Heterogeneous Large Datasets Integration Using Bayesian Factor Regression

Alejandra Avalos-Pacheco*,†,‡, David Rossell¶∥, and Richard S. Savage§

Abstract. Two key challenges in modern statistical applications are the large amount of information recorded per individual, and that such data are often not collected all at once but in batches. These batch effects can be complex, causing distortions in both mean and variance. We propose a novel sparse latent factor regression model to integrate such heterogeneous data. The model provides a tool for data exploration via dimensionality reduction and sparse low-rank covariance estimation while correcting for a range of batch effects. We study the use of several sparse priors (local and non-local) to learn the dimension of the latent factors. We provide a flexible methodology for sparse factor regression which is not limited to data with batch effects. Our model is fitted in a deterministic fashion by means of an EM algorithm for which we derive closed-form updates, contributing a novel scalable algorithm for non-local priors of interest beyond the immediate scope of this paper. We present several examples, with a focus on bioinformatics applications. Our results show an increase in the accuracy of the dimensionality reduction, with non-local priors substantially improving the reconstruction of factor cardinality. The results of our analyses illustrate how failing to properly account for batch effects can result in unreliable inference. Our model provides a novel approach to latent factor regression that balances sparsity with sensitivity in scenarios both with and without batch effects and is highly computationally efficient.

Keywords: Bayesian factor analysis, EM, non-local priors, shrinkage.


1 Introduction

A first important task when dealing with large datasets is to conduct an exploratory analysis. Dimensionality reduction techniques have proven a highly popular tool for this purpose. Those techniques provide a lower-dimensional representation that can give insights into the underlying structure to visualise, denoise or extract meaningful features from the data. See Johnson and Wichern (1988, chap. 3) or Hastie et al. (2001,
Large datasets are common in modern statistical applications. For instance, technological advances in bioinformatics such as high-throughput sequencing, microarrays, mass spectrometry and single cell genomics allow the gathering of a vast amount of biological data, enabling researchers to create models to explain the complex processes and interactions of biological systems (see Bersanelli et al. (2016) for a recent review). Cancer is a prominent example. Large-scale projects such as The Cancer Genome Atlas (TCGA), Cancer Genome Project (CGP) and the International Cancer Genome Consortium (ICGC), as well as many individual laboratories are generating extensive amounts of biological data (e.g. gene expression, mutation annotation, DNA methylation profiles, copy number changes) in addition to recording other covariates (e.g. gender, tumour stage, medical treatment and patient history). These projects aim to give a better understanding of the disease and improve prognosis, prevention and treatment. However, the large and heterogeneous nature of the data make the analyses and interpretations challenging. Furthermore, such data are often generated under different experimental conditions, when new samples are incrementally added to existing samples, or in analyses coming from different projects, laboratories, or platforms; collecting data in this matter often produces batch effects (Rhodes et al., 2004). These, unless properly adjusted for, may lead to incorrect conclusions (Leek et al., 2010; Goh et al., 2017). In the context of bioinformatics, several approaches have been developed for removing batch effects (see Scherer (2009) for a review and examples). These include data “normalization” methods using control metrics or regression methods (Schadt et al., 2001; Yang et al., 2002), matrix factorisation (Alter et al., 2000; Benito et al., 2004) and location-scale methods (Leek and Storey, 2007; Johnson and Li, 2009; Parker et al., 2014; Hornung et al., 2016). Strategies for batch effect correction include data pre-processing, for example via the so-called ComBat empirical Bayes approach (Johnson et al., 2007) or via singular value decomposition (SVD) (Leek and Storey, 2007). As shown in our examples applying standard dimension reduction methods on such normalised data can produce unreliable results. Intuitively this is due to using a two-step rather than a joint inference procedure on batch effects and dimension reduction. Our examples focus on cancer-related gene expression; nonetheless, batch effects are also present in many other settings, e.g. structural magnetic resonance imaging (MRI) data from Alzheimer’s disease (Shinohara et al., 2014; Fortin et al., 2016), multiple sclerosis (Shah et al., 2011), attention deficit hyperactivity disorder (Olivetti et al., 2012) or even different tissues of marine mussels (Avio et al., 2015).

We address dimensionality reduction via a model-based framework relying on Bayesian factor analysis and latent factor regression. Our model builds on the approaches introduced by Lopes and West (2004); Lucas et al. (2006); Carvalho et al. (2008) and Ročková and George (2017). An important practical extension of these works is to increase the flexibility to account for systematic biases or sources of variation that do not reflect any underlying patterns of interest, i.e. batch effects. Our main contribution is to provide a model-based approach for tackling dimensionality reduction and batch effect correction simultaneously, avoiding the use of two-step procedures. Another
important contribution is to develop a scalable non-local prior based formulation to induce sparsity and learn the underlying number of factors; for this we provide a prior parameter elicitation, of practical importance to increase power to detect non-zero loadings. A strategy related to ours is to use factor models to learn, on the one hand, the biological patterns via common factors shared across the different data sources and, on the other hand, the non-common sources of variation via data-specific factors (De Vito et al., 2018b,a). However, such a strategy is not designed for batch effects and requires MCMC estimation, making the inference slower. Another related approach is to regress the covariance on batches and other explanatory variables, either parametrically or non-parametrically (Hoff and Niu, 2012; Fox and Dunson, 2015). While useful, this method is not focused on dimension reduction and does not lead to sparse factor loadings that facilitate interpretation and, as shown in our examples, can improve inference.

We model observations with a regression on latent factors with sparse loadings, observed covariates, and batch effects that can alter the mean and intrinsic variance structures. Model fitting is done via a novel Expectation-Maximisation (EM) algorithm to obtain maximum posterior mode parameter estimates in a computationally efficient manner. We focus on three different continuous prior formulations for the loadings: flat, Normal-spike-and-slab (George and McCulloch, 1993) and a novel Normal-spike-and-MOM-slab, based on a continuous relaxation of the non-local prior configuration by Johnson and Rossell (2010, 2012). We also discuss non-local Laplace-tailed extensions, along the lines of Roˇckov´a and George (2017). Spike-and-slab priors provide sparse loadings, effectively performing model selection on the number of required factors and non-zero loadings. We obtain closed-form EM updates, a novel contribution to the non-local prior literature. As we will discuss later, the main advantage of non-local priors in this setting is to help achieve a better balance between sparsity and sensitivity in inferring non-zero loadings. To our knowledge, this is the first adaptation of non-local priors to factor models. See also Bar et al. (2018) who argued for improved sensitivity via 3-component mixture priors that resemble non-local priors in generalised linear models, and Shi et al. (2019) for an application to linear regression via Gibbs sampling.

Our work is meant to contribute to substantive applied aspects in data analysis that we show via examples to be of practical relevance. Firstly, as far as dimension reduction is concerned, we provide sparse solutions that increase the quality of our estimations (see Section 5.1). Secondly, with regard to batch effect correction, we model dependence structure across batches. As a motivating example, Figure 5 (top row) displays systematic differences in mean and variance, thus revealing the problem of not accounting for batch effects. These batches represent comparable patients in terms of disease characteristics, hence we view them as exchangeable realisations from a common distribution and as such they should show the same distribution in any low dimensional projection. After two-step procedures most of these differences are corrected, but distinct covariances are still present across batches (see rows 2 and 3). Thirdly, as a prediction tool for survival analysis in ovarian, lung and colon cancer datasets, we provide competitive concordance indexes (see Section 5.3). Furthermore, our method could be of use for downstream analyses. Finally, from the computational viewpoint, via our model-based approach we study high-dimensional sparse models that facilitate the use of non-local priors to applications.
The outline of this paper is as follows. Section 2 reviews latent factor regression and introduces our extension, which includes a variance batch effect adjustment. Section 3 proposes prior formulations including non-local priors on the loadings and important aspects related to prior parameter elicitation. Section 4 describes several EM algorithms for model fitting, parameter initialisation and post-processing steps required for effective model selection and dimension reduction. Section 5 presents applications on simulations and on cancer datasets under unsupervised and supervised settings. Section 6 concludes. The supplementary material contains the derivation of the EM algorithm and additional results. Software implementing our methodology is available at https://github.com/AleAviP/BFR.BE.

2 Latent factor regression with batch effects

Consider vectors \( x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^p \), observed for \( i = 1, \ldots, n \) individuals. The factor regression model defines \( x_i \) as a regression on \( p_v \) observed covariates denoted by \( v_i \in \mathbb{R}^{p_v} \), and \( q \) low-dimensional latent variables denoted \( z_i \in \mathbb{R}^q \), also known as latent coordinates or factors. Let \( X \) be the \( n \times p \) matrix with the \( i \)th row equal to \( x_i^\top \), \( V \) the \( n \times p_v \) matrix of known covariates with the \( i \)th row equal to \( v_i^\top \) and \( Z \) the \( n \times q \) matrix of latent coordinates, containing \( z_i^\top \) in the \( i \)th row. The standard factor regression model is

\[
x_i = \theta v_i + M z_i + e_i,
\]

where \( \theta \in \mathbb{R}^{p \times p_v} \) is the matrix of regression coefficients, \( M \in \mathbb{R}^{p \times q} \) is the matrix of factor loadings and \( e_i \in \mathbb{R}^p \) is the error, distributed as \( e_i \sim N(0, \Sigma) \) independently across \( i = 1, \ldots, n \), where \( \Sigma \) is a diagonal matrix. Factors are assumed to be standard normal, \( z_i \sim N(0, I) \), independent across \( i = 1, \ldots, n \) and also independent of \( e_i \).

Equation (2.1) regresses the observed data \( X \) on known covariates and on a latent factor structure. In particular, it allows additive batch effects to be accounted for by incorporating the variables recording the batches into \( v_i \). However, in practice one often observes more complex batch effects; specifically in bioinformatics it is common to observe multiplicative effects on the variance (Johnson et al., 2007). We will later describe an example of this, shown in Figure 5. Such artefacts cannot be captured by (2.1) given that \( \Sigma \) is assumed constant across all individuals.

To address this issue we extend (2.1) by allowing \( \Sigma \) to depend on \( i \). Suppose the data were obtained in \( p_b \) batches, e.g. from different days, laboratories or instrumental calibrations, with \( n_l \) individuals in batch \( l \), for \( l = 1, \ldots, p_b \), such that \( n_1 + n_2 + \cdots + n_{p_b} = n \). Let \( b_i \) be the indicator vector of length \( p_b \) defined as \( b_{il} := 1 \) if individual \( i \) is in batch \( l \), \( b_{il} := 0 \) otherwise.

We incorporate batch effects by adding a mean and variance adjustment. We let

\[
x_i = \theta v_i + M z_i + \beta b_i + e_i,
\]

where \( \theta, v_i, M \) and \( z_i \) are as (2.1), \( \beta \in \mathbb{R}^{p \times p_b} \) captures additive batch effects and the variance of \( e_i \) captures multiplicative batch effects. We denote by \( \tau_{jl}, j = 1, \ldots, p \) and
l = 1, . . . , pb as the jth idiosyncratic precision element in batch l. Then, given bi = 1,
the errors are independently distributed as ejj ∼ N(0, τjl−1). Further, denote by T
the p × pb matrix that has τjl as its (j, l) element.

