Journal of Machine Learning Research 17 (2016) 1-47 Submitted 11/14; Revised 11/15; Published 4/16

Bayesian group factor analysis with structured sparsity

Shiwen Zhao SHIWEN.ZHAO@DUKE.EDU
Computational Biology and Bioinformatics Program
Department of Statistical Science
Duke University
Durham, NC 27708, USA

Chuan Gao CHUAN.GAO@DUKE.EDU
Department of Statistical Science
Duke University
Durham, NC 27708, USA

Sayan Mukherjee SAYAN@STAT.DUKE.EDU
Departments of Statistical Science, Computer Science, Mathematics
Duke University
Durham, NC 27708, USA

Barbara E Engelhardt BEE@PRINCETON.EDU
Department of Computer Science
Center for Statistics and Machine Learning
Princeton University
Princeton, NJ 08540, USA

Editor: Samuel Kaski

Abstract

Latent factor models are the canonical statistical tool for exploratory analyses of low-dimensional linear structure for a matrix of p features across n samples. We develop a structured Bayesian group factor analysis model that extends the factor model to multiple coupled observation matrices; in the case of two observations, this reduces to a Bayesian model of canonical correlation analysis. Here, we carefully define a structured Bayesian prior that encourages both element-wise and column-wise shrinkage and leads to desirable behavior on high-dimensional data. In particular, our model puts a structured prior on the joint factor loading matrix, regularizing at three levels, which enables element-wise sparsity and unsupervised recovery of latent factors corresponding to structured variance across arbitrary subsets of the observations. In addition, our structured prior allows for both dense and sparse latent factors so that covariation among either all features or only a subset of features can be recovered. We use fast parameter-expanded expectation-maximization for parameter estimation in this model. We validate our method on simulated data with substantial structure. We show results of our method applied to three high-dimensional data sets, comparing results against a number of state-of-the-art approaches. These results illustrate useful properties of our model, including i) recovering sparse signal in the presence of dense effects; ii) the ability to scale naturally to large numbers of observations; iii) flexible observation- and factor-specific regularization to recover factors with a wide variety of sparsity levels and percentage of variance explained; and iv) tractable inference that scales to modern genomic and text data sizes.

Keywords: Bayesian structured sparsity, canonical correlation analysis, sparse priors, sparse and low-rank matrix decomposition, mixture models, parameter expansion

1. Introduction

Factor analysis models have attracted attention recently due to their ability to perform exploratory analyses of the latent linear structure in high-dimensional data (West, 2003; Carvalho et al., 2008; Engelhardt and Stephens, 2010). A latent factor model finds a low-dimensional representation $\boldsymbol{x}_i \in \mathbb{R}^{k\times 1}$ of high-dimensional data with p features, $\boldsymbol{y}_i \in \mathbb{R}^{p\times 1}$ in $i = 1, \ldots, n$ samples. A sample in the low-dimensional space is linearly projected to the original high-dimensional space through a *loadings matrix* $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times k}$ with Gaussian noise $\boldsymbol{\epsilon}_i \in \mathbb{R}^{p\times 1}$:

$$\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \tag{1}$$

for $i = 1, \ldots, n$. It is often assumed that $\boldsymbol{x}_i$ follows a $\mathcal{N}_k(\boldsymbol{0}, \boldsymbol{I}_k)$ distribution, where $\boldsymbol{I}_k$ is the identity matrix of dimension k, and $\boldsymbol{\epsilon}_i \sim \mathcal{N}_p(\boldsymbol{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a $p \times p$ diagonal covariance matrix with σ_j^2 for $j = 1, \ldots, p$ on the diagonal. In many applications of factor analysis, the number of latent factors k is much smaller than the number of features p and the number of samples n. Integrating over factor $\boldsymbol{x}_i$, this model produces a low-rank estimation of the feature covariance matrix. In particular, the covariance of $\boldsymbol{y}_i$, $\boldsymbol{\Omega} \in \mathbb{R}^{p\times p}$, is estimated as

$$\boldsymbol{\Omega} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Sigma} = \sum_{h=1}^{k} \boldsymbol{\lambda}_{\cdot h}\boldsymbol{\lambda}_{\cdot h}^T + \boldsymbol{\Sigma},$$

where $\boldsymbol{\lambda}_{\cdot h}$ is the h^{th} column of $\boldsymbol{\Lambda}$. This factorization suggests that each factor contributes to the covariance of the sample through its corresponding loading. Traditional exploratory data analysis methods including principal component analysis (PCA) (Hotelling, 1933), independent component analysis (ICA) (Comon, 1994), and canonical correlation analysis (CCA) (Hotelling, 1936) all have interpretations as latent factor models. Indeed, the field of latent variable models is extremely broad, and robust unifying frameworks are desirable (Cunningham and Ghahramani, 2015).

Considering latent factor models (Equation 1) as capturing a low-rank estimate of the feature covariance matrix, we can characterize canonical correlation analysis (CCA) as modeling paired observations $\boldsymbol{y}_i^{(1)} \in \mathbb{R}^{p_1\times 1}$ and $\boldsymbol{y}_i^{(2)} \in \mathbb{R}^{p_2\times 1}$ across n samples to identify a linear latent space for which the correlations between the two observations are maximized (Hotelling, 1936; Bach and Jordan, 2005). The Bayesian CCA (BCCA) model extends this covariance representation to two observations: the combined loading matrix jointly models covariance structure shared across both observations and covariance local to each observation (Klami et al., 2013). Group factor analysis (GFA) models further extend this representation to m coupled observations for the same sample, modeling, in its fullest generality, the covariance associated with every subset of observations (Virtanen et al., 2012; Klami et al., 2014b). GFA becomes intractable when m is large due to exponential explosion of covariance matrices to estimate.

In a latent factor model, the loading matrix $\boldsymbol{\Lambda}$ plays an important role in the subspace mapping. In applications where there are fewer samples than features—the $n \ll p$ scenario (West, 2003)—it is essential to include strong regularization on the loading matrix

because the optimization problem is under-constrained and has many equivalent solutions that optimize the data likelihood. In the machine learning and statistics literature, priors or penalties are used to regularize the elements of the loading matrix, occasionally by inducing sparsity. Element-wise sparsity corresponds to *feature selection*. This has the effect that a latent factor contributes to variation in only a subset of the observed features, generating interpretable results (West, 2003; Carvalho et al., 2008; Knowles and Ghahramani, 2011). For example, in gene expression analysis, sparse factor loadings are interpreted as non-disjoint clusters of co-regulated genes (Pournara and Wernisch, 2007; Lucas et al., 2010; Gao et al., 2013).

