability to a member of the true* (Λ, η) *equivalence class as* $n \to \infty$

Proof. Assuming that there is a true equivalence class of factor loadings $(\Lambda^0 R, \eta^0 R)$ for any orthogonal matrix $R : R^T R = I$, where we will take Λ^0 to be an orthogonalized member of the class. For a consistent posterior distribution $\mathcal{P}$, $\Lambda \sim \mathcal{P}$ implies that for given $\epsilon > 0$, as $n \to \infty$, $\Pr(||\Lambda - \Lambda^0||_F^2 > \epsilon) \to 0$ for orthogonalized Λ that is aligned with Λ_0. Let $\Lambda^s, \Lambda^p \sim^{iid} \mathcal{P}$ be our sample and pivot respectively. We have that for any column l of our pivot $\Pr(\min_{j \in \{1,\ldots,2k\}} ||\Lambda_l^p - (\Lambda^s, -\Lambda^s)_j||_2^2 > \epsilon) \to 0$ as $n \to \infty$. Let $h_l = \arg\min_{j \in \{1,\ldots,2k\}} ||\Lambda_l^p - (\Lambda^s, -\Lambda^s)_j||_2^2$. Say Λ^0 shares column label and sign alignment with the pivot, then $\Pr(||\Lambda_l^0 - (\Lambda^s, -\Lambda^s)_{h_l}||_2^2 > \epsilon) \to 0$ as $n \to \infty$. Thus columns are matched correctly. $\square$

Note that we compare to the Λ^0 that shares column label and sign alignment with our pivot, which is permitted because the true equivalence class contains all column permutations and sign switches of Λ^0. This proof does not omit the possibility of multiple columns sharing a limit point and being matched with each other for any n, but due to the orthogonalization the only limit point that can be shared by multiple columns is $\mathbf{0}$.

3.3 Examples

Both the `infinitefactor` package for R and the MSF algorithm are motivated by a Gaussian factor model, so we provide two examples of that model in use here. The first is a simulation study analogous in parameter value and structure to an application in epidemiology, where we often see between tens and hundreds of covariates that form relatively strong groups based around chemical metabolites or compound classifications. The second is an application to real gene expressions data with very small n and very large p.

In both of these cases we estimate the posterior distribution of the parameters in a linear latent factor model using the functions in the `infinitefactor` package. We do not enforce any constraints on the matrix Λ. We consider the Dirichlet Laplace prior [BD11] on the rows of the factor loadings matrix Λ, an isotropic normal prior on η, and an inverse-Gamma prior on the diagonal elements of the matrix Σ, which is implemented in the `linearDL()` function. The DL prior provides flexible shrinkage on the factor loadings matrix, generalizing the Bayesian Lasso [PC08] to have a careful chosen hierarchical structure on covariate-specific and local shrinkage. This induces a prior with concentration at zero, to strongly shrink small signals, and heavy-tails, to avoid over-shrinking large signals. The DL prior induces near sparsity row-wise in the matrix Λ. This a priori belief that that the rows of Λ contain few large values is applicable to both of these modelling problems.

3.3.1 Simulation

We simulate the data according to the latent factor model presented in the introduction. We are particularly motivated by data in environmental epidemiology such as NHANES[JPRO+13], where the covariates are measured chemical concentrations and the interest is in discovering relationships between covariate groups learnt by the the latent factor model. Following the structure of common chemical exposures data we initialize a factor loadings matrix with five columns representing major chemical groups and four normally distributed columns representing differential exposure tendencies. The simulated $p = 200 \times k = 9$ factor loadings matrix is displayed on