To help interpret the practical implications of the model, suppose that one has
orthonormal factor loadings M ⊤M = I. Then (2.2) implies

\[ z_i = M^\top (x_i - (\theta v_i + \beta b_i + e_i)) \]  (2.3)

and thus, \( E(z_i | x_i, v_i, b_i, M, \theta, \beta) = M^\top x_i - M^\top \theta v_i - M^\top \beta b_i \). That is, the mean of the
latent coordinates is the projection \( M^\top x_i \) plus a translation given by the batch effect
adjustment and (potentially) the observed covariates. An interesting observation is that
their covariance \( \text{Cov}(z_i | x_i, v_i, b_i, M, \theta, \beta, T) = M^\top T_{b_i}^{-1} M \) depends on the batch bi.
As an example, the top-left panel in Figure 5 shows the first two factors of an ovarian
dataset that contains two batches. The latent coordinates of these batches exhibit a
different mean and variance. The middle-left panels show the results after applying the
method ComBat to standardise the mean and variance across batches. After ComBat
correction, the latent coordinates exhibit the same mean and variance across batches (as
expected) but a very different covariance structure. Given that patients in both batches
are believed to be roughly exchangeable, such a difference in covariance is likely due
to technical artifacts and to a two-step procedure that fits the factor model separately
from the adjustment step. The bottom-left panel illustrates that estimating \( T \) jointly
with \( (M, \theta, \beta) \) via our proposed methods addresses this issue; in particular, both batches
exhibited similar mean, variance and covariance structure.

Model (2.2) can be represented in matrix notation as

\[ X = V \theta^\top + Z M^\top + B \beta^\top + E, \]  (2.4)

where \( E \in \mathbb{R}^{n \times p} \) is the matrix of errors.

The latent factor model is non-identifiable up to orthogonal transformations, of the
form \( M^* = A^\top M^\top \) and \( Z^* = ZA \), where A is any orthogonal q × q matrix. Thus, the
factor model in (2.4) can equivalently be rewritten as \( X = V \theta^\top + Z^* M^* + B \beta^\top + E \). To
obtain unique point estimates of M and Z, several alternative prior specifications have
been developed. One option is restricting the parameter space. Seber (1984) constrained
M such that \( M^\top \Omega M \) is diagonal. Lopes and West (2004) restricted M to be lower-
triangular with a strictly positive diagonal, \( m_{jj} > 0 \), and assumed M to be full-rank.
More recently, Frühwirth-Schnatter and Lopes (2018) suggested a factor reordering
via a Generalized Lower Triangular loading matrix. However, under this approach the
interpretation of M depends on the arbitrary ordering of the columns in X, and it
gives special roles to the first factors. Another option is to encourage sparsity in M,
e.g. the classical varimax solution (Kaiser, 1958) maximises the variance in the squared
rotated loadings. A more modern strategy is to favour sparse solutions containing exact
zero loadings, e.g. Ročková and George (2017) proposed an EM algorithm that seeks
rotations based on a so-called Parameter Expansion (PX) that aims to avoid local
suboptimal regions. We adopt a similar strategy where sparse solutions are preferred by
the introduced non-local penalties.
In this paper, the main reasons for considering a sparse $M$, rather than addressing identifiability, are the following: First, a sparse $M$ facilitates interpretation of factors as linear combinations of a smaller set of variables. This motivation goes back to classical work on varimax rotations and more recent work on Ročková and George’s rotations. These sparsity considerations are important in many applications to help assign a meaning to the latent coordinates. Second, the sparsity assumption, when warranted, has significant potential gains for estimation accuracy. For illustration, the sparse scenarios in Table 1 show that MOM-SS attains low errors when estimating both the covariance and the expected value of the response variable, and requires significantly less factors than our competitors. Of course, if the sparsity assumption is not warranted then there is the risk of over-enforcing sparsity, but one may overcome this issue by carefully eliciting the prior parameters, as our results show. For instance, in our ovarian cancer example MOM-SS chose a sparse $M$ and achieved a concordance index (CI) similar to ComBat-MLE but with considerably less factors (4 instead of 101). On the other hand, in the lung and colon cancer settings, MOM-SS selected a non-sparse $M$ (74 and 53 factors respectively), achieving competitive CI for lung and the highest CI for colon cancer. Overall, MOM-SS provided a more stable performance than competing methods in balancing sparsity and prediction accuracy. Third, inducing sparsity allows one to work with large $q$ and let the data learn how many factors are needed. Namely, since in practice the value of $q$ is unknown, one might consider large $q$, in which case the dimension of $Z$ and $M$ could be substantial. Thus, considering sparse solutions acts as a model selection tool to learn the number of factors from data.

3 Prior formulation

To complete Model (2.2) we set priors for the loadings $M$, precisions $\tau_{jl}$, and regression parameters ($\theta, \beta$). Through our proposed default prior formulation we assume that the columns in $X$ have been centred to zero mean and unit variance. For the idiosyncratic precisions $\tau_{jl}$ we set

$$\tau_{jl} \mid \eta, \xi \sim \text{Gamma}(\eta/2, \eta \xi/2) \tag{3.1}$$

independently across $j = 1, \ldots, p$ and $l = 1, \ldots, p_b$. By default in our examples we set the fairly informative values $\eta = \xi = 1$, leading to diffuse though proper priors.

For the regression parameters we set

$$(\theta_j, \beta_j) \sim N(0, \psi I), \quad j = 1, \ldots, p, \tag{3.2}$$

where $\psi$ is a user-defined prior dispersion that in our examples by default we set to $\psi = 1$. The choice of $\psi = 1$ assigns the same marginal prior variances to elements in $(\theta_j, \beta_j)$ as the unit information prior often adopted as a default for linear regression (Schwarz, 1978).

We remark that this prior does not encourage sparsity in the regression parameters $(\theta, \beta)$ or factor loadings, which we view as reasonable provided the number of variables $p_v$ and batches $p_b$ are moderate. For large $p_v$ or $p_b$, a direct extension of our prior on the loadings $M$ could be adopted.
A. Avalos-Pacheco, D. Rossell, and R. S. Savage

The loadings matrix \( M \) plays an important role in improving shrinkage and simplifying interpretation. Some recent strategies include a LASSO-based method (Witten et al., 2009), horseshoe priors (Carvalho et al., 2009), an Indian buffet process (Knowles and Ghahramani, 2011), an infinite factor model (Dunson and Bhattacharya, 2011) among others. In this paper, we consider three priors on the loadings: an improper flat prior \( p(M) \propto 1 \), a Normal spike-and-slab and a novel non-local pMoM spike-and-slab.

The local and non-local spike-and-slab prior formulations are detailed below, along with Laplace-based extensions. These build on the approach by Roˇckov´a and George (2014, 2017), our main contribution being the introduction of non-local-based variations.

### 3.1 Local spike-and-slab prior

A traditional Bayesian approach to variable selection is the spike-and-slab prior, a two-component mixture prior (Mitchell and Beauchamp, 1988; George and McCulloch, 1993). This prior aims to discriminate those loadings that warrant inclusion, modelled by the slab component, from those that should be excluded, modelled by the spike component.

Specifically, a spike-and-slab prior density for the loadings \( M \) has the form

\[
p(M | \gamma, \lambda_0, \lambda_1) = \prod_{j=1}^{p} \prod_{k=1}^{q} (1 - \gamma_{jk}) p(m_{jk} | \lambda_0, \gamma_{jk} = 0) + \gamma_{jk} p(m_{jk} | \lambda_1, \gamma_{jk} = 1),
\]

(3.3)

where \( p(m_{jk} | \lambda_0, \gamma_{jk} = 0) \) is a continuous density, \( \lambda_0 \) is a given dispersion parameter of the spike component and \( \lambda_1 > \lambda_0 \) is that of the slab component. The indicators \( \gamma_{jk} \in \{0, 1\} \) signal which \( m_{jk} \) were generated by each component, and serve as a proxy for which loadings are significantly non-zero. We take as a base formulation the Normal-spike-and-slab prior by George and McCulloch (1993) where the spike is a Normal density with a small variance \( \lambda_0 \) and the slab a Normal distribution with large variance \( \lambda_1 \). Although Laplace-Spike-and-Slab priors have been shown to possess better properties for sparse inference (Roˇckov´a and George, 2018), as discussed below the introduction of non-local penalties improves certain undesirable features of the Normal-based prior. The elicitation of \( \lambda_0 \) and \( \lambda_1 \) is an important aspect of the formulation and will be discussed in Section 3.3. Specifically, the Normal-spike-and-slab is

\[
p(m_{jk} | \gamma_{jk} = l, \lambda_l) = N(m_{jk}; 0, \lambda_l).
\]

(3.4)

The continuity of the spike distribution gives closed form expressions for the EM algorithm, making it computationally appealing. We refer to (3.4) as Normal-SS.

We complete the model specification with a hierarchical prior over the latent indicator \( \gamma = \{\gamma_{jk}, j = 1, \ldots, p, k = 1, \ldots, q\} \) as follows,

\[
\gamma_{jk} | \zeta_k \sim \text{Bernoulli}(\zeta_k), \\
\zeta_k | a_\zeta, b_\zeta \sim \text{Beta}\left(\frac{a_\zeta}{k}, b_\zeta\right),
\]

(3.5)

with independence across \((j, k)\) where \( a_\zeta > 0 \) and \( b_\zeta > 0 \) are given prior parameters. By default we set \( a_\zeta = b_\zeta = 1 \), which leads to a uniform prior for the first factor \((k = 1)\),
ζ_k | a_c, b_c ~ U(0, 1). Furthermore, note that $\frac{a_c}{b_c}$ encourages increasingly sparse solutions in subsequent factors. That is, related to our earlier discussion of non-identifiability (Section 2), we encourage loadings where the first factors have larger importance, leading to solutions that are sparse both in the rank of $M$ and its non-zero entries.

A potential concern with Normal-SS is that the slab density assigns non-negligible probability to regions of the parameter space that are also consistent with the spike, namely when $m_{jk}$ lies close to zero. We will address this via non-local priors and show that these, by enforcing separation between two components, help increase sensitivity.

### 3.2 Non-local spike-and-slab prior

Non-local priors (NLPs) are a family of distributions that assign vanishing prior density to a neighbourhood of the null hypothesis (Johnson and Rossell, 2010). Definition 3.1 is an adaptation of the definition in Johnson and Rossell (2010) to (3.3).

**Definition 3.1.** An absolutely continuous measure with density $p(m_{jk} | \gamma_{jk} = 1)$ is a non-local prior if $\lim_{m_{jk} \to 0} p(m_{jk} | \gamma_{jk} = 1) = 0$.