Element-wise sparsity has been imposed in latent factor models through regularization via ℓ_1 type penalties (Zou et al., 2006; Witten et al., 2009; Salzmann et al., 2010). More recently, Bayesian shrinkage methods using sparsity-inducing priors have been introduced for latent factor models (Archambeau and Bach, 2009; Carvalho et al., 2008; Virtanen et al., 2012; Bhattacharya and Dunson, 2011; Klami et al., 2013). The spike-and-slab prior (Mitchell and Beauchamp, 1988), the classic two-groups Bayesian sparsity-inducing prior, has been used for sparse Bayesian latent factor models (Carvalho et al., 2008). A computationally tractable one-group prior, the automatic relevance determination (ARD) prior (Neal, 1995; Tipping, 2001), has also been used to induce sparsity in latent factor models (Engelhardt and Stephens, 2010; Pruteanu-Malinici et al., 2011). More sophisticated structured regularization approaches for linear models have been studied in classical statistics (Zou and Hastie, 2005; Kowalski and Torrésani, 2009; Jenatton et al., 2011; Huang et al., 2011).

Global structured regularization of the loading matrix, in fact, has been used to extend latent factor models to multiple observations. The BCCA model (Klami et al., 2013) assumes a latent factor model for each observation through a shared latent vector $\boldsymbol{x}_i \in \mathbb{R}^{k\times 1}$. This BCCA model may be written as a latent factor model by vertical concatenation of observations, loading matrices, and Gaussian residual errors. By inducing group-wise sparsity—explicit blocks of zeros—in the combined loading matrix, the covariance shared across the two observations and the covariance local to each observation are estimated (Klami and Kaski, 2008; Klami et al., 2013). Extensions of this approach to multiple coupled observations $\boldsymbol{y}_i^{(1)} \in \mathbb{R}^{p_1\times 1}, \ldots, \boldsymbol{y}_i^{(m)} \in \mathbb{R}^{p_m\times 1}$ have resulted in group factor analysis models (GFA) (Archambeau and Bach, 2009; Salzmann et al., 2010; Jia et al., 2010; Virtanen et al., 2012).

In addition to linear factor models, flexible non-linear latent factor models have been developed. The Gaussian process latent variable model (GPLVM) (Lawrence, 2005) extends Equation (1) to non-linear mappings with a Gaussian process prior on latent variables. Extensions of GPLVM include models that allow multiple observations (Shon et al., 2005; Ek et al., 2008; Salzmann et al., 2010; Damianou et al., 2012). Although our focus will be on linear maps, we will keep the non-linear possibility open for model extensions, and we will include the GPLVM model in our model comparisons.

The primary contribution of this study is that we develop a GFA model using Bayesian shrinkage with hierarchical structure that encourages both element-wise and column-wise sparsity; the resulting flexible Bayesian GFA model is called BASS (Bayesian group factor Analysis with Structured Sparsity). The structured sparsity in our model is achieved with multi-scale application of a hierarchical sparsity-inducing prior that has a computa-

tionally tractable representation as a scale mixture of normals, the three parameter beta prior ($\mathcal{TPB}$) (Armagan et al., 2011; Gao et al., 2013). Our BASS model i) shrinks the loading matrix globally, removing factors that are not supported in the data; ii) shrinks loading columns to decouple latent spaces from arbitrary subsets of observations; iii) allows factor loadings to have either an element-wise sparse or a non-sparse prior, combining interpretability with dimension reduction. In addition, we developed a parameter-expanded expectation maximization (PX-EM) method based on rotation augmentation to tractably find *maximum a posteriori* estimates of the model parameters (Rocková and George, 2015). PX-EM has the same computational complexity as the standard EM algorithm, but produces more robust solutions by enabling fast searching over posterior modes.

In Section 2 we review current work in sparse latent factor models and describe our BASS model. In Sections 3 and 4, we briefly review Bayesian shrinkage priors and introduce the structured hierarchical prior in BASS. In Section 5, we introduce our PX-EM algorithms for parameter estimation. In Section 6, we show the behavior of our model for recovering simulated sparse signals among m observation matrices and compare the results from BASS with state-of-the-art methods. In Section 7, we present results that illustrate the performance of BASS on three high-dimensional data sets. We first show that the estimates of shared factors from BASS can be used to perform multi-label learning and prediction in the Mulan Library data and the 20 Newsgroups data. Then we demonstrate that BASS can be used to find biologically meaningful structure and construct condition-specific co-regulated gene networks using the sparse factors specific to observations. We conclude by considering possible extensions to this model in Section 8.

2. Bayesian group factor model

Here, we review current work in sparse latent factor models and describe our Bayesian group factor Analysis with Structured Sparsity (BASS) model in the context of related work.

2.1 Latent factor models

Factor analysis has been extensively used for dimension reduction and low-dimensional covariance matrix estimation. For concreteness, we re-write the basic factor analysis model here as

$$\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i,$$

where $\boldsymbol{y}_i \in \mathbb{R}^{p\times 1}$ is modeled as a linear transformation of a latent vector $\boldsymbol{x}_i \in \mathbb{R}^{k\times 1}$ through loading matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times k}$ (Figure 1A). Here, $\boldsymbol{x}_i$ is assumed to follow a $\mathcal{N}_k(\boldsymbol{0}, \boldsymbol{I}_k)$ distribution, where $\boldsymbol{I}_k$ is the k-dimensional identity matrix, and $\boldsymbol{\epsilon}_i \sim \mathcal{N}_p(\boldsymbol{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a $p\times p$ diagonal matrix. With an isotropic noise assumption, $\boldsymbol{\Sigma} = \boldsymbol{I}\sigma^2$, this model has a probabilistic principal components analysis interpretation (Roweis, 1998; Tipping and Bishop, 1999b). For factor analysis, and in this work, it is assumed that $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \cdots, \sigma_p^2)$ representing independent idiosyncratic noise (Tipping and Bishop, 1999a).

Integrating over the factors $\boldsymbol{x}_i$, we see that the covariance of $\boldsymbol{y}_i$ is estimated with a low-rank matrix factorization: $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Sigma}$. We further let $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots \boldsymbol{y}_n]$ be the collection of n samples $\boldsymbol{y}_i$, and similarly let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ and $\boldsymbol{E} = [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n]$. Then the factor

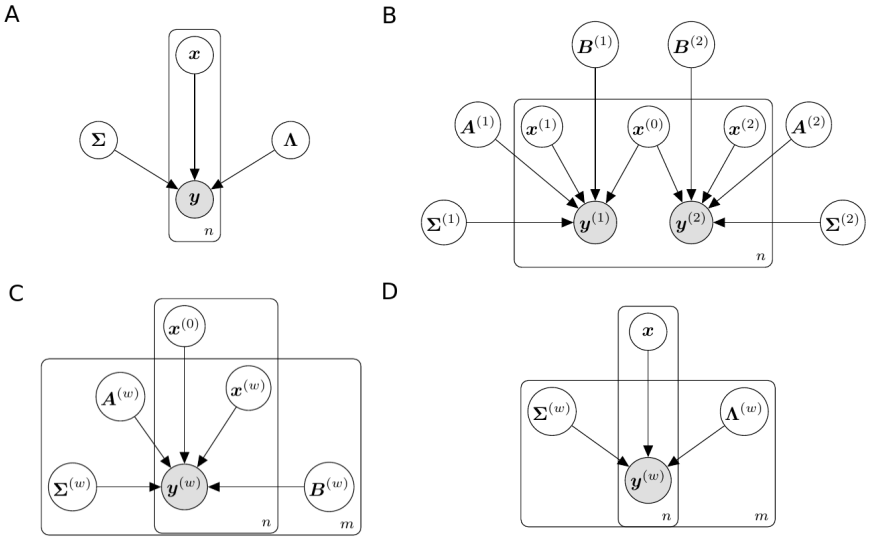

Figure 1: **Graphical representation of different latent factor models.** Panel A: Factor analysis model. Panel B: Bayesian canonical correlation analysis model (BCCA). Panel C: An extension of BCCA model to multiple observations. Panel D: Our Bayesian group factor analysis model (BASS).