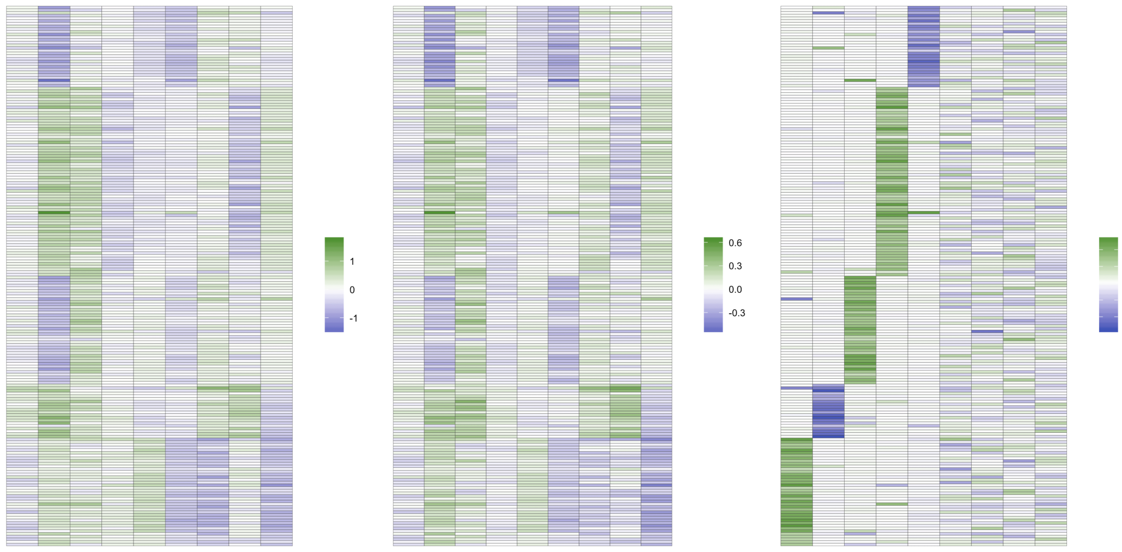

Figure 3.4: (Left) The posterior sample mean of Λ before post processing. (Center) The posterior sample mean of Λ after each sample is rotated by the Varimax algorithm. (Right) The posterior sample mean of Λ after alignment by MSF.

the left of *Figure 3.5* under Varimax orthogonalization. We simulate η from its prior distribution where we set $n = 500$. We set $\sigma_j^2 = 1$ for $j = 1, \ldots, p$. We construct X from the simulated Λ and η parameters with added noise: $X = \eta\Lambda^T + \epsilon$ where $\epsilon_i \sim \mathcal{N}_p(\mathbf{0}, \Sigma)$.

To perform sampling, we pass the data matrix X and the sampling parameters to the `linearDL()` function. This is not a case that requires modification of the shrinkage hyperparameters, and a random initialization of the latent factors and factor loadings is suitable, so no function modifications must be performed. We set the sampling chain length to 10,000, and discard the first 5,000 samples for a burn in period to allow the sampler to reach approximate stationarity. We set the value of k used in sampling to the true value of k for illustration. The output of the sampling function is the list of posterior samples on which we can perform inference. The resulting samples of the Λ and η parameters display the rotational ambiguity discussed previously.

We apply the MSF algorithm to each posterior sample of the factor loadings matrix to orthogonalize the matrices, using the `jointRot` function. We use the Varimax objective in orthogonalization of the samples, which is implemented by default. The resulting posterior mean estimate of the factor loadings matrix visually matches the true case we simulated from, as seen in *Figure 3.5*. To illustrate the motivating problem, *Figure 3.4* shows the factor loadings posterior mean estimate at different stages of sample processing. Even after Varimax rotation the label and sign switching heavily biases this estimate. The values of the elements of the estimated Λ are lesser in magnitude than the corresponding elements in the true Λ because of the shrinkage performed by the Dirichlet-Laplace shrinkage prior. Similar to the effect of the Bayesian Lasso, these parameter values are biased toward 0, and this shrinkage will not create false significance in hypothesis testing. The group structure is recovered visually, and hypothesis tests on the inclusion of a covariate in a factor loading are now possible from the post-processed samples of Λ. The computational time required for the alignment of the samples was 61 seconds on a single thread of a 4.5GHz processor. Of the total alignment time, the Varimax orthogonalization step ran for 60 seconds and the MSF matching procedure ran for 0.6 seconds until completion In comparison, the same samples were passed to the implementation of the PRA algorithm in the `label.switching` package and ran for 114 hours before completion on the same computer. The resulting mean estimates from the two alignment algorithms are shown in *Figure 3.5*.