We call any prior not satisfying Definition 3.1 a local prior. Non-local priors possess appealing properties for Bayesian model selection. They discard spurious parameters faster as the sample size $n$ grows, but preserve exponential rates to detect important coefficients (Johnson and Rossell, 2010; Fúquene et al., 2018) and can lead to improved parameter estimation shrinkage (Rossell and Telesca, 2017). To illustrate the motivation for NLPs in our setting consider Figure 1. Normal-SS assigns positive probability to $m_{jk} = 0$. Correspondingly, the conditional inclusion probability $p(\gamma_{jk} = 1 | m_{jk})$ remains non-negligible, even when $m_{jk} = 0$ (lower left panel).

As an alternative, we consider a product moment (pMOM) prior (Johnson and Rossell, 2012).  

$$p(m_{jk} | \gamma_{jk} = 0, \lambda_0) = N(m_{jk}; 0, \lambda_0),$$

$$p(m_{jk} | \gamma_{jk} = 1, \lambda_1) = \frac{m_{jk}^2}{\lambda_1} N(m_{jk}; 0, \lambda_1).$$

We denote (3.6) as MOM-SS. This prior assigns zero density to $m_{jk} = 0$ given $\gamma_{jk} = 1$, which implies $p(\gamma_{jk} = 1 | m_{jk} = 0) = 0$ (Figure 1). Prior elicitation for $\lambda_0$ and $\lambda_1$ is discussed in Section 3.3. From a computational point of view, the EM algorithm can accommodate this extension by using a trivial extra gradient evaluation at negligible additional cost relative to the Normal-SS. Parameter estimation and algebraic details are described in Section 4. The prior on the inclusion indicators is set as in (3.5).

Beyond (3.4) and (3.6), another natural extension is to use Laplace-based priors based on the Spike-and-Slab LASSO by Ročková and George (2018)  

$$p(m_{jk} | \gamma_{jk}, \lambda_0, \lambda_1) = (1 - \gamma_{jk}) \text{Laplace}(m_{jk}; 0, \lambda_0) + \gamma_{jk} \text{Laplace}(m_{jk}; 0, \lambda_1),$$

with a slab component with variance $2\lambda_0^2$, and a spike component with $2\lambda_1^2$, where $\text{Laplace}(m_{jk}; 0, \lambda) = \frac{1}{2\lambda} \exp \left( \frac{-|m_{jk}|}{\lambda} \right)$. We refer to (3.7) as Laplace-SS. As illustrated in
Figure 1 (right panels) this prior can help encourage sparsity, setting \( \Pr(\gamma_{jk} = 1 \mid m_{jk} = 0) \) to smaller values (though still non-zero) than the Normal-SS.

As an extension, akin to (3.6), one could set a moment penalty on the Laplace density.

\[
\begin{align*}
\Pr(m_{jk} \mid \gamma_{jk} = 0, \lambda_0) &= \text{Laplace}(m_{jk}; 0, \lambda_0), \\
\Pr(m_{jk} \mid \gamma_{jk} = 1, \lambda_1) &= \frac{m_{jk}^2}{2\lambda_1} \text{Laplace}(m_{jk}; 0, \lambda_1).
\end{align*}
\]  

(3.8)

We denote (3.8) as Laplace-MOM-SS. Relative to (3.6), as illustrated in Figure 1, Laplace-MOM-SS leads to lower \( \Pr(\gamma_{jk} = 1 \mid m_{jk} = 0) \) and higher \( \Pr(\gamma_{jk} = 1 \mid m_{jk}) \) for moderately large \( m_{jk} \).

We discuss prior elicitation for Laplace-MOM-SS in Section 3.3 and derive an EM algorithm in Section 4.2 but in our examples we focus on the MOM-SS for simplicity. However, the Laplace-based (3.8) can also be shown to lead to closed-form EM updates.

### 3.3 Prior elicitation for the variance of the spike-and-slab priors

A crucial aspect in a spike-and-slab prior is the choice of the prior scale parameters. It is common to fix the variance of the spike distribution \( \lambda_0 \) to a value close to zero. Regarding \( \lambda_1 \), one option is to set a hyper-prior or to try to estimate it from the data (George and McCulloch, 1993, 1997; Rocková and George, 2014, 2018). Setting a hyper-prior does not bypass prior elicitation, as one then needs to set the hyper-prior parameters, whereas estimating \( \lambda_1 \) from the data increases the cost of computations. Instead, we capitalise on the fact that factor loadings have a natural interpretation in terms of the fraction of explained variance in \( X \). Thus, we propose default values that dictate which coefficients are considered as meaningfully different from zero. These defaults are guidelines in the absence of a priori knowledge. A convenient feature of such an elicitation is that it can be easily extended to local priors and other non-Gaussian spike-and-slab priors.

Our goal is to find values \( \tilde{\lambda}_0 \) and \( \tilde{\lambda}_1 \) for the MOM-SS that distinguish practically relevant factors. In the absence of covariates, the factor model decomposes the total variance in variable \( j \) as \( \text{Var}(x_{ij}) = \sum_{k=1}^{q} m_{jk}^2 + \tau_{jj}^{-1} \), hence \( m_{jk}^2 \) is the proportion of variance in variable \( j \) explained by factor \( k \). We take \( m_{jk}^2 > 0.1 \) as a threshold for practical relevance. Specifically, we set \( \tilde{\lambda}_0 \) such that \( \Pr(|m_{jk}| \leq \sqrt{0.1} \mid \lambda_0) = 0.95 \), that is \( \lambda_0 = \Phi^{-1}(0.025)^2 \approx 0.026 \), where \( \Phi^{-1} \) denotes the standard normal quantile function. Likewise we set \( \Pr(|m_{jk}| \geq \sqrt{0.1} \mid \lambda_1) = 0.95 \) under the MOM-SS, obtaining the default \( \lambda_1 \approx 0.2842 \).

Regarding the Normal-SS prior, we set \( \lambda_0 = \tilde{\lambda}_0 \) and \( \lambda_1 \) such that it is comparable to the MOM-SS in terms of informativeness, namely it matches the variance of the MOM-SS, obtaining that \( \lambda_1 = 3\lambda_1 \approx 0.8526 \).
In Laplace-MOM-SS, we analogously set $\tilde{\lambda}_0 = -\frac{\sqrt{0.1}}{\log(0.005)} \approx 0.1056$ so that $p(|m_{jk}| \leq \sqrt{0.1} | \tilde{\lambda}_0)$ and $\tilde{\lambda}_1 \approx 0.3867$ such that $p(|m_{jk}| \geq \sqrt{0.1} | \tilde{\lambda}_1) = 0.95$ for the Laplace-spike-and-MOM-slab prior. Finally for the Laplace-SS we set $\lambda_1 = \sqrt{6}\tilde{\lambda}_1 \approx 0.9473$ and $\lambda_0 = \tilde{\lambda}_0$ for the spike and slab component, respectively, matching the variances of the non-local Laplace-based priors.

The resulting priors are in Figure 1. We remark that a considerable difference can be observed between the local prior based and the non-local prior based formulations, particularly in the conditional inclusion probability around $m_{jk} = 0$. In our examples we will focus on the Normal MOM-SS. Deeper analysis of Laplace-based non-local priors, whose thicker tails might help improve estimation accuracy, is left for interesting future work.

Figure 1: Prior comparison (top panels) for $m_{jk}$ under different prior specifications and its inclusion probabilities $p(\gamma_{jk} \mid m_{jk})$ (bottom panels). Comparison between Normal-based (left) and Laplace-based (right) priors. Scales ($\lambda_0, \lambda_1$) are set to the defaults from Section 3.3.
Parameter estimation

Parameter estimation in factor analysis is usually conducted using Expectation-Maximisation (EM, Dempster et al. (1977)), MCMC algorithms (Lopes and West, 2004) or approximated via variational inference (Ghahramani and Beal, 2000). At the core of these algorithms is the fact that, conditional on the data and all other model parameters, we can set $\tilde{x}_i = x_i - \theta v_i - \beta b_i$ and express the model in (2.2) as a linear regression $\tilde{x}_i = Mz_i + e_i$, where $M$ and $\Sigma$ are fixed at their current values of each MCMC iteration or maximisation step (West, 2003; Carvalho et al., 2008). We develop a deterministic optimisation along the lines of the EM algorithm of Rockova and George (2017). Section 4.1 provides two EM algorithms to obtain posterior modes for our factor regression with batch effect correction with and without sparse formulation. Section 4.2 outlines algorithms separately for Normal-SS, MOM-SS, Laplace-SS and Laplace-MOM-SS priors. Section 4.3 discusses parameter initialisation and Section 4.4 how to post-process the fitted model to obtain sparse solutions and variance-adjusted dimensionality reduction.

4.1 EM algorithm under a uniform prior

We outline an EM algorithm to fit Model (2.2) under a uniform prior $p(M) \propto 1$ on the loadings via maximum a posteriori (MAP) estimation. The algorithm maximises the log-posterior by treating the latent factors $Z$ as missing data and setting them to their expectation (conditional on all other parameters) in the E-step. Then, the remaining parameters $\Delta = (M, \theta, \beta, T)$ are optimised in the M-step. In other words, the EM algorithm obtains a local mode of the log-posterior $p(M, \theta, \beta, T | X)$ by maximising the expected complete-data log-posterior $p(M, \theta, \beta, T | X, Z)$ iteratively. For convenience we denote by $T_b^l$ the idiosyncratic precision matrix in batch $l$, i.e. if $b_i^l = 1$ by $\tau_j^l$, then the errors are distributed as $e_i \sim N(0, T_b^l)$. We also denote with $\hat{\Delta} = (\hat{M}, \hat{\theta}, \hat{\beta}, \hat{T})$ the current value of the parameters. We briefly describe the algorithm; see Supplementary Avalos-Pacheco et al. (2020) Section 4 for its full derivation.

The E-step takes the expectation of $\log p(M, \theta, \beta, T | X, Z)$ with respect to $p(Z | \hat{\Delta}, X)$. Specifically, let

$$Q(\Delta) = \mathbb{E}_{z_i | \hat{\Delta}, X} [\log p(M, \theta, \beta, T | X, Z)]$$

$$= C - \frac{1}{2} \sum_{i=1}^{n} [(x_i - \theta v_i - \beta b_i)^T T_b (x_i - \theta v_i - \beta b_i)]$$

$$- 2(x_i - \theta v_i - \beta b_i)^T T_b M \mathbb{E}[z_i | \hat{\Delta}, X] + \text{tr} \left( M^T T_b M \mathbb{E}[z_i z_i^T | \hat{\Delta}, X] \right) + p$$

$$+ \sum_{i=1}^{p} \frac{n_i + \eta - 2}{2} \log | T_i | - \sum_{i=1}^{p} \frac{\eta_i}{2} \text{tr}(T_i) - \frac{1}{2} \sum_{j=1}^{p} (\theta_j^T, \beta_j^T) - I(\theta_j, \beta_j),$$

where $C$ is a constant. Expression (4.1) only depends on $Z$ through the conditional posterior mean

$$\mathbb{E}[z_i | \hat{\Delta}, X] = (I_q + M^T T_b M)^{-1} M^T T_b (x_i - \theta v_i - \beta b_i)$$

(4.2)
and the conditional second moments
\[ E[z_i z_i^\top | \hat{\Delta}, X] = (I_q + \hat{\hat{M}}^\top \hat{T}_b \hat{M})^{-1} + E[z_i | \hat{\Delta}, X] E[z_i | \hat{\Delta}, X]^\top, \]  
(4.3)
where \((I_q + \hat{\hat{M}}^\top \hat{T}_b \hat{M})^{-1} = \text{Cov}[z_i | \hat{\Delta}, X]\) is the conditional covariance matrix of the latent factors. We emphasise that (4.2) and (4.3) depend on batch-specific precisions \(T_b\).