analysis model for the observation $\boldsymbol{Y}$ is written as

$$\boldsymbol{Y} = \boldsymbol{\Lambda}\boldsymbol{X} + \boldsymbol{E}. \tag{2}$$

2.2 Probabilistic canonical correlation analysis

In the context of two paired observations $\boldsymbol{y}_i^{(1)} \in \mathbb{R}^{p_1 \times 1}$ and $\boldsymbol{y}_i^{(2)} \in \mathbb{R}^{p_2 \times 1}$ on the same n samples, canonical correlation analysis (CCA) seeks to find linear projections (canonical directions) such that the sample correlations in the projected space are mutually maximized (Hotelling, 1936). The work of interpreting CCA as a probabilistic model can be traced back to classical descriptions (Bach and Jordan, 2005). With a common latent factor, $\boldsymbol{x}_i \in \mathbb{R}^{k \times 1}$, $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{y}_i^{(2)}$ are modeled as

$$\begin{aligned} \boldsymbol{y}_i^{(1)} &= \boldsymbol{\Lambda}^{(1)}\boldsymbol{x}_i + \boldsymbol{e}_i^{(1)}, \\ \boldsymbol{y}_i^{(2)} &= \boldsymbol{\Lambda}^{(2)}\boldsymbol{x}_i + \boldsymbol{e}_i^{(2)}. \end{aligned} \tag{3}$$

In this model, the errors are distributed as $\boldsymbol{e}_i^{(1)} \sim \mathcal{N}_{p_1}(\boldsymbol{0}, \boldsymbol{\Psi}^{(1)})$ and $\boldsymbol{e}_i^{(2)} \sim \mathcal{N}_{p_2}(\boldsymbol{0}, \boldsymbol{\Psi}^{(2)})$, where $\boldsymbol{\Psi}^{(1)}$ and $\boldsymbol{\Psi}^{(2)}$ are positive semi-definite matrices, and not necessarily diagonal, allowing dependencies among the residual errors within an observation. The maximum likelihood estimates of the loading matrices in the classical CCA framework, $\boldsymbol{\Lambda}^{(1)}$ and $\boldsymbol{\Lambda}^{(2)}$, are the first k canonical directions up to orthogonal transformations (Bach and Jordan, 2005).

2.3 Bayesian CCA with group-wise sparsity

Building on the probabilistic CCA model, a Bayesian CCA (BCCA) model has the following form (Klami et al., 2013)

$$\begin{aligned} \boldsymbol{y}_i^{(1)} &= \boldsymbol{A}^{(1)}\boldsymbol{x}_i^{(0)} + \boldsymbol{B}^{(1)}\boldsymbol{x}_i^{(1)} + \boldsymbol{\epsilon}_i^{(1)}, \\ \boldsymbol{y}_i^{(2)} &= \boldsymbol{A}^{(2)}\boldsymbol{x}_i^{(0)} + \boldsymbol{B}^{(2)}\boldsymbol{x}_i^{(2)} + \boldsymbol{\epsilon}_i^{(2)}, \end{aligned} \tag{4}$$

with $\boldsymbol{x}_i^{(0)} \in \mathbb{R}^{k_0 \times 1}$, $\boldsymbol{x}_i^{(1)} \in \mathbb{R}^{k_1 \times 1}$ and $\boldsymbol{x}_i^{(2)} \in \mathbb{R}^{k_2 \times 1}$ (Figure 1B). The latent vector $\boldsymbol{x}_i^{(0)}$ is shared by both $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{y}_i^{(2)}$, and captures their common variation through loading matrices $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$. Two additional latent vectors, $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$, are specific to each observation; they are multiplied by observation-specific loading matrices $\boldsymbol{B}^{(1)}$ and $\boldsymbol{B}^{(2)}$. The two residual error terms are $\boldsymbol{\epsilon}_i^{(1)} \sim \mathcal{N}_{p_1}(\boldsymbol{0}, \boldsymbol{\Sigma}^{(1)})$ and $\boldsymbol{\epsilon}_i^{(2)} \sim \mathcal{N}_{p_2}(\boldsymbol{0}, \boldsymbol{\Sigma}^{(2)})$, where $\boldsymbol{\Sigma}^{(1)}$ and $\boldsymbol{\Sigma}^{(2)}$ are diagonal matrices. This model was originally called inter-battery factor analysis (IBFA) (Browne, 1979) and recently has been studied under a full Bayesian inference framework (Klami et al., 2013). It may be interpreted as the probabilistic CCA model (Equation 3) with an additional low-rank factorization of the observation-specific error covariance matrices. In particular, we re-write the residual error term specific to observation w ($w = 1, 2$) from the probabilistic CCA model (Equation 3) as $\boldsymbol{e}_i^{(w)} = \boldsymbol{B}^{(w)}\boldsymbol{x}_i^{(w)} + \boldsymbol{\epsilon}_i^{(w)}$; then marginally $\boldsymbol{e}_i^{(w)} \sim \mathcal{N}_{p_w}(\boldsymbol{0}, \boldsymbol{\Psi}^{(w)})$ where $\boldsymbol{\Psi}^{(w)} = \boldsymbol{B}^{(w)}(\boldsymbol{B}^{(w)})^T + \boldsymbol{\Sigma}^{(w)}$.

Recent work has re-written the BCCA model as a factor analysis model with group-wise sparsity in the loading matrix (Klami et al., 2013). Let $\boldsymbol{y}_i \in \mathbb{R}^{p \times 1}$ (where $p = p_1 + p_2$) be the vertical concatenation of $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{y}_i^{(2)}$; let $\boldsymbol{x}_i \in \mathbb{R}^{k \times 1}$ (where $k = k_0 + k_1 + k_2$) be the vertical concatenation of $\boldsymbol{x}_i^{(0)}$, $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$; and let $\boldsymbol{\epsilon}_i \in \mathbb{R}^{p \times 1}$ be the vertical concatenation of the two residual errors. Then, the BCCA model (Equation 4) may be written as a factor analysis model

$$\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i,$$

with $\boldsymbol{\epsilon}_i \sim N_p(\boldsymbol{0}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{A}^{(1)} & \boldsymbol{B}^{(1)} & \boldsymbol{0} \\ \boldsymbol{A}^{(2)} & \boldsymbol{0} & \boldsymbol{B}^{(2)} \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}^{(2)} \end{bmatrix}.$$

The structure in the loading matrix $\boldsymbol{\Lambda}$ has a specific meaning: the non-zero columns (i.e., $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$) project the shared latent factors (i.e., the first k_0 elements of $\boldsymbol{x}_i$) to $\boldsymbol{y}_i^{(1)}$ and $\boldsymbol{y}_i^{(2)}$, respectively; these latent factors represent the covariance shared across the observations. The columns with zero blocks (i.e., $[\boldsymbol{B}^{(1)}; \boldsymbol{0}]$ or $[\boldsymbol{0}; \boldsymbol{B}^{(2)}]$) relate factors to only one of the two observations; they model covariance specific to that observation. Under this model, the block sparse structure of $\boldsymbol{\Lambda}$ is imposed via observation-wise sparsity on each factor.