In the previous example, it is not difficult to assess visually the structure in Λ to determine the success or failure of the alignment. This is generally the case in epidemiology, but other applications involve thousands of covariates and prevent comparison of the results. In this scenario we need to report a numeric assessment

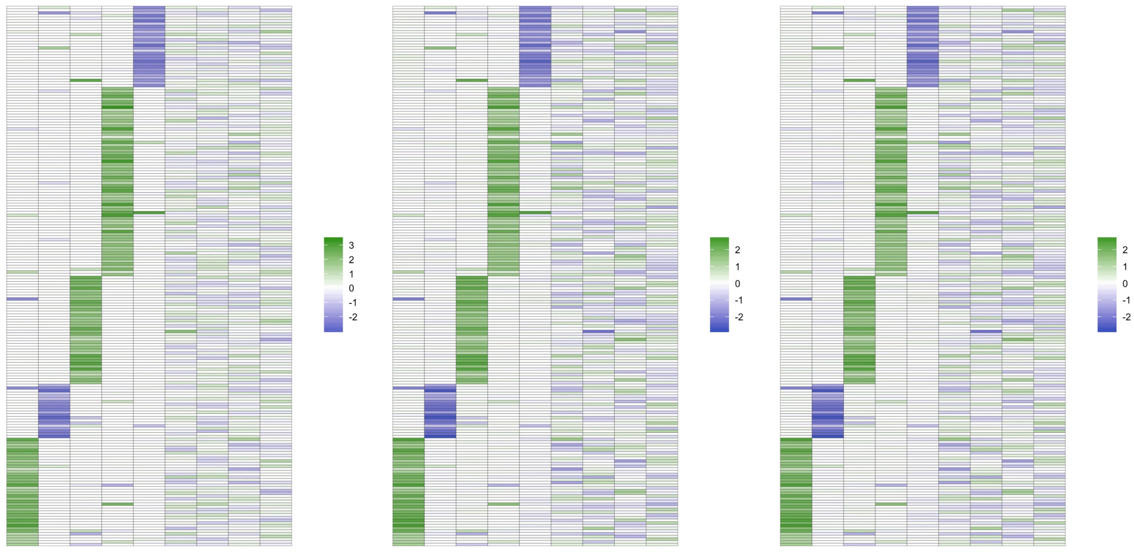

Figure 3.5: (Left) A member of the true Λ equivalence class under Varimax rotation. (Center) Posterior sample mean after applying MSF. (Right) Posterior sample mean after applying PRA. Note that mean estimates have been aligned in column order and sign with the true case for comparison

of sample alignment. One possibility is to use

$$||\overline{\Lambda\Lambda^T} - \bar{\Lambda}_*\bar{\Lambda}_*^T||_F$$

where $||\cdot||_F$ is the Frobenius norm. This summary compares the sample mean of an identifiable element $\overline{\Lambda\Lambda^T} = \frac{1}{M}\sum_{i=1}^{M}\Lambda^{(i)}(\Lambda^{(i)})^T$ and the same element constructed from the sample mean of the processed factor loadings samples $\bar{\Lambda}_* = \frac{1}{M}\sum_{i=1}^{M}\Lambda_*^{(i)}$. This criterion follows from the intuition that conclusions about Λ should not largely deviate from our understanding of any identifiable parameters. This difference will never be exactly 0 when sampling variation exists, but we expect the norm value to increase for values of $\bar{\Lambda}_*$ estimated using rotationally mis-aligned samples. This metric does not necessitate knowing a true equivalence class for (Λ, η), so it can be used on real data. The results of this metric calculated for the simulated case are found in *Table 1*. We find that only rotating the samples, which is necessary for

Table 3.1: Numeric summary of alignment of samples of factor loadings in the simulated data example with and without column scaling. Higher values indicate misalignment.

Scaling	Unprocessed	Rotated	Matched	P.R.A.
Unscaled	287.4098	395.1234	17.12201	17.12201
Scaled	157.1791	157.9243	8.42244	8.42244

interpretation and makes individual samples interpretable, increases this metric, but full application of the MSF and PRA algorithms results in much reduced values.