The M-step maximises \(Q(\Delta)\) with respect to \(M, \theta, \beta, T\). Setting its partial derivatives to 0 gives the updates
\[
\hat{m}_j = \left[ \sum_{i=1}^{n} (\hat{r}_j^\top b_i E[z_i | \hat{\Delta}, X]) \right] \left[ \sum_{i=1}^{n} (\hat{r}_j^\top b_i E[z_i z_i^\top | \hat{\Delta}, X]) \right]^{-1},
\]
(4.4)
\[
\hat{T}_i^{-1} = \frac{1}{n_l + \eta} \text{diag}\left\{ \sum_{i: b_{ii}=1} (x_{ij} - \hat{\theta} v_i - \hat{\beta} b_{ij}) (x_{ij} - \hat{\theta} v_i - \hat{\beta} b_{ij})^\top \right\} + \eta \xi I_p,
\]
(4.5)
where \(\hat{x}_i = x_i - \hat{\theta} v_i - \hat{\beta} b_i\) and \(\hat{x}_{ij} = x_{ij} - \hat{\theta} v_{ij} - \hat{\beta} b_{ij}\).

The updates for \((\theta_j, \beta_j)\) are
\[
(\hat{\theta}_j^\top, \hat{\beta}_j^\top) = \sum_{i=1}^{n} \left[ \hat{r}_j^\top b_i (x_{ij} - \hat{m}_j^\top E[z_i | \hat{\Delta}, X]) (v_i, b_i)^\top \right] \times \left[ \sum_{i=1}^{n} \left[ \hat{r}_j^\top b_i (v_i, b_i)^\top \right] + \frac{1}{\psi} I \right]^{-1}.
\]
(4.6)
Equation (4.6) has the form of a ridge regression estimator with penalty \(\psi\).

Algorithm 1 summarises the EM algorithm. The stopping criteria is reaching a tolerance \(\epsilon^*\) in the log-posterior change, a maximum number of iterations \(T\) or a change \(\epsilon^*_M\) on the loadings. By default we set \(\epsilon^* = 0.001\), \(T = 100\) and \(\epsilon^*_M = 0.05\). Parameter initialisation is an important aspect that helps obtain better local modes and reduce computational time; its discussion is deferred to Section 4.3.

### 4.2 EM algorithm for spike-and-slab priors

The algorithm is derived analogously to Section 4.1. The expected complete-data log-posterior can be split into \(Q(\Delta) = C + Q_1(\theta, M, \beta, T) + Q_2(\zeta)\), where
\[
Q_1(\theta, M, \beta, T) = -\frac{1}{2} \sum_{i=1}^{n} \left[ (x_i - \theta v_i - \beta b_i)^\top T_b (x_i - \theta v_i - \beta b_i) \right. \\
- 2(x_i - \theta v_i - \beta b_i)^\top T_b M E[z_i | \hat{\Delta}, X]
\]
Algorithm 1: EM algorithm for factor regression model with uniform p(M).

initialise $\hat{M} = M^{(0)}$, $\hat{\theta} = \theta^{(0)}$, $\hat{\beta} = \beta^{(0)}$, $\bar{T}_b = \mathcal{T}_b^{(0)}$

while $\epsilon > \epsilon^*$, $\epsilon_M > \epsilon_M^*$ and $t < T$ do

E-step:
  Latent factors: $\mathbb{E}[z_i | \hat{\Delta}, X] = (I_q + \hat{M}_i^T \bar{T}_b M)^{-1} \hat{M}_i^T \bar{T}_b (x_i - \hat{\theta} v_i - \hat{\beta} b_i)$

M-step:
  Loadings: $\hat{m}_{ij} = \left[ \sum_{t=1}^{T} (\hat{\tau}_j b_i \hat{\gamma}_t \mathbb{E}[z_i^T | \Delta, X]) \right]^{-1} \mathbb{E}_t \left[ \sum_{t=1}^{T} (\hat{\tau}_j b_i \mathbb{E}[z_i^T | \Delta, X]) \right]$
  Variances: $\hat{\tau}_j^{-1} = \frac{p_n + n - 2}{2} \log | \mathcal{T}_j | - \sum_{t=1}^{p_n} \frac{\eta_t}{2} \text{tr}(\mathcal{T}_j)$
  Coefficients: $(\hat{\theta}_j, \hat{\beta}_j) = \sum_{t=1}^{T} \mathbb{E}[\gamma_t | \mathcal{T}_j, \Delta, X] (\tau_j b_i)^T \mathbb{E}[z_i^T | \mathcal{T}_j, \Delta, X] (\tau_j b_i)^T$

set $\Delta(t+1) = \hat{\Delta}$ and $M(t+1) = \hat{M}$

compute $\epsilon = Q(\Delta(t+1)) - Q(\Delta^t)$, $\epsilon_M = \max |m_{jk}^{(t+1)} - m_{jk}^{(t)}|$ and $t = t + 1$

end

$$+ \text{tr} \left( \hat{M}_{ii}^T \bar{T}_b M \mathbb{E}[z_i z_i^T | \hat{\Delta}, X] \right)$$

$$+ \sum_{j=1}^{p} \frac{n_j + q - 2}{2} \log | \mathcal{T}_j | - \sum_{j=1}^{p} \frac{\eta_j}{2} \text{tr}(\mathcal{T}_j)$$

$$- \frac{1}{2} \sum_{j=1}^{p} (\hat{\theta}_j, \hat{\beta}_j)^T \frac{1}{\psi} I(\theta^j, \beta_j) + \sum_{j=1}^{p} \sum_{k=1}^{q} \mathbb{E}_t [\hat{\gamma}_j] \log p(m_{jk} | \gamma_j, \lambda_0, \lambda_1)\right),$$

$$Q_2(\zeta) = \sum_{j=1}^{p} \sum_{k=1}^{q} \log \left( \frac{\zeta_k}{1 - \zeta_k} \right) \mathbb{E}[\gamma_j k | \hat{\Delta}]$$

$$+ \sum_{k=1}^{q} \left( \frac{\zeta_k}{1 - \zeta_k} \right) \log (1 - \zeta_k) + (p + b - 1) \log (1 - \zeta_k) \right),$$

with C a constant and $\mathbb{E}[z_i | \hat{\Delta}, X]$ and $\mathbb{E}[z_i z_i^T | \hat{\Delta}, X]$ as in (4.2) and (4.3).

$Q_1(\theta, M, \beta, \mathcal{T})$ resembles the E-step for the flat prior in Section 4.1, plus an extra conditional expectation $\mathbb{E}_t [\hat{\gamma}_j k | \Delta] \log p(m_{jk} | \gamma_j, \lambda_0, \lambda_1)$. $Q_2(\zeta)$ arises from the Beta-Binomial prior on $\gamma_j k$ and the $\mathbb{E}[\gamma_j k | \cdot]$ are straightforward to compute. In the M-step we maximise $Q_1$ w.r.t. $(\theta, M, \beta, \mathcal{T})$, this can be done in a completely independent fashion from optimising $Q_2$ w.r.t. $\zeta$.

Further the conditional expectation of $\mathbb{E}[\gamma_j k | \hat{\Delta}] = \hat{p}_{jk}$ is

$$\hat{p}_{jk} = \frac{p(\hat{m}_{jk} | \gamma_j = 1, \lambda_0, \lambda_1)p(\gamma_j k = 1)}{p(\hat{m}_{jk} | \gamma_j = 0, \lambda_0, \lambda_1)p(\gamma_j k = 1) + p(\hat{m}_{jk} | \gamma_j = 1, \lambda_0, \lambda_1)p(\gamma_j k = 1)}.$$  

For the Normal-SS prior, (4.9) is

$$\hat{p}_{jk} = \left[ 1 + \sqrt{\frac{\lambda_1}{\lambda_0}} \exp \left( -\frac{1}{2} \hat{m}_{jk}^2 \left( \frac{1}{\lambda_0} - \frac{1}{\lambda_1} \right) \right) \frac{1 - \mathbb{E}[\zeta]}{\mathbb{E}[\zeta]} \right]^{-1},$$  

(4.10)
for the MOM-SS
\[
\hat{p}_{jk} = \left[ 1 + \frac{\hat{\lambda}_1}{\hat{m}_{jk}} \sqrt{\frac{\hat{\lambda}_1}{\hat{\lambda}_0}} \exp \left( -\frac{1}{2} \hat{m}_{jk}^2 \left( \frac{1}{\hat{\lambda}_0} - \frac{1}{\hat{\lambda}_1} \right) \right) \frac{1 - \mathbb{E}[\zeta_j]}{\mathbb{E}[\zeta_j]} \right]^{-1},
\]
(4.11)
for the Laplace-SS
\[
\hat{p}_{jk} = \left[ 1 + \frac{\hat{\lambda}_1}{\hat{\lambda}_0} \exp \left( -|\hat{m}_{jk}| \left( \frac{1}{\hat{\lambda}_0} - \frac{1}{\hat{\lambda}_1} \right) \right) \frac{1 - \mathbb{E}[\zeta_j]}{\mathbb{E}[\zeta_j]} \right]^{-1},
\]
(4.12)
and for the Laplace-MOM-SS
\[
\hat{p}_{jk} = \left[ 1 + \frac{2\hat{\lambda}_1^2}{\hat{m}_{jk}} \frac{\hat{\lambda}_1}{\hat{\lambda}_0} \exp \left( -|\hat{m}_{jk}| \left( \frac{1}{\hat{\lambda}_0} - \frac{1}{\hat{\lambda}_1} \right) \right) \frac{1 - \mathbb{E}[\zeta_j]}{\mathbb{E}[\zeta_j]} \right]^{-1}.
\]
(4.13)

Equations (4.10) and (4.12) are analogous to the EM posterior update for \( m_{jk} \) in a two-component Gaussian or Laplace mixture (Ročková and George, 2014). Equations (4.11) and (4.13) are similar to their local counterparts, but incorporate a penalty for small \( m_{jk}^2 \).

The main difference between the local and non-local priors lies in updating the loadings and the idiosyncratic variances. We discuss these separately for each prior later in this section.