2.4 Extensions to multiple observations

Classical and Bayesian extensions of the CCA model to allow multiple observations ($m > 2$) have been proposed (McDonald, 1970; Browne, 1980; Archambeau and Bach, 2009; Qu and

Chen, 2011; Ray et al., 2014). Generally, these approaches partition the latent variables into those that are shared and those that are observation-specific as follows:

$$\boldsymbol{y}_i^{(w)} = \boldsymbol{A}^{(w)}\boldsymbol{x}_i^{(0)} + \boldsymbol{B}^{(w)}\boldsymbol{x}_i^{(w)} + \boldsymbol{\epsilon}_i^{(w)} \quad \text{for} \quad w = 1, \ldots, m.$$

By vertical concatenation of $\boldsymbol{y}_i^{(w)}$, $\boldsymbol{x}_i^{(w)}$ and $\boldsymbol{\epsilon}_i^{(w)}$, this model can be viewed as a latent factor model (Equation 1) with the joint loading matrix $\boldsymbol{\Lambda}$ having a similar observation-wise sparsity pattern as the BCCA model

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{A}^{(1)} & \boldsymbol{B}^{(1)} & \cdots & \mathbf{0} \\ \boldsymbol{A}^{(2)} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{(m)} & \mathbf{0} & \cdots & \boldsymbol{B}^{(m)} \end{bmatrix}. \tag{5}$$

Here, the first column of blocks ($\boldsymbol{A}^{(w)}$) is a non-zero loading matrix across the features of all observations; the remaining columns have a block diagonal structure with observation-specific loading matrices ($\boldsymbol{B}^{(w)}$) on the diagonal. However, those extensions are limited by the strict diagonal structure of the loading matrix. Structuring the loading matrix in this way prevents this model from capturing covariance structure among arbitrary subsets of observations. On the other hand, there are an exponential number of possible subsets of observations, making estimation of covariance structure among all observation subsets intractable for large m.

The structure on $\boldsymbol{\Lambda}$ in Equation (5) has been relaxed to model covariance among subsets of the observations (Jia et al., 2010; Virtanen et al., 2012; Klami et al., 2014b). In the relaxed formulation, each observation $\boldsymbol{y}_i^{(w)}$ is modeled by its own loading matrix $\boldsymbol{\Lambda}^{(w)}$ and a shared latent vector $\boldsymbol{x}_i$ (Figure 1D):

$$\boldsymbol{y}_i^{(w)} = \boldsymbol{\Lambda}^{(w)}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i^{(w)} \quad \text{for} \quad w = 1, \ldots, m. \tag{6}$$

By allowing columns in $\boldsymbol{\Lambda}^{(w)}$ to be zero, the model decouples certain latent factors from certain observations. The covariance structure of an arbitrary subset of observations is modeled by factors with non-zero loading columns corresponding to the observations in that subset. Factors that correspond to non-zero entries for only one observation capture covariance specific to that observation. Two different approaches have been proposed to achieve column-wise shrinkage in this framework: Bayesian shrinkage (Virtanen et al., 2012; Klami et al., 2014b) and explicit penalties (Jia et al., 2010). The group factor analysis (GFA) model puts an ARD prior (Tipping, 2001) on the loading column for each observation to allow column-wise shrinkage (Virtanen et al., 2012; Klami et al., 2014b):

$$\begin{aligned} \lambda_{jh}^{(w)} &\sim \mathcal{N}\left(0, \left(\alpha_h^{(w)}\right)^{-1}\right) \quad \text{for} \quad j = 1, \ldots, p_w, \\ \alpha_h^{(w)} &\sim Ga(a_0, b_0), \end{aligned}$$

for observation $w = 1, \ldots, m$ and loading column $h = 1, \ldots, k$. This prior assumes that each element of observation-specific loading $\boldsymbol{\lambda}_{\cdot h}^{(w)}$ is jointly regularized. This prior encourages the

parameter $\alpha_h^{(w)}$ to have large values or values near zero, either pushing elements of $\boldsymbol{\lambda}_{\cdot h}^{(w)}$ toward zero or imposing minimal shrinkage, and enabling observation-specific, column-wise sparsity.

Other work puts alternative structured regularizers on $\boldsymbol{\Lambda}^{(w)}$ (Jia et al., 2010). To induce observation-specific, column-wise sparsity, GFA used mixed norms: an ℓ_1 norm penalizes each observation-specific column, and either ℓ_2 or ℓ_∞ norms penalize the elements in an observation-specific column:

$$\phi(\boldsymbol{\Lambda}^{(w)}) = \sum_{h=1}^{k} ||\boldsymbol{\lambda}_{\cdot h}^{(w)}||_2 \qquad \text{or} \qquad \phi(\boldsymbol{\Lambda}^{(w)}) = \sum_{h=1}^{k} ||\boldsymbol{\lambda}_{\cdot h}^{(w)}||_\infty.$$

The ℓ_1 norm penalty achieves observation-specific column-wise shrinkage. Both of these mixed-norm penalties create a bi-convex problem in $\boldsymbol{\Lambda}$ and $\boldsymbol{X}$.

These two approaches of adaptive structured regularization in GFA models capture covariance uniquely shared among arbitrary subsets of the observations and avoid modeling shared covariance in non-maximal subsets. But neither the ARD approach nor the mixed-norm penalties encourages element-wise sparsity within loading columns. Adding element-wise sparsity is important because it results in interpretable latent factors, where features with non-zero loadings in a specific factor have an interpretation as a cluster (West, 2003; Carvalho et al., 2008). To induce element-wise sparsity, one can either use Bayesian shrinkage on each loading (Carvalho et al., 2010) or a mixed norm with ℓ_1 type penalties on each element (i.e., $\sum_{h=1}^{k}\sum_{j=1}^{p}|\lambda_{jh}^{(w)}|$).

A more recent GFA model is a step toward both column-wise and element-wise sparsity (Khan et al., 2014). In this model, element-wise sparsity is achieved by putting independent ARD priors on each loading element, and column-wise sparsity is achieved by a spike-and-slab prior on the loading columns. However, ARD priors do not allow the model to adjust shrinkage levels within each factor, and this approach does not include sparse and dense factors. One contribution of our work is to define a carefully structured Bayesian shrinkage prior on the loading matrix of a GFA model that encourages both element-wise and column-wise shrinkage, and that includes both sparse and dense factors.