3.3.2 Genomics Application

Dimension Reduction is key in genomics, where data often present linearly dependent features in high dimensional spaces. Here we model the linear relationships in normalized pancreatic tissue gene expression data from the Broad Institute GTEx database [1]. This dataset contains 149 observations of 24112 gene expression levels. We initially perform exploratory analysis to verify the joint normality assumption of the Gaussian linear factor model, which is largely satisfied in this case. There are no missing data in the data matrix X, so we pass this matrix as input directly into the sampling function.

We use the `linearDL()` to perform sampling under the chosen Dirichlet-Laplace prior. The default specification for shrinkage hyperparameters is suitable for this sampling. The choice of latent dimension size is difficult in this case, and due to the relatively small number of observations we choose to sample with $k = 10$. For comparison with the simulated case, we sample one chain for 10,000 iterations with 5,000 iterations discarded in burn-in. Without theoretical or computational issues, any number of sampling chains can be run in parallel on a suitable computer to

[1] The Genotype-Tissue Expression (GTEx) Project was supported by the Common Fund of the Office of the Director of the National Institutes of Health, and by NCI, NHGRI, NHLBI, NIDA, NIMH, and NINDS.

Table 3.2: Numeric summary of alignment of samples of the factor loadings in the gene expressions data with and without column scaling. Higher values indicate misalignment.

Scaling	Unprocessed	Rotated	Matched
Unscaled	40233.11	40050.28	7701.999
Scaled	16542.05	20071.55	1861.084

generate a suitable number of posterior samples with a large parameter space such as this one. We apply the MSF algorithm using the `jointRot()` function and the default Varimax rotation to the samples. We compare the alignment of the samples after post-processing numerically using the metric developed in *Section 3.3.1.* The results are displayed in *Table 2* and show that alignment with the MSF algorithm results in much less disparity between the covariance surrogates. The computation time required to perform Varimax rotation on the samples was 4.1 hours, and the computation time to perform matching was 62 seconds on a single thread of a 4.5GHz processor. The computational scaling of this matching step for larger p is limited to the increased complexity of taking the vector norm, but this can be efficiently parallelized using openBLAS or other multi-threaded linear algebra libraries. Comparison is not performed for this example with PRA due to the excessive computational time required for the alignment.

Chapter 4

Conclusion

In this thesis we have presented two major tools for semi- and non- parametric Gaussian factor modeling. These models are applicable to a wide range of problems, and we aim to make the recently developed priors for these models as accessible to researchers as possible. The `infinitefactor` package for R provides the implementations of significant theoretical work and creates a platform for future development. The open source nature of the distribution license and the CRAN philosophy encourages the use of any elements of this package in future work. The computational advantages of C++ and the Armadillo library are available to any R function that makes use of a full conditional parameter update component through the `RCPP` package. In addition to the implementation of previous theoretical work, we also present a new algorithm for the resolution of rotational ambiguity in Bayesian posterior samples of matrices. This post-processing algorithm removes the need for identifiability constraints on factor loadings, and allows inference on the factor loadings and factors under the prior specifications in the `infinitefactor` package.

Throughout the work presented here, there is a strong focus on computational efficiency and scaling. Computational efficiency can allow the application of good models to new problems, that were previously inestimable or not considered due to sheer cost. To make Bayesian latent factor models applicable to genomics, it is necessary to consider the computational efficiencies possible in every step of sampling and processing. These efficiencies come from both the coding and processing perspective, where we take advantage of lower level languages and compiled functions, as well as the mathematical perspective, where we identify simplifications in linear algebra and make good approximations. The computationally efficient code developed for this thesis is available for use in any model through the modular sampler design and wide applicability of the MSF algorithm.

Work will continue on this package to include novel models and priors as they are developed. There has already been success in both parts of our goal for the use of this codebase. Researchers are already using the samplers and MSF algorithm for inference, and other programmers are building component functions for inclusion in the package framework. We hope that this work will continue and the strengths of Gaussian factor models will be applied wherever they can be.

Bibliography

[ABHP16] Christian Aßmann, Jens Boysen-Hogrefe, and Markus Pape. Bayesian analysis of static and dynamic factor models: An ex-post approach towards the rotation problem. *Journal of Econometrics*, 192(1):190–206, 2016.