The updates for the precision \( T_l \) and the regression parameters \((\theta, \beta)\) are given in (4.5) and (4.6) respectively.

Maximising \( Q_2(\zeta) \) with respect to \( \zeta_k \) gives
\[
\hat{\zeta}_k = \frac{\sum_{j=1}^p \hat{p}_{jk} + a_\zeta - 1}{ \frac{a_\zeta}{k} + b_\zeta + p - 1 },
\]
(4.14)
for \( k = 1, \ldots, q \).

Algorithm 2 summarises the algorithm. It is initialised with the two-stage least-squares method described in Section 4.3 and \( \zeta_k = 0.5 \) for \( k = 1, \ldots, q \). The stopping criteria are as in Algorithm 1. The different updates for \( M \) are outlined below, separately for each prior specification.

Let \( d_{jk} = [(1 - \gamma_{jk})\lambda_0 + \gamma_{jk}\lambda_1]^{-1} \). In Expression (4.7), under a Normal-SS prior
\[
\mathbb{E}_{\gamma|\Delta} \left[ \log p(m_{jk} \mid \gamma_{jk}, \lambda_0, \lambda_1) \right] \propto -\frac{1}{2} \hat{m}_{jk}^2 \mathbb{E} \left[ d_{jk} \mid \hat{\Delta} \right] = -\frac{1}{2} \hat{m}_{jk}^2 \left[ \frac{1 - \hat{p}_{jk}}{\lambda_0} + \frac{\hat{p}_{jk}}{\lambda_1} \right],
\]
(4.15)
where \( \hat{p}_{jk} \) is as in (4.10).
Algorithm 2: EM algorithm for factor regression model with spike-and-slab $p(M)$.

initialise $\hat{M} = M^{(0)}$, $\hat{\theta} = \theta^{(0)}$, $\hat{\beta} = \beta^{(0)}$, $\hat{T}_b = T_b^{(0)}$, $\hat{\zeta} = \zeta^{(0)}$

while $\epsilon > \epsilon^*$, $\epsilon_M > \epsilon_M^*$ and $t < T$ do

E-step:
- Latent factors: $E[z_i | \Delta, X] = (I_n + \hat{M}^T \hat{T}_n \hat{M})^{-1} \hat{M}^T \hat{T}_n (x_i - \hat{\theta} v_i - \hat{\beta} b_i)$
- Latent indicators$^+$: $E[\gamma_{jk} | \Delta] = \hat{\gamma}_{jk}$

M-step:
- Loadings$: \hat{m}_{jk} = \arg \max_{m_{jk}} Q_1(\hat{\Delta})$
- Variances: $\tau_{jk}^{-1} = \frac{\sum_{i=1}^n \| \hat{y}_i \|^2}{\sum_{i=1}^n \| \hat{y}_i \|^2 + \hat{m}_{jk}^2 + \hat{\gamma}_{jk}^2}$
- Coefficients: $(\hat{\theta}_{jk}^T, \hat{\beta}_{jk}^T) = \frac{\sum_{i=1}^n \| \hat{y}_i \|^2}{\sum_{i=1}^n \| \hat{y}_i \|^2 + \hat{m}_{jk}^2 + \hat{\gamma}_{jk}^2} (\hat{\theta}_{jk}^T, \hat{\beta}_{jk}^T)$
- Weights: $\hat{\zeta}_k = \frac{\sum_{i=1}^n \hat{z}_{ik} \hat{\gamma}_{jk}}{\sum_{i=1}^n \hat{z}_{ik}^2}$

set $\Delta^{(t+1)} = \hat{\Delta}$ and $M^{(t+1)} = \hat{M}$

compute $\epsilon = Q(\Delta^{t+1}) - Q(\Delta^t)$, $\epsilon_M = \max |m_{jk}^{(t+1)} - m_{jk}^{(t)}|$ and $t = t + 1$

end

$^+$ see Section 4.2, Supplementary Avalos-Pacheco et al. (2020) Sections 5 and 6 for details.

Thus, the EM update for the $j^{th}$ row of matrix $M$ is,

$$\hat{m}_{jk} = \sum_{i=1}^n \left( \hat{x}_{ij}^T b_i \hat{\gamma}_{jk} | \hat{\Delta}, X \right) \times \left[ \text{diag}\left( E[d_{j1} | \hat{\Delta}], \ldots, E[d_{jq} | \hat{\Delta}] \right) + \sum_{i=1}^n \left( \hat{x}_{ij}^T b_i E[z_i^T | \hat{\Delta}, X] \right) \right]^{-1}, \quad (4.16)$$

for $j = 1, \ldots, p$, where $\hat{x}_{ij} = x_{ij} - \hat{\theta} v_{ij} - \hat{\beta} b_{ij}$. A full derivation is given in Supplementary Avalos-Pacheco et al. (2020) Section 5.

For the MOM-SS

$$E_{\gamma | \Delta} \left[ \log p(m_{jk} | \gamma_{jk}, \lambda_0, \lambda_1) \right] \propto -\frac{1}{2} m_{jk}^2 \left[ \frac{1}{\hat{\lambda}_0} + \frac{\hat{\gamma}_{jk}}{\hat{\lambda}_1} \right] + \hat{\gamma}_{jk} \log(m_{jk}^2), \quad (4.17)$$

where $\hat{\gamma}_{jk}$ is given in (4.11).

For the M-step, we use a coordinate descent algorithm (CDA) that performs successive univariate optimisation on (4.7) with respect to each $m_{jk}$. An advantage is that the updates have a closed-form that is computationally inexpensive. As a potential drawback it could require a larger number of iterations to converge relative to performing joint optimisation with respect to multiple elements in $M$. However, we have not found this to be a practical problem in our examples.

Viewed as a function of only $m_{jk}$, it is possible to express $Q_1(m_{jk})$ as

$$Q_1(m_{jk}) = am_{jk}^2 + bm_{jk} + c \log(m_{jk}^2), \quad (4.18)$$
where
\[ a = -\frac{1}{2}\left(\frac{1 - \hat{p}_{jk}}{\lambda_0} + \frac{\hat{p}_{jk}}{\lambda_1}\right) + \sum_{i=1}^{n} \hat{\tau}_{ij}^T b_i E[z_{ik} z_{ik}^T | \hat{\Delta}, X] , \]
\[ b = \sum_{i=1}^{n} \left[ \hat{\tau}_{ij}^T b_i (x_{ij} - \hat{\theta} v_{ij} - \hat{\beta} b_{ij}) E[z_{ik} | \hat{\Delta}, X] - \sum_{r \neq k} \hat{m}_{jr} \hat{\tau}_{jr}^T b_i E[z_{ir} z_{ik}^T | \hat{\Delta}, X] \right] , \]
\[ c = \hat{p}_{jk} . \]

See Supplementary Avalos-Pacheco et al. (2020) Section 6. The global maximum of (4.18) is summarised in Lemma 1.

**Lemma 1.** Let \( f(m_{jk}) = am_{jk}^2 + bm_{jk} + c \log(m_{jk}^2) \), where \( a < 0 \) and \( c > 0 \). Define \( \overline{m}_{jk} = \frac{-b - \sqrt{b^2 - 16ac}}{4a} \) and \( \bar{m}_{jk} = \frac{-b + \sqrt{b^2 - 16ac}}{4a} \).

If \( b > 0 \), then \( \overline{m}_{jk} = \arg \max m_{jk} f(m_{jk}) \). If \( b < 0 \), then \( \bar{m}_{jk} = \arg \max m_{jk} f(m_{jk}) \).

Akin to the MOM-SS, we can express \( Q_1(m_{jk}) \) as function of \( m_{jk} \) for the Laplace-based priors as:
\[ Q_1(m_{jk}) = am_{jk}^2 + bm_{jk} + c|m_{jk}| + d \log(m_{jk}^2) , \]
\[ a = -\frac{1}{2} \sum_{i=1}^{n} \hat{\tau}_{ij}^T b_i E[z_{ik} z_{ik}^T | \hat{\Delta}, X] , \]
\[ b = \sum_{i=1}^{n} \left[ \hat{\tau}_{ij}^T b_i (x_{ij} - \hat{\theta} v_{ij} - \hat{\beta} b_{ij}) E[z_{ik} | \hat{\Delta}, X] - \sum_{r \neq k} \hat{m}_{jr} \hat{\tau}_{jr}^T b_i E[z_{ir} z_{ik}^T | \hat{\Delta}, X] \right] , \]
\[ c = -\left[ \frac{1 - \hat{p}_{jk}}{\lambda_0} + \frac{\hat{p}_{jk}}{\lambda_1}\right] , \]
\[ d = \begin{cases} 0 & \text{for Laplace-SS} \\ \hat{p}_{jk} & \text{for Laplace-MOM-SS} \end{cases} \]

for \( j = 1, \ldots, p \) and where \( \hat{p}_{jk} \) is as in (4.12) and (4.13) for Laplace-SS and Laplace-MOM-SS respectively.

Lemma 2 summarises the global maximum for Laplace-SS.

**Lemma 2.** Let \( f(m_{jk}) = am_{jk}^2 + bm_{jk} + c|m_{jk}| \), where \( a < 0 \) and \( c < 0 \). Define \( m_{jk}^+ = \frac{-b + c}{2a} \) and \( m_{jk}^- = \frac{-b - c}{2a} \).

If \( b > -c \), then \( m_{jk}^+ = \arg \max m_{jk} f(m_{jk}) \). If \( b < c \), then \( m_{jk}^- = \arg \max m_{jk} f(m_{jk}) \).

Finally for the Laplace-MOM-SS, we emphasise that when \( m_{jk} = 0 \), \( Q_1(m_{jk} = 0) = -\infty \). Thus the solution for \( m_{jk} \) is given by setting \( \frac{\partial Q_1}{\partial m_{jk}} = 0 \) as given in Lemma 3.
Lemma 3. Let \( f(m_{jk}) = am_{jk}^2 + bm_{jk} + c|m_{jk}| + d \log(m_{jk}^2) \), where \( a < 0 \), \( c < 0 \) and \( d > 0 \). Define \( m_{jk}^+ = \frac{-(b+c)-\sqrt{(b+c)^2-16ad}}{4a} \) and \( m_{jk}^- = \frac{-(b+c)+\sqrt{(b+c)^2-16ad}}{4a} \).

If \( b > 0 \), then \( m_{jk}^+ = \text{arg max}_{m_{jk}} f(m_{jk}) \). If \( b < 0 \), then \( m_{jk}^- = \text{arg max}_{m_{jk}} f(m_{jk}) \).

If \( b = 0 \), then \( m_{jk}^+ = m_{jk}^- = \text{arg max}_{m_{jk}} f(m_{jk}) \).

We remark that if either \( x_i \) or \( v_i \) are continuous, the event of \( b = 0 \) has zero probability. If both \( x_i \) and \( v_i \) are discrete and in presence of the rare event of \( b = 0 \), then the sign of the update for \( m_{jk} \) is set to the previous one.