3. Bayesian structured sparsity

The column-wise sparse structure of $\boldsymbol{\Lambda}$ in GFA models belongs to a general class of structured sparsity methods that has drawn attention recently (Zou and Hastie, 2005; Yuan and Lin, 2006; Jenatton et al., 2011, 2010; Kowalski, 2009; Kowalski and Torrésani, 2009; Zhao et al., 2009; Huang et al., 2011; Jia et al., 2010). For example, in structured sparse PCA, the loading matrix is constrained to have specific patterns (Jenatton et al., 2010). Later work discussed more general structured variable selection methods in a regression framework (Jenatton et al., 2011; Huang et al., 2011). However, there has been little work in using Bayesian structured sparsity, with some exceptions (Kyung et al., 2010; Engelhardt and Adams, 2014; Wu et al., 2014). Starting from Bayesian sparse priors, we propose a structured hierarchical sparse prior that includes three levels of shrinkage, which is conceptually similar to tree structured shrinkage (Romberg et al., 2001), or global-local priors in the regression framework (Polson and Scott, 2011).

3.1 Bayesian sparsity-inducing priors

Bayesian shrinkage priors have been widely used in latent factor models due to their flexible and interpretable solutions (West, 2003; Carvalho et al., 2008; Polson and Scott, 2011; Knowles and Ghahramani, 2011; Bhattacharya and Dunson, 2011). In Bayesian statistics, a regularizing term, $\phi(\mathbf{\Lambda})$, may be viewed as a marginal prior proportional to $\exp(-\phi(\mathbf{\Lambda}))$; the regularized optimum then becomes the maximum a posteriori (MAP) solution (Polson and Scott, 2011). For example, the well known ℓ_2 penalty for coefficients in linear regression models corresponds to Gaussian priors, also known as ridge regression or Tikhonov regularization (Hoerl and Kennard, 1970). In contrast, an ℓ_1 penalty corresponds to double exponential or Laplace priors, also known as the Bayesian Lasso (Tibshirani, 1996; Park and Casella, 2008; Hans, 2009).

When the goal of regularization is to induce sparsity, the prior distribution should be chosen so that it has substantial probability mass around zero, which draws small effects toward zero, and heavy tails, which allows large signals to escape from substantial shrinkage (O'Hagan, 1979; Carvalho et al., 2010; Armagan et al., 2011). The canonical Bayesian sparsity-inducing prior is the spike-and-slab prior, which is a mixture of a point mass at zero and a flat distribution across the space of real values, often modeled as a Gaussian with a large variance term (Mitchell and Beauchamp, 1988; West, 2003). The spike-and-slab prior has elegant interpretability by estimating the probability that certain loadings are excluded, modeled by the 'spike' distribution, or included, modeled by the 'slab' distribution (Carvalho et al., 2008). This interpretability comes at the cost of having exponentially many possible configurations of model inclusion parameters in the loading matrix.

Recently, scale mixtures of normal priors have been proposed as a computationally efficient alternative to the two component spike-and-slab prior (West, 1987; Carvalho et al., 2010; Polson and Scott, 2011; Armagan et al., 2013, 2011; Bhattacharya et al., 2014). Such priors generally assume normal distributions with a mixed variance term. The mixing distribution of the variance allows strong shrinkage near zero but weak regularization away from zero. For example, an inverse gamma distribution on the variance term results in an ARD prior (Tipping, 2001), and an exponential distribution on the variance term results in a Laplace prior (Park and Casella, 2008). The horseshoe prior, with a half Cauchy distribution on the standard deviation as the mixing density, has become popular due to its strong shrinkage and heavy tails (Carvalho et al., 2010).

A more general class of beta mixtures of normals is the three parameter beta distribution (Armagan et al., 2011). Although these continuous shrinkage priors do not directly model the probability of feature inclusion, it has been shown in the regression framework that two layers of regularization—global regularization, across all coefficients, and local regularization, specific to each coefficient (Polson and Scott, 2011))—has behavior that is similar to the spike-and-slab prior in effectively modeling signal and noise separately, but with computational tractability (Carvalho et al., 2009). In this study, we extend and structure the beta mixture of normals prior to three levels of hierarchy to induce desirable behavior in the context of GFA models.

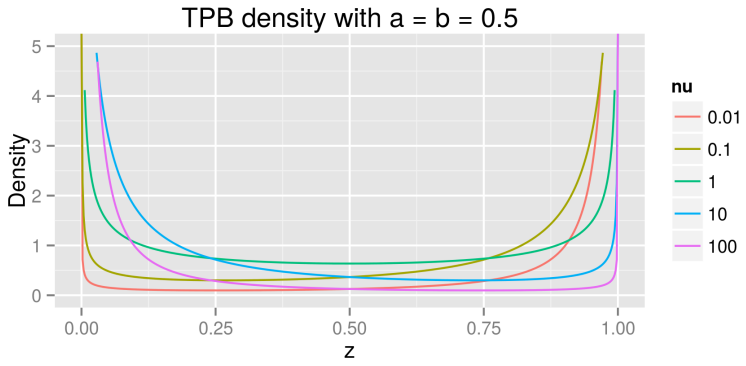

Figure 2: **Density of the three parameter beta ($\mathcal{TPB}$) distribution with different values of ν.** Five different values of $\nu = \{0.01, 0.1, 1, 10, 100\}$ for the three parameter beta distribution with $a = b = 0.5$. The x-axis represents the value of random variable z, and the y-axis represents the density of random variable z.

3.2 Three parameter beta prior

The three parameter beta ($\mathcal{TPB}$) distribution for a random variable $Z \in (0, 1)$ has the following density (Armagan et al., 2011):

$$f(z; a, b, \nu) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \nu^b z^{b-1} (1-z)^{a-1} \{1 + (\nu - 1)z\}^{-(a+b)}, \tag{7}$$

where $a, b, \phi > 0$. We denote this distribution as $\mathcal{TPB}(a, b, \nu)$. When $0 < a < 1$ and $0 < b < 1$, the distribution is bimodal, with modes at 0 and 1 (Figure 2). The variance parameter ν gives the distribution freedom: with fixed a and b, smaller values of ν put greater probability on $z = 1$, while larger values of ν move the probability mass towards $z = 0$ (Armagan et al., 2011). With $\nu = 1$, this distribution is identical to a beta distribution (i.e., $Be(b, a)$).