[BD11] A. Bhattacharya and D. B. Dunson. Sparse Bayesian infinite factor models. *Biometrika*, pages 291–306, 2011.

[BPPD15] Anirban Krishna Bhattacharya, Debdeep Pati, Natesh S. Pillai, and David B. Dunson. Dirichlet-Laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, 110 512:1479–1490, 2015.

[CCL+08] Carlos M Carvalho, Jeffrey Chang, Joseph E Lucas, Joseph R Nevins, Quanli Wang, and Mike West. High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008.

[Dur17] Daniele Durante. A note on the multiplicative gamma process. *Statistics & Probability Letters*, 122:198–204, 2017.

[EC11] Elena A Erosheva and S McKay Curtis. Dealing with rotational invariance in bayesian confirmatory factor analysis. *Department of Statistics, University of Washington, Seattle, Washington, USA*, 2011.

[FD19] Federico Ferrari and David B Dunson. Bayesian factor analysis for inference on interactions. *arXiv preprint arXiv:1904.11603*, 2019.

[GD09] Joyee Ghosh and David B. Dunson. Default prior distributions and efficient posterior computation in bayesian factor analysis. *Journal of Computational and Graphical Statistics*, 18(2):306–320, 2009. PMID: 23997568.

[GR+92] Andrew Gelman, Donald B Rubin, et al. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

[GZ96] John Geweke and Guofu Zhou. Measuring the pricing error of the arbitrage pricing theory. *The review of financial studies*, 9(2):557–587, 1996.

[HW64] Alan E. Hendrickson and Paul Owen White. Promax: A quick method for rotation to oblique simple structure. *British Journal of Statistical Psychology*, 17(1):65–70, 1964.

[Jac05] J Edward Jackson. Quartimax rotation. *Encyclopedia of Biostatistics*, 6, 2005.

[JPRO+13] Clifford L. Johnson, Ryne Paulose-Ram, Cynthia L. Ogden, Margaret D. Carroll, Deanna Kruszan-Moran, Sylvia M. Dohrmann, and Lester R. Curtin. National health and nutrition examination survey. analytic guidelines, 1999-2010. *Vital and health statistics. Series 10, Data from the National Health Survey*, 161, 2013.

[Kai58] Henry F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, Sep 1958.

[LCW+06] Joe Lucas, Carlos Carvalho, Quanli Wang, Andrea Bild, Joseph R Nevins, and Mike West. Sparse statistical modelling in gene expression genomics. *Bayesian inference for gene expression and proteomics*, 1:0–1, 2006.

[LDD19] Sirio Legramanti, Daniele Durante, and David B Dunson. Bayesian cumulative shrinkage for infinite factorizations. *arXiv preprint arXiv:1902.04349*, 2019.

[LS99] Urbano Lorenzo-Seva. Promin: A method for oblique factor rotation. *Multivariate Behavioral Research*, 34(3):347–365, 1999.

[MLMR05] Jean-Michel Marin, Kerrie L. Mengersen, and Christian Robert. Bayesian modelling and inference on mixtures of distributions. *Handbook of Statistics*, 25, 12 2005.

[NW54] Jack O Neuhaus and Charles Wrigley. The quartimax method: An analytic approach to orthogonal simple structure 1. *British Journal of Statistical Psychology*, 7(2):81–91, 1954.

[PC08] Trevor Park and George Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[PI10] Panagiotis Papastamoulis and George Iliopoulos. An artificial allocations based solution to the label switching problem in bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics*, 19(2):313–331, 2010.

[RM11] Judith Rousseau and Kerrie Mengersen. Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):689–710, 2011.

[SJW10] M. Sperrin, T. Jaki, and E. Wit. Probabilistic relabelling strategies for the label switching problem in bayesian mixture models. *Statistics and Computing*, 20(3):357–366, Jul 2010.

[Thu47] L. L. Thurstone. *Multiple-factor analysis*. Univ. of Chicago Press, 1947.

[Wes03] Mike West. Bayesian factor regression models in the "large p, small n" paradigm. In *Bayesian Statistics*, pages 723–732. Oxford University Press, 2003.