4.3 Initialisation of parameters

The EM algorithm can be sensitive to parameter initialisation. We propose two different strategies: least-squares and least-squares with rotation.

The first option is a simple two-step least-squares that is computationally efficient and performs well in many of our examples.

Step 1: initialise \((\theta^{(0)}, \beta^{(0)}) = [(V,B)^\top (V,B)]^{-1}(V,B)^\top X\).

Step 2: Let \( \hat{E} = X - (V\theta^{(0)} + B\beta^{(0)}) \). Consider the eigendecomposition of \( \frac{1}{n}\hat{E}^\top \hat{E} \) where \( l_1 \geq l_2 \geq \cdots \geq l_q \) are the eigenvalues and \( u_1, \ldots, u_q \) the eigenvectors. Set \( M^{(0)} = [\sqrt{l_1}u_1 | \cdots | \sqrt{l_q}u_q] \) and \( T_l^{(0)} = [\text{diag}(\frac{1}{n}\hat{E}^\top \hat{E} - M^{(0)}M^{(0)\top})]^{-1} \) for \( l = 1, \ldots, p_b \).

The rotated least-squares adds an extra step.

Step 3: varimax rotation for the loadings obtained in Step 2.

The reason for this extra step is to help escape local modes. The EM algorithm does not guarantee convergence to a global maximum, but it increases the log-posterior at each iteration. This local maxima issue is intensified by the non-identifiability of the factor model through the rotational ambiguity of the likelihood and the strong association between the updates of loadings and factors.

4.4 Post-processing for model selection and dimensionality reduction

The EM algorithm gives point estimates \((\hat{M}, \hat{\theta}, \hat{T}, \hat{\zeta})\). Under Laplace-SS one can obtain exact sparsity via \( \hat{m}_{jk} = 0 \), however this is not the case for our other priors. To address this, we define \( \hat{\gamma} \) as the solution of the following optimisation problem

\[
\hat{\gamma} = \text{arg max}_\gamma p(\gamma \mid X, \hat{M}, \hat{\theta}, \hat{T}, \hat{\zeta}) = \text{arg max}_\gamma \prod_{jk} p(\gamma_{jk} \mid \hat{m}_{jk}, \hat{\zeta}_k), \tag{4.21}
\]

where the right-hand side follows from the assumed conditional independence of \( m_{jk} \). That is, we set \( \hat{\gamma}_{jk} = 1 \) if \( p(\gamma_{jk} = 1 \mid \hat{m}_{jk}, \hat{\zeta}_k) > 0.5 \) and \( \gamma_{jk} = 0 \) otherwise. When \( \hat{\gamma}_{jk} = 0 \) we set \( \hat{m}_{jk} = 0 \) effectively selecting the number of factors and the non-zero loadings within each factor.
As an alternative post-processing step we consider that in some applications one may want to select only the number of factors. We then consider to setting $\hat{\gamma}_{jk} = 1$ if $\sum_{j=1}^{p} \hat{\gamma}_{jk} \neq 0$ and $\hat{\gamma}_{jk} = 0$ otherwise.

The combination of the two initialisation alternatives and two different post-processing options gives four possible solutions for $\hat{M}$. To choose which is best in our examples, we use weighted 10-fold cross-validation, where the weights reflect that batches with higher variance should receive lower weight, selecting the model with smallest weighted cross validation reconstruction error (see Supplementary Avalos-Pacheco et al. (2020) Section 9 for details).

Finally we re-order the factors so that $\sum_{j=1}^{p} \gamma_{jk}$ is decreasing in $k$, which under our prior (3.5) is guaranteed to increase the log-posterior. This is the so-called left-ordered inclusion matrix of Griffiths and Ghahramani (2011). This facilitates the interpretation of latent factors.

Latent factors are also post-processed for data visualisation purposes. The aim of this is to obtain new standardised factors $\tilde{z}_{i} = \left[ \text{Cov}(z_{i} | \hat{\Delta}, X) \right]^{-1} E[z_{i} | \hat{\Delta}, X]$, with $\text{Cov}(z_{i} | \Delta, X) = (I_{q} + \hat{M}^{T} \hat{M})^{-1}$, whose covariance does not depend on their batch.

## 5 Results

We assess our approach on simulated and experimental datasets. In Section 5.1 we evaluate the accuracy of our prior in obtaining sparse factor loadings and in estimating the covariance and low-dimensional representations, by comparing its performance to other methods in a setting where there are no batch effects. In Section 5.2 we show the importance of accounting for batch effects in simulated data. In Section 5.3 we do the same analysis for three cancer datasets, assessing the ability of our dimension reduction to predict survival outcomes.

Sections 5.1 and 5.2 study simulations under two different loading matrices $M$ (truly sparse and dense) and two different scenarios (without and with batch effects). We compare our methods with the Fast Bayesian Factor Analysis via Automatic Rotations to Sparsity (FastBFA) of Ročková and George (2017) and the Penalized Likelihood Factor Analysis with a LASSO penalty (LASSO-BIC) of Hirose and Yamamoto (2015). We also use the ComBat empirical Bayes batch effect correction of Johnson et al. (2007) for scenarios with batch effects, doing an MLE estimation of the factor analysis model (ComBat-MLE). In Section 5.3 we analyse a high-dimensional gene expression data under a supervised and an unsupervised framework. We use the clinically annotated data for the ovarian cancer transcriptome from R package curatedOvarianData 1.16.0 (Ganzfried et al., 2013), the lung cancer data from The Cancer Genome Atlas (TCGA) from R package TCGA2STAT 1.2 (Wan et al., 2015) and gene expression data from the colon cancer datasets of Culon et al. (2012).

The R package for our model is available at [https://github.com/AleAviP/BFR.BE](https://github.com/AleAviP/BFR.BE). The software sets all tuning parameters to the default values suggested in this paper. We used R function FACTOR_ROTATE of Ročková and George (2017) for FastBFA, the R package fanc 2.2 for LASSO-BIC (Hirose et al., 2016) and package sva 3.26.0.
for ComBat (Leek et al., 2017). Hyper-parameters for the Normal-SS and MOM-SS were set as in Section 3.3, the hyper-parameters for FastBFA were set via Dynamic Posterior Exploration as in Roˇckov´a and George (2017) with $1/\lambda_0 = 0.001$ and $1/\lambda_1 \in \{5, 10, 20, 30\}$ and using varimax robustifications. For the LASSO-BIC we selected the model with smallest BIC to set the regularisation parameter. Finally, for scenarios with batch effects, we adjusted the data via a ComBat correction and performed a Factor Analysis via EM algorithm to maximise likelihood with the \texttt{fa.em} function in the \texttt{cate} package (Wang and Zhao, 2015).

5.1 No batch effect

To assess the precision of the parameter estimates returned by the EM algorithm, we simulated data from two different data-generating truths: truly sparse and dense for the loadings $M$. In both, the truth was set to $q^* = 10$ factors. The dense loadings matrix has a grid of elements set uniformly between $(-1, 1)$, whereas the truly sparse $M$ has a banded-diagonal structure with $m_{jk} = 1$ for the non-zero elements, as shown in Figure 2.

Some visual representations of our findings are display in Figure 3 and in the Supplementary Avalos-Pacheco et al. (2020) Figures 3–8.

We simulated $n = 100$ observations from $x_i = M^\star z_i + e_i$, growing $p = 1,000$ and 1,500, where the factors $z_i \sim N(0, I_q)$, the errors $e_i \sim N(0, T^{-1})$ with $T^{-1} = I_p$, and the loadings $M^\star$ are set as dense or sparse as in Figure 2. Our simulations have been normalised to zero mean and unit variance. For comparison, FastBFA was initialised as our models via two-step least-squares (Section 4.3).

Table 1 shows the selected number of factors $\hat{q}$, the number of estimated non-zero loadings $|\hat{M}|_0 = \sum_{j,k} \mathbb{1}(\hat{m}_{jk} \neq 0)$, the Frobenius norm (F.N.) between the true expected value and its reconstruction $\|E[X] - \hat{E}[X]\|_F = \|ZM^T - E[Z | \hat{\Delta}, X]\hat{M}^T\|_F$ and between the true and reconstructed covariances $\|\text{Cov}[x_i] - \hat{\text{Cov}}[x_i]\|_F = \|(MM^T + T^{-1}) - (\hat{M}M^T + T^{-1})\|_F$, the number of iterations until convergence, and the computation time in minutes. The mean across 100 different simulations is displayed.

We first considered the unrealistic scenario where $M$ is dense and one guessed correctly the true number of factors $q = q^* = 10$. The aim of this setting was to investigate if MOM-SS shrinkage provided a poor estimation when the factors were not truly sparse. MOM-SS and Normal-SS performed similarly as $p$ grew, and competitively relative to the flat prior. To extend our example, we then set $q = 100$ to illustrate the performance when there is sparsity in terms of the number of factors, but not within factors. LASSO-BIC had the best reconstruction for the mean but performed poorly on the covariance, whereas FastBFA outperformed all the models to estimate the covariance but performed poorly for the mean. However, MOM-SS had a good balance in terms of estimating the expected value and the covariance, being the second best in both cases but requiring only 6 factors instead of 100.

We further illustrate our model under the arguably more interesting case of truly sparse loadings. First we set $q = 10$ the true cardinality. In this scenario MOM-SS
and Normal-SS presented the best results both for mean and covariance. This example reflects the advantages of shrinkage and the varimax rotation for the initialised in the loadings, leading to good sparse solutions. Finally we considered the same scenario with \( q = 100 \). LASSO-BIC was best to estimate the mean at the cost of reduced precision in the covariance reconstruction. MOM-SS displayed the lowest error for the mean covariance and the first or second lowest error for the covariance, showing a good balance between those metrics.