Let λ denote the parameter to which we are applying sparsity-inducing regularization. We assign the following $\mathcal{TPB}$ normal scale mixture distribution, $\mathcal{TPBN}$, to λ:

$$\lambda | \varphi \sim \mathcal{N}\left(0, \frac{1}{\varphi} - 1\right), \qquad \text{with} \qquad \varphi \sim \mathcal{TPB}(a, b, \nu),$$

where the *shrinkage parameter* φ follows a $\mathcal{TPB}$ distribution. With $a = b = 1/2$ and $\nu = 1$, this prior becomes the horseshoe prior (Carvalho et al., 2010; Armagan et al., 2011; Gao et al., 2013). The bimodal property of φ induces two distinct shrinkage behaviors: the mode near one encourages $\frac{1}{\varphi} - 1$ towards zero and induces strong shrinkage on λ; the mode near zero encourages $\frac{1}{\varphi} - 1$ large, creating a diffuse prior on λ. Further decreasing the variance parameter ν supports stronger shrinkage (Armagan et al., 2011; Gao et al., 2013). If we let $\theta = \frac{1}{\varphi} - 1$, then this mixture has the following hierarchical representation:

$$\lambda \sim \mathcal{N}(0, \theta), \quad \theta \sim Ga(a, \delta), \quad \delta \sim Ga(b, \nu).$$

Note the difference between the ARD prior and the $\mathcal{TPB}$: the ARD prior induces sparsity using an inverse gamma prior on θ, whereas the $\mathcal{TPB}$ induces sparsity by using a gamma prior on the θ variable and then regularizing the rate parameter δ using a second gamma prior. These differences lead to different behavior of ARD and the $\mathcal{TPB}$ in theory (Polson and Scott, 2011) and in practice, as we show below.

3.3 Global-factor-local shrinkage

The flexible representation of the $\mathcal{TPB}$ prior makes it an ideal choice for latent factor models. Our recent work extended the $\mathcal{TPB}$ prior to three levels of regularization on a loading matrix (Gao et al., 2013):

$$\begin{aligned}
\varrho &\sim \mathcal{TPB}(e, f, \nu), && \text{(Global)} \\
\zeta_h &\sim \mathcal{TPB}\left(c, d, \frac{1}{\varrho} - 1\right), && \text{(Factor-specific)} \\
\varphi_{jh} &\sim \mathcal{TPB}\left(a, b, \frac{1}{\zeta_h} - 1\right), && \text{(Local)} \\
\lambda_{jh} &\sim \mathcal{N}\left(0, \frac{1}{\varphi_{jh}} - 1\right). && (8)
\end{aligned}$$

At each of the three levels, a $\mathcal{TPB}$ distribution is used to induce sparsity via its estimated variance parameter (ν in Equation 7), which in turn is regularized using a $\mathcal{TPB}$ distribution. Specifically, the global shrinkage parameter ϱ applies strong shrinkage across the k columns of the loading matrix and jointly adjusts the support of column-specific parameter ζ_h, $h \in \{1, \ldots, k\}$ close to either zero or one. This can be interpreted as inducing sufficient shrinkage across loading columns to recover the number of factors supported by the observed data. In particular, when ζ_h is close to one, all elements of column h are close to zero, effectively removing the h^{th} component. When near zero, the factor-specific regularization parameter ζ_h adjusts the shrinkage applied to each element of the h^{th} loading column, estimating the column-wise shrinkage by borrowing strength across all elements (i.e., features) in that column. The local shrinkage parameter, φ_{jh}, creates element-wise sparsity in the loading matrix through a $\mathcal{TPBN}$. Three levels of shrinkage allow us to model both column-wise and element-wise shrinkage simultaneously, and give the model nonparametric behavior in the number of factors via model selection.

Equivalently, this global-factor-local shrinkage prior can be written as (Armagan et al., 2011; Gao et al., 2013):

$$\begin{aligned}
\text{Global} \quad & \begin{cases} \gamma \sim Ga(f, \nu), \\ \eta \sim Ga(e, \gamma)), \end{cases} \\
\text{Factor-specific} \quad & \begin{cases} \tau_h \sim Ga(d, \eta), \\ \phi_h \sim Ga(c, \tau_h), \end{cases} \\
\text{Local} \quad & \begin{cases} \delta_{jh} \sim Ga(b, \phi_h), \\ \theta_{jh} \sim Ga(a, \delta_{jh}), \end{cases} \\
& \lambda_{jh} \sim \mathcal{N}(0, \theta_{jh}). \qquad (9)
\end{aligned}$$

We further extend our prior to jointly model sparse and dense components by assigning to the local shrinkage parameter a two-component mixture distribution (Gao et al., 2013):

$$\theta_{jh} \sim \pi Ga(a, \delta_{jh}) + (1-\pi)\delta_{\phi_h}(\cdot), \tag{10}$$

where $\delta_{\phi_h}(\cdot)$ is the Dirac delta function centered at ϕ_h. The motivation for this two component mixture is that, in real applications such as the analysis of gene expression data, it has been shown that much of the variation in the observation is due to technical (e.g., batch, platform) or biological effects (e.g., sex, ethnicity), which impact a large number of features (Leek et al., 2010). Therefore, loadings corresponding to these effects will often not be sparse. A two-component mixture (Equation 10) allows the prior on the loading (Equation 8) to select between element-wise sparsity or column-wise sparsity. Element-wise sparsity is encouraged via the $\mathcal{TPBN}$ prior. Column-wise sparsity jointly regularizes each element of the column with a shared variance term: $\lambda_{jh} \sim \mathcal{N}\left(0, \frac{1}{\zeta_h} - 1\right)$. Modeling each element in a column using a shared regularized variance term has two possible behaviors: i) ζ_h in Equation (8) is close to 1 and the entire column is shrunk towards zero, effectively removing this factor; ii) ζ_h is close to zero, and all elements of the column have a shared Gaussian distribution, inducing only non-zero elements in that loading. We call included factors that have only non-zero elements *dense factors*.

Jointly modeling sparse and dense factors effectively combines low-rank covariance factorization with interpretability (Zou et al., 2006; Parkhomenko et al., 2009). The dense factors capture the broad effects of observation confounders, model a low-rank approximation of the covariance matrix, and usually account for a large proportion of variance explained (Chandrasekaran et al., 2011). The sparse factors, on the other hand, capture the small groups of interacting features in a (possibly) high-dimensional sparse space, and usually account for a small proportion of the variance explained.

We introduce indicator variables z_h, $h = 1, \ldots, k$, to indicate which mixture component each θ_{jh} is generated from in Equation (10), where $z_h = 1$ means $\theta_{jh} \sim Ga(a, \delta_{jh})$ and $z_h = 0$ means $\theta_{jh} \sim \delta_{\phi_h}(\cdot)$. Thus, a component is a sparse factor when $z_h = 1$ and either a dense factor or eliminated when $z_h = 0$. We let $\boldsymbol{z} = [z_1, \ldots, z_k]$ and put a Bernoulli distribution with parameter π on z_h. We further let π have a flat beta distribution $Be(1, 1)$. This construct allows us to quantify the posterior probability that each factor h is generated from each mixture component type via z_h.