In general, MOM-SS achieved a good balance between estimating the mean, which is useful for dimensionality reduction, and sparse covariance estimation. Recall that we used a coordinate descent algorithm for the non-local prior, which as a potential drawback could require a larger number of iterations than performing jointly optimising multiple elements in \( M \). However, Table 1 showed that MOM-SS required roughly the
| Model     | \( q \) | \( |M^*| \) | \( \|E[X] - E[X]\|_F \) | \( \|\text{Cov}[x_i] - \text{Cov}[x_i]\|_F \) | \( \text{it} \) | \( \text{time} \) | \( q \) | \( |M^*| \) | \( \|E[X] - E[X]\|_F \) | \( \|\text{Cov}[x_i] - \text{Cov}[x_i]\|_F \) | \( \text{it} \) | \( \text{time} \) |
|-----------|---------|-----------|----------------|----------------|------|--------|---------|---------|----------------|----------------|------|--------|
| **Dense, \( q = 10 \)** |         |           |                 |                 |      |        |         |         |                 |                 |      |        |
| Flat      | 10.0    | 10000.0   | 59.5            | 569.4           | 2.0  | 0.4    | 10.0    | 10000.0 | 73.9           | 917.4           | 2.0  | 1.4    |
| Normal-SS | 10.0    | 1048.5    | 54.5            | 604.8           | 4.1  | 1.0    | 10.0    | 1467.6  | 67.4           | 957.7           | 4.0  | 2.9    |
| MOM-SS    | 10.0    | 986.6     | 54.4            | 607.9           | 4.4  | 1.9    | 10.0    | 1389.2  | 67.2           | 957.5           | 4.0  | 4.5    |
| Fast-BFA  | 9.3     | 959.0     | 78.9            | 594.0           | 11.1 | 0.5    | 9.3     | 1447.9  | 96.2           | 891.1           | 11.0 | 0.8    |
| LASSO-BIC | 10.0    | 5322.8    | 49.5            | 794.2           | NA   | 2.1    | 10.0    | 8600.8  | 60.3           | 1196.9          | NA   | 4.5    |
| **Sparse, \( q = 10 \)** |         |           |                 |                 |      |        |         |         |                 |                 |      |        |
| Flat      | 100.0   | 100000.0  | 163.9           | 576.2           | 3.0  | 4.1    | 100.0   | 100000.0| 201.1          | 926.8           | 3.0  | 10.9   |
| Normal-SS | 7.3     | 1745.3    | 72.1            | 589.8           | 4.0  | 8.7    | 28.6    | 1875.5  | 125.2          | 951.2           | 4.0  | 18.7   |
| MOM-SS    | 6.4     | 1385.6    | 80.8            | 577.1           | 5.0  | 10.4   | 5.9     | 1082.5  | 97.9           | 935.2           | 4.1  | 22.4   |
| Fast-BFA  | 70.7    | 1516.9    | 150.1           | 478.6           | 13.2 | 1.5    | 77.3    | 2331.1  | 191.2          | 703.6           | 12.5 | 2.1    |
| LASSO-BIC | 11.0    | 4830.2    | 45.0            | 794.4           | NA   | 61.5   | 11.2    | 7844.7  | 53.4           | 1197.0          | NA   | 128.4  |
| **Dense, \( q = 100 \)** |         |           |                 |                 |      |        |         |         |                 |                 |      |        |
| Flat      | 10.0    | 10000.0   | 73.5            | 125.3           | 2.0  | 0.5    | 10.0    | 10000.0 | 89.4           | 263.7           | 2.0  | 1.1    |
| Normal-SS | 10.0    | 1298.6    | 43.9            | 89.1            | 3.0  | 1.3    | 10.0    | 1931.4  | 54.2           | 180.7           | 3.0  | 2.7    |
| MOM-SS    | 10.0    | 1296.6    | 43.3            | 80.7            | 3.1  | 2.2    | 10.0    | 1919.3  | 56.2           | 169.4           | 3.0  | 4.3    |
| Fast-BFA  | 9.9     | 778.1     | 60.3            | 165.0           | 13.2 | 0.6    | 9.9     | 1157.8  | 72.8           | 247.7           | 13.3 | 0.7    |
| LASSO-BIC | 10.0    | 5288.7    | 54.9            | 270.2           | NA   | 0.7    | 10.0    | 8414.6  | 67.2           | 408.4           | NA   | 0.8    |
| **Sparse, \( q = 100 \)** |         |           |                 |                 |      |        |         |         |                 |                 |      |        |
| Flat      | 100.0   | 100000.0  | 299.5           | 185.7           | 3.0  | 5.1    | 100.0   | 100000.0| 259.2          | 280.2           | 3.0  | 7.1    |
| Normal-SS | 31.0    | 1228.6    | 109.0           | 144.6           | 4.3  | 9.9    | 56.4    | 1568.2  | 181.3          | 231.9           | 4.0  | 13.8   |
| MOM-SS    | 9.7     | 836.8     | 79.4            | 143.3           | 4.8  | 10.8   | 5.2     | 743.4   | 105.0          | 245.6           | 4.0  | 17.5   |
| Fast-BFA  | 83.6    | 1389.9    | 198.1           | 141.9           | 12.0 | 1.7    | 87.2    | 763.9   | 206.2          | 211.3           | 6.5  | 1.0    |
| LASSO-BIC | 10.0    | 4787.3    | 54.1            | 271.4           | NA   | 19.6   | 10.0    | 7976.6  | 66.1           | 409.3           | NA   | 31.6   |

Table 1: Synthetic data without batch effects for \( n = 100 \), \( q^* = 10 \), \( p = 1,000 \) or \( 1,500 \) parameters, truly sparse and dense loadings \( M^* \).
same number of iterations and the same time to converge as the Normal-SS. We can see that MOM-SS and LASSO-BIC estimated $\hat{q}$ accurately in most scenarios. Note that in general FastBFA had the highest estimated latent cardinality $\hat{q}$, due to the fat tails of the Laplace priors, which adds some columns of $M$ that contain very few non-zero loadings after the tenth factor, as shown in Supplementary Avalos-Pacheco et al. (2020) Sections 10 and 11. Nonetheless, this model displayed a mean number of non-zero loadings closer to the ground truth (1,300 and 1,940 for the $p = 1,000$ and $p = 1,500$ respectively under sparse $M$).

5.2 Batch effects

We evaluate our method in our main setting of interest where there are mean and variance batch effects. We emphasise that, the competing methods are not designed to account for batch effects; thus, this is not a fair comparison but rather an illustration of how much inference can suffer when not properly accounting for batches. Also, since Flat-SS, Normal-SS and MOM-SS do incorporate batches, comparing them illustrates the advantages of NLP-based sparsity, e.g. see the bottom row in Figure 4.

We simulated data with a mean and variance batch effect, $x_i = \theta^* v_i + M^* z_i + \beta^* b_i + e_i$, sample size $n = 200$ and growing $p = 250$ or $p = 500$. We set $q^* = 10$, $p_v = 1$ and $p_b = 2$ batches and considered the truly sparse and dense loadings $M^*$ in Figure 2. Factors $z_i$ were drawn from $N(0, I_q)$, errors $e_i$ from $N(0, \tau^{-1})$, where $\tau^{-1} = 0.5$ and $\tau^{-1}_j = 1.5\tau^{-1}$ for $j = 1, \ldots, p$; $v_i$ from a continuous Uniform$(0,3)$ and $b_i$ from a discrete Uniform${\{0,1\}}$. We set the first $p/2$ values of $\theta^* \in \mathbb{R}^p$ to $-2$ and the other $p/2$ to 2 and $\beta^*_j = 0, \beta^*_j = 2$ for $j = 1, \ldots, p$ we fixed to 2 for the first batch and 0 for the second.

We then normalised our simulations to zero mean and unit variance. We compared our models with FastBFA and LASSO-BIC without batch effect correction for illustration of the importance of a proper mean and variance batch effect adjustment; and with empirical Bayes batch effect correction, ComBat, followed with an MLE estimation of the parameters ComBat-MLE. Table 2 shows the results. The following plots show the comparison between the true $ZM^\top$ against their reconstruction $E[Z \mid \Delta, X]$ in the scenario with sparsity with factors $q = 100$.

Firstly, we considered the scenario when one correctly guesses $q = 10$ and loadings are truly dense and sparse solutions could provide poor estimations. MOM-SS and Normal-SS achieved similar performance as the case without batch effect and similar results were observed for the $q = 100$.

Secondly, we studied the scenario with sparse factors. MOM-SS and Normal-SS achieved a small estimation error for the mean and were effective in estimating $q^* = 10$. LASSO-BIC had a small estimation error of the mean, although solutions were generally less sparse in the number of non-zero loadings.

It is important to highlight that even though FastBFA and LASSO-BIC achieved a precise reconstruction of $E[X]$ for purposes of dimensionality reduction, the estimates of $ZM^\top$ are less precise, as shown in Table 2 and Figure 4 (right panels), clearly suggesting two clusters. We remark that for FastBFA and LASSO-BIC these results mainly highlight that one should take into account batch effects. For Combat-MLE
Figure 3: Scatterplots comparing $Z^T$ vs. $E[Z | \hat{\Delta}, X]M^T$ between the different models under dense (top) and truly sparse (bottom) loadings $M$ with $q = 100$ in simulations without batch effect.

they highlight the limitations of using two-step procedures relative to a joint estimation of the factor model and batch effects. The reconstruction of $Z^TM$ for Combat-MLE corrected the batch effects but was less precise than our suggested models.

Finally, Supplements Avalos-Pacheco et al. (2020) 14 and 15 present the results of our methods without standardising the data to zero mean and unit variance, to assess the sensitivity of the various methods. The results showed that the performance of the two-step method COMBAT-MLE varied significantly, suggesting that such a method may not be robust to re-scaling the data. MOM-SS showed a robust performance across the two scenarios.

5.3 Applications to cancer datasets

We applied our method to three high-dimensional cancer datasets, related to ovarian, lung and colorectal cancer. For the ovarian cancer we combined information from two datasets from the package curatedOvarianData 1.16.0. The first was the Illumina Human microRNA array expression dataset E.MTAB.386, formed by Angiogenic mRNA and microRNA gene expression signature with $n_1 = 129$ patients (Bentink et al., 2012). The second was the NCI-60 GEO dataset GSE30161 and consisted of multi-gene expression predictors of single drug responses to adjuvant chemotherapy in ovarian carcinoma for $n_2 = 52$ patients (Ferriss et al., 2012). For the lung cancer, we used microarray and mRNA-array, data from two different high-throughput platforms: Affymetrix Human Genome U133A 2.0 Array with $n_1 = 133$ patients and Affymetrix Human Exon 1.0 ST Array with $n_2 = 112$ (Wan et al., 2016). For the colorectal cancer we integrated the two colon cancer datasets used by Calon et al. (2012): The Omnibus gene expression dataset GSE17538 with $n_1 = 232$ patients and the gene expression data of $n_2 = 101$ patients from the Austraili hospital GSE14333.
| Model       | $q$ | $M_{10}$ | $||E[X] - E[Z]| | $||ZM^T - E[Z] | \Delta X | M^T ||_F$ | it | time |
|------------|-----|---------|----------------|-----------------|----|------|
| Dense $M$, $q = 10$ |     |         |               |                 |    |      |
| Flat       | 10.0 | 2500.0 | 34.4          | 104.3           | 7.0 | 0.3  |
| Normal-SS  | 10.0 | 511.1  | 32.4          | 103.1           | 8.3 | 0.3  |
| MOM-SS     | 10.0 | 395.3  | 33.3          | 103.3           | 14.6| 0.6  |
| ComBat-MLE | 10.0 | 2500.0 | 103.6         | 107.8           | 2.0 | 0.0  |
| FastBFA    | 7.9  | 271.7  | 38.0          | 143.6           | 9.2 | 0.0  |
| LASSO-BIC  | 10.0 | 1290.7 | 34.0          | 143.3           | 0.0 | 0.6  |

<table>
<thead>
<tr>
<th>Dense $M$, $q = 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
</tr>
<tr>
<td>100.0</td>
</tr>
<tr>
<td>Normal-SS</td>
</tr>
<tr>
<td>8.6</td>
</tr>
<tr>
<td>MOM-SS</td>
</tr>
<tr>
<td>7.4</td>
</tr>
<tr>
<td>ComBat-MLE</td>
</tr>
<tr>
<td>100.0</td>
</tr>
<tr>
<td>FastBFA</td>
</tr>
<tr>
<td>9.9</td>
</tr>
<tr>
<td>LASSO-BIC</td>
</tr>
<tr>
<td>11.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sparse $M$, $q = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>Normal-SS</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>MOM-SS</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>ComBat-MLE</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>FastBFA</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>LASSO-BIC</td>
</tr>
<tr>
<td>10.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sparse $M$, $q = 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat</td>
</tr>
<tr>
<td>100.0</td>
</tr>
<tr>
<td>Normal-SS</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>MOM-SS</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>ComBat-MLE</td>
</tr>
<tr>
<td>100.0</td>
</tr>
<tr>
<td>FastBFA</td>
</tr>
<tr>
<td>10.0</td>
</tr>
<tr>
<td>LASSO-BIC</td>
</tr>
<tr>
<td>10.3</td>
</tr>
</tbody>
</table>

Table 2: Synthetic data with batch effects for $n = 200$, $q^* = 10$, $p = 250$ or 500 parameters, truly sparse and dense loadings $M^*$. 

...
We considered two main tasks: to give a visual representation of the latent factors of the data, i.e. an unsupervised dimension reduction task and a supervised survival analysis using the factors obtained in our method as predictions. Prior to our analyses, for ovarian and lung datasets we selected the 10% genes with highest total variance across all samples obtaining $p = 1,007$ for ovarian and $p = 1,198$ for lung, and we included the age at initial pathologic diagnosis as a covariate. For colorectal cancer we included tumour stage as covariate and we considered the $p = 172$ genes identified in Calon et al. (2012) as they are potentially related to the Transforming Growth Factor beta (TGF-β) pathway. TGF-β signatures are essential to a better understanding of colorectal cancers, as a large proportion of such cancers display a high production of TGF-β. All data sets have been normalised to zero mean and unit variance.

**Unsupervised: data visualisation**

Our first goal was to demonstrate the usefulness of our method as a data visualisation tool. We remark that there are no other model-based approaches to jointly adjust for batch effects and estimate latent factors. Thus, for comparison we first corrected the data using ComBat and then estimated the latent parameters via MLE and FastBFA akin to Section 5.2. To decide the number of factors for ComBat-MLE, we carried a principal component analysis to the corrected data prior to factor analysis and chose a number of components $\hat{q}$ that explained 90% or 70% of the total variance. It is important to notice that we are doing an over-optimistic assessment of ComBat-MLE and ComBat-FastBFA as we are doing a cross-validated factor analysis over the ComBat-corrected data, as opposed to also running ComBat in an out-of-sample fashion.

Figure 5 illustrates the advantages of our method. We can appreciate the usefulness of ComBat correction (middle panels) relative to the raw uncorrected data (top panels). ComBat removed systematic differences in location and scale across the two
batches. The latent coordinates displayed distinct covariances for the ovarian dataset which, given that patients in both batches are thought to be roughly exchangeable, is likely due to a technical artifact. Such distinct covariances were not present in the MOM-SS latent factors (bottom panels). In all panels we displayed the two factors that contributed the most to the covariance, i.e. with highest $\sum_{j=1}^{p} \hat{m}_{jk}^2$: the latent coordinates were post-processed to standardise their variance $\text{Cov}(z_i \mid \hat{\Delta}, X)$, as explained in Section 4.4. For colon cancer, ComBat-MLE (5f) again showed differences in the covariance structure of the two different batches. Such differences were not present in the ComBat-FastBFA (5i) and MOM-SS (5l) latent factors, as expected due to the batch exchangeability. Differences between ComBat-MLE and ComBat-FastBE highlighted
the variability of two-step methods. MOM-SS showed a robust performance across the different datasets.

**Supervised: survival analysis**

We also illustrate the potential of our method as a supervised tool, performing a survival analysis that aims to predict the time until death. To do that, we applied a Cox proportional hazards model (Cox, 1972) using as covariates the latent coordinates obtained in our models. We used the `coxph` function of the R package `survival 2.38` (Therneau, 2015). We then used the concordance index to assess the quality of our predictions. This index is a non-parametric metric to quantify the power of a prediction rule via a pair-wise comparison that measures the probability of concordance between the predicted and the observed survival time (Harrell Jr. et al., 1982). To obtain the concordance index we used the function `concordance.index` in the R package `survcomp` (Schröder et al., 2011). The presented results are from 10 independent runs of 10-fold cross-validation. We initialised MOM-SS with the values obtained for the Flat model along with the other initialisations discussed in Section 4.3 and chose the one with smallest leave-one-out cross-validated concordance index.

We first performed survival analysis to each batch separately for all the cancer datasets (non-integrative analysis). We then conducted the integrative survival analysis. We finally compared the concordance index obtained with non-integrative and integrative analyses. Our results in Table 3 show that, for the ovarian cancer dataset, the MOM-SS integrative analysis displays competitive CI compared to the non-integrative survival analysis, with a sparser latent representation, requiring only 4 factors. Further, for lung and colorectal cancers, integrative MOM-SS shows a higher CI than the non-integrative survival analysis (from CI = 0.522 to CI = 0.665 for lung and from CI = 0.736 to CI = 0.764 for colorectal cancer). Our results highlight the importance of data integration and batch effect correction, as our integrative method is able to capture common latent information of different studies and provides competitive or higher CI compared to the single standard factor analysis.

For the integrative analyses of the cancer data sets, Table 3 shows that Flat-SS achieved a high concordance index, even though loadings are not sparse. Normal-SS gave sparse loading representations but displayed a concordance index lower than Flat-SS; this illustrates a lack of power to detect truly non-zero loadings. In general, MOM-SS provided sparse loadings and a good concordance index. In the ovarian cancer data, MOM-SS achieved a concordance index similar to ComBat-MLE 90% with considerably less factors (4 instead of 101) and a bit higher than Normal-SS. In the lung cancer data MOM-SS achieved a high concordance index, particularly relative to Normal-SS and ComBat-MLE 70%. In the colon cancer data the highest concordance index was obtained by MOM-SS. The competing methods generally lead to less sparse solutions and their performance fluctuates across scenarios. In the lung cancer data ComBat-MLE, despite its good performance, had a concordance index that proved to be sensitive to the number of factors (see ComBat-MLE 90% vs 70%). ComBat-FastBFA provided competitive results with a non-sparse reconstruction, recovering values in the latent loadings that were close to zero (even though not exactly zero) and smaller than the
ones of the Flat-SS. MOM-SS proved to have practical advantages as a supervised tool in comparison with the two-step approaches considered here. Overall, MOM-SS provided a more stable performance that achieved a good balance between sparsity and prediction accuracy.

<table>
<thead>
<tr>
<th></th>
<th>Ovarian</th>
<th>Lung</th>
<th>Colon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch 1-MLE 90%</td>
<td>67.1</td>
<td>52.1</td>
<td>52.9</td>
</tr>
<tr>
<td>Batch 1-MLE 70%</td>
<td>27.0</td>
<td>35.2</td>
<td>17.0</td>
</tr>
<tr>
<td>Batch 2-MLE 90%</td>
<td>40.4</td>
<td>36.6</td>
<td>48.1</td>
</tr>
<tr>
<td>Batch 2-MLE 70%</td>
<td>23.4</td>
<td>23.2</td>
<td>23.3</td>
</tr>
<tr>
<td>Flat Normal-SS</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>MOM-SS</td>
<td>4.0</td>
<td>74.0</td>
<td>53.4</td>
</tr>
<tr>
<td>ComBat-MLE 90%</td>
<td>101.0</td>
<td>79.0</td>
<td>67.0</td>
</tr>
<tr>
<td>ComBat-MLE 70%</td>
<td>41.0</td>
<td>30.0</td>
<td>24.0</td>
</tr>
<tr>
<td>ComBat-FastBFA</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 3: Survival analysis for ovarian ($p = 1,007$ genes), lung ($p = 1,198$ genes) and colon ($p = 172$ genes) cancer data sets.

6 Discussion

We have presented a novel model to integrate data from multiple sources using joint dimension reduction and batch effect adjustment via high-dimensional latent factor regression. We outlined three different prior configurations for the loadings and Laplace-tailed extensions whose deeper analysis remain as future work. To our knowledge this is the first time NLPs are implemented in the factor analysis context. We gave novel EM algorithms to obtain posterior modes. We showed that the use of sparse models increases the quality of our estimations even in the absence of batches. In our empirical results MOM-SS priors proved to be appealing, improving the estimation of factor cardinality and encouraging parsimony and selective shrinkage.

We illustrated the utility of our method in unsupervised and supervised frameworks. MOM-SS provided dimension reduction that corrected distinct covariance patterns present in two-stage methods that adjust variances separately from fitting the factor model. Such patterns are highly likely to be technical artefacts, since patients from different batches are believed to be exchangeable. Our model demonstrated to be useful for downstream analyses, achieving a competitive concordance indexes, in some cases with substantially less factors. It is important to notice that although our examples focus on gene expression of cancer datasets, the applications should also be useful in other settings.

We also remark that our novel MOM-SS and its closed-form EM updates can be extended to frameworks of interest beyond factor models such as: linear regression, generalised linear models as well as graphical models.

Our model assumes common factors across the datasets being integrated. An interesting extension for future research is to consider more complex settings where some
of the factors differ across data sources or where one wishes to integrate datasets by adding variables (as opposed to adding individuals as we did here), or where potentially same variables were only recorded for a subset of the individuals.

**Supplementary Material**

Supplementary material to “Heterogeneous large datasets integration using Bayesian factor regression” (DOI: 10.1214/20-BA1240SUPP; .pdf). The supplementary materials are as follow: Proofs for Lemma 1, Lemma 2 and Lemma 3. EM algorithm under a flat, Normal-SS, MOM-SS, Laplace-SS and Laplace-MOM-SS on the loadings, a pseudo-code-algorithm for the weighted 10-fold cross-validation, and heatmaps for $\hat{M}$, $\hat{\Cov}(\hat{x}_i | \cdot)^{-1}$ and $\hat{\gamma}$ for the different simulated scenarios and setting $q = 100$. Robust analysis for simulations without batch effect and with batch effect.

**References**


Heterogeneous Datasets Integration Using Bayesian Factor Regression


A. Avalos-Pacheco, D. Rossell, and R. S. Savage


Expression and Proteomics, 155–176, Cambridge University Press. MR2655722. doi: https://doi.org/10.1198/016214508000000869. 34


Heterogeneous Datasets Integration Using Bayesian Factor Regression


Acknowledgments

We thank Chris Yau for valuable insights and Veronika Roˇckov´a for providing the FastBFA package.