4. Bayesian group factor analysis with structured sparsity

In this work, we use global-factor-local $\mathcal{TPB}$ priors in the GFA model to enable both element-wise and column-wise shrinkage. Specifically, we put a $\mathcal{TPB}$ prior independently on each loading matrix corresponding to the w^{th} observation, $\boldsymbol{\Lambda}^{(w)}$. Let $\boldsymbol{Z} = [\boldsymbol{z}^{(1)}; \ldots; \boldsymbol{z}^{(m)}] \in \mathbb{R}^{m \times k}$. The indicator variable $z_h^{(w)}$ is associated with the h^{th} factor and specific to observation w. When $z_h^{(w)} = 1$, the h^{th} factor has a sparse loading for observation w; when $z_h^{(w)} = 0$, then either the h^{th} factor has a dense loading column for observation w, or observation w is not represented in that loading column. A zero loading column for observation w effectively decouples the factor from that observation, leading to the column-wise sparse behavior in previous GFA models (Virtanen et al., 2012; Klami et al., 2014b). In our model, factors

that include no observations in the associated loading column are removed from the model. We refer to this model as Bayesian group factor Analysis with Structured Sparsity (*BASS*).

We summarize BASS as follows. The generative model for m coupled observations $\boldsymbol{y}_i^{(w)}$ with $w = 1, \ldots, m$ and $i = 1, \ldots, n$ is

$$\boldsymbol{y}_i^{(w)} = \boldsymbol{\Lambda}^{(w)} \boldsymbol{x}_i + \boldsymbol{\epsilon}_i^{(w)}, \quad \text{for} \quad w = 1, \ldots, m.$$

This model is written as a latent factor model by concatenating the m feature vectors into vector $\boldsymbol{y}_i$

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{\Lambda} \boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \\ \boldsymbol{x}_i &\sim \mathcal{N}_k(0, \boldsymbol{I}_k), \\ \boldsymbol{\epsilon}_i &\sim \mathcal{N}_p(0, \boldsymbol{\Sigma}), \end{aligned} \tag{11}$$

where $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$ and $p = \sum_{w=1}^m p_w$. We put independent global-factor-local $\mathcal{TPB}$ priors (Equation 9) on $\boldsymbol{\Lambda}^{(w)}$:

$$\begin{aligned} \text{Global} \quad & \begin{cases} \gamma^{(w)} \sim Ga(f, \nu), \\ \eta^{(w)} \sim Ga(e, \gamma^{(w)})), \end{cases} \\ \text{Factor-specific} \quad & \begin{cases} \tau_h^{(w)} \sim Ga(d, \eta^{(w)}), \\ \phi_h^{(w)} \sim Ga(c, \tau_h^{(w)}), \end{cases} \\ \text{Local} \quad & \begin{cases} \delta_{jh}^{(w)} \sim Ga(b, \phi_h^{(w)}), \\ \theta_{jh}^{(w)} \sim Ga(a, \delta_{jh}^{(w)}), \end{cases} \\ & \lambda_{jh}^{(w)} \sim \mathcal{N}(0, \theta_{jh}^{(w)}). \end{aligned}$$

We allow local shrinkage to follow a two-component mixture

$$\theta_{jh}^{(w)} \sim \pi^{(w)} Ga(a, \delta_{jh}^{(w)}) + (1 - \pi^{(w)}) \delta_{\phi_h^{(w)}}(\cdot),$$

where the mixture proportion has a beta distribution

$$\pi^{(w)} \sim Be(1, 1).$$

We put a conjugate inverse gamma distribution on the residual variance parameters

$$\sigma_j^{-2} \sim Ga(a_\sigma, b_\sigma).$$

In our application of BASS, we set the hyperparameters of the global-factor-local $\mathcal{TPB}$ prior to $a = b = c = d = e = f = 0.5$, which recapitulates the horseshoe prior at all three levels of the hierarchy. The hyperparameters for the error variances, a_σ and b_σ, were set to 1 and 0.3 respectively to allow a relatively wide support of variances (Bhattacharya and Dunson, 2011). When there are two coupled observations, the BASS framework is a Bayesian CCA model (Equation 4) based on its column-wise shrinkage.

5. Parameter estimation

Given our setup, the full joint distribution of the BASS model factorizes as

$$\begin{aligned} p(\boldsymbol{Y}, \boldsymbol{X}, &\boldsymbol{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{Z}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \\ &= p(\boldsymbol{Y}|\boldsymbol{\Lambda}, \boldsymbol{X}, \boldsymbol{\Sigma}) p(\boldsymbol{X}) \\ &\quad \times p(\boldsymbol{\Lambda}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\boldsymbol{\Delta}, \boldsymbol{Z}, \boldsymbol{\Phi}) p(\boldsymbol{\Delta}|\boldsymbol{\Phi}) p(\boldsymbol{\Phi}|\boldsymbol{T}) p(\boldsymbol{T}|\boldsymbol{\eta}) p(\boldsymbol{\eta}|\boldsymbol{\gamma}) \\ &\quad \times p(\boldsymbol{\Sigma}) p(\boldsymbol{Z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}), \end{aligned}$$

where $\boldsymbol{\Theta} = \{\theta_{jh}^{(w)}\}$, $\boldsymbol{\Delta} = \{\delta_{jh}^{(w)}\}$, $\boldsymbol{\Phi} = \{\phi_h^{(w)}\}$, $\boldsymbol{T} = \{\tau_h^{(w)}\}$, $\boldsymbol{\eta} = \{\eta^{(w)}\}$ and $\boldsymbol{\gamma} = \{\gamma^{(w)}\}$ are the collections of the global-factor-local $\mathcal{TPB}$ prior parameters. The posterior distributions of model parameters may be either simulated through Markov chain Monte Carlo (MCMC) methods or approximated using variational Bayes approaches. We derive an MCMC algorithm based on a Gibbs sampler (Appendix A). The MCMC algorithm updates the joint loading matrix row by row using block updates, enabling relatively fast mixing (Bhattacharya and Dunson, 2011).

In many applications, we are interested in a single point estimate of the parameters instead of the complete posterior estimate; thus, often an expectation maximization (EM) algorithm is used to find a *maximum a posteriori* (MAP) estimate of model parameters using conjungate gradient optimization (Dempster et al., 1977). In EM, the latent factors $\boldsymbol{X}$ and the indicator variables $\boldsymbol{Z}$ are treated as missing data and their expectations estimated in the E-step conditioned on the current values of the parameters; then the model parameters are optimized in the M-step conditioning on the current expectations of the latent variables. Let $\boldsymbol{\Xi} = \{\boldsymbol{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\pi}, \boldsymbol{\Sigma}\}$ be the collection of the parameters optimized in the M-step. The expected complete log likelihood, denoted $Q(\cdot)$, may be written as

$$Q(\boldsymbol{\Xi}|\boldsymbol{\Xi}_{(s)}) = \mathbb{E}_{\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Xi}_{(s)}, \boldsymbol{Y}} \left[\log \left(p(\boldsymbol{\Xi}, \boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{Y}) \right) \right].$$

Since $\boldsymbol{X}$ and $\boldsymbol{Z}$ are conditionally independent given $\boldsymbol{\Xi}$, the expectation may be calculated using the full conditional distributions of $\boldsymbol{X}$ and $\boldsymbol{Z}$ derived for the MCMC algorithm. The derivation of the EM algorithm for BASS is then straightforward (Appendix B); note that, when estimating $\boldsymbol{\Lambda}$, the loading columns specific to each observation are estimated jointly.

5.1 Identifiability

The latent factor model (Equation 1) is identifiable up to orthonormal rotations: for any orthogonal matrix $\boldsymbol{P}$ with $\boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}$, letting $\boldsymbol{\Lambda}' = \boldsymbol{\Lambda}\boldsymbol{P}^T$ and $\boldsymbol{x}' = \boldsymbol{P}\boldsymbol{x}$ produces the same estimate of the data covariance matrix and has an identical likelihood. When using factor analysis for prediction or covariance estimation, rotational invariance is irrelevant. However, for all applications that interpret the factors or use individual factors or loadings for downstream analysis, this rotational invariance cannot be ignored. One traditional solution is to restrict the loading matrix to be lower triangular (West, 2003; Carvalho et al., 2008). This solution gives a special role to the first $k-1$ features in $\boldsymbol{y}$, namely, that the h^{th} feature does not contribute to the $k - h^{th}$ through the k^{th} factor. For this reason, the lower triangular approach does not generalize easily and requires domain knowledge that may not be available (Carvalho et al., 2008).

In the BASS model, we have rotational invariance when we right multiply the joint loading matrix by $\boldsymbol{P}^T$ and left multiply $\boldsymbol{x}$ by $\boldsymbol{P}$, producing an identical covariance matrix and likelihood. This rotation invariance is addressed in BASS because the non-sparse rotations of the loading matrix violates the prior structure induced by the observation-wise and element-wise sparsity.

Scale invariance is a second identifiability problem inherent in latent factor models. In particular, scale invariance means that a loading can be multiplied by a non-zero constant and the corresponding factor by the inverse of that constant, and this will result in the same data likelihood. This problem we and others have addressed satisfactorily by using posterior probabilities as optimization objectives instead of likelihoods and by including regularizing priors on the factors that restrict the magnitude of the constant. We make an effort to not interpret the relative or absolute scale of the factors or loadings including sign beyond setting a reasonable threshold for zero.

Finally, factor analysis is identifiable up to *label switching*, or shuffling the $h = 1, \ldots, k$ indices of the loadings and factors, assuming we do not take the lower triangular approach. Other approaches put distributions on the loading sparsity or proportion of variance explained in order to address this problem (Bhattacharya and Dunson, 2011). We do not explicitly order or interpret the order of the factors, so we do not address this non-identifiability in the model. Label switching is handled here and elsewhere by a post-processing step, such as ordering factors according to proportion of variance explained. In our simulation studies, we interpret results with this non-identifiability in mind.

5.2 Sparse rotations via PX-EM

Another general problem with latent factor models, including BASS, is the convergence to local optima and sensitivity to parameter initializations. Once the model parameters are initialized, the EM algorithm may be stuck in locally optimal but globally suboptimal regions with undesirable factor orientations. To address this problem, we take advantage of the rotational invariance of the factor analysis framework. Parameter expansion (PX) has been shown to reduce the initialization dependence by introducing auxiliary variables that rotate the current estimate of the loading matrix to best respect the prior while keeping the likelihood stable (Liu et al., 1998; Dyk and Meng, 2001).

We extend our model (Equation 11) using parameter expansion $\boldsymbol{R}$, a positive definite $k \times k$ matrix, as

$$\begin{aligned}
\boldsymbol{y}_i &= \boldsymbol{\Lambda}\boldsymbol{R}_L^{-1}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \\
\boldsymbol{x}_i &\sim \mathcal{N}_k(\boldsymbol{0}, \boldsymbol{R}), \\
\boldsymbol{\epsilon}_i &\sim \mathcal{N}_k(\boldsymbol{0}, \boldsymbol{\Sigma}),
\end{aligned}$$

where $\boldsymbol{R}_L$ is the lower triangular matrix of the Cholesky decomposition of $\boldsymbol{R}$. The covariance of $\boldsymbol{y}_i$ is invariant under this expansion, and, correspondingly, the likelihood is stable. Note $\boldsymbol{R}_L^{-1}$ is not an orthogonal matrix; however, because it is full rank, it can be transformed into an orthogonal matrix times a rotation matrix via a polar decomposition (Rocková and George, 2015). We let $\boldsymbol{\Lambda}^\star = \boldsymbol{\Lambda}\boldsymbol{R}_L^{-1}$ and assign our BASS $\mathcal{TPBN}$ prior to this *rotated* loading matrix.

We let $\Xi^\star = \{\Lambda^\star, \Theta, \Delta, \Phi, T, \eta, \gamma, \pi, \Sigma\}$, and the parameters of our expanded model are $\{\Xi^\star \cup R\}$. The EM algorithm in this expanded parameter space generates a sequence of parameter estimates $\{{\Xi^\star}_{(1)} \cup R_{(1)}, {\Xi^\star}_{(2)} \cup R_{(2)}, \dots\}$, which corresponds to a sequence of parameter estimates in the original space $\{\Xi_{(1)}, \Xi_{(2)}, \dots\}$, where Λ is recovered via $\Lambda^\star R_L$ (Rocková and George, 2015). We initialize $R_{(0)} = I_k$. The expected complete log likelihood of this PX BASS model is

$$Q(\Xi^\star, R|\Xi_{(s)}) = \mathbb{E}_{X,Z|\Xi_{(s)},Y,R_0} \log\big(p(\Xi^\star, R, X, Z|Y)\big). \tag{12}$$

In our parameter-expanded EM (PX-EM) for BASS, the conditional distributions of X and Z still factorize in the expectation. However, the distribution of x_i depends on expansion parameter R. The full joint distribution (Equation 11) has a single change in $p(X)$, with $\Lambda^\star$ in the place of Λ. In the M-step, the R that maximizes Equation (12) is

$$R_{(s)} = \arg\max_R Q(\Xi^\star, R|\Xi_{(s)}) = \arg\max_R \left(\text{const} - \frac{n}{2}\log|R| - \frac{1}{2}\text{tr}\big(R^{-1}S^{XX}\big)\right),$$

where $S^{XX} = \sum_{i=1}^n \langle x_{\cdot i} x_{\cdot i}^T \rangle$. The solution is $R_{(s)} = \frac{1}{n} S^{XX}$. For the E-step, Λ is first calculated and the expectation is taken in the original space (details in Appendix C).

Note that the proposed PX-EM for the BASS model keeps the likelihood invariant but does not keep the prior invariant after transformation of Λ. This is different from the earlier PX-EM algorithm (Liu et al., 1998), as discussed in recent work (Rocková and George, 2015). Because the resulting posterior is not invariant, we run PX-EM only for a few iterations and then switch to the EM algorithm. The effect is that the BASS model is substantially less sensitive to initialization (see simulation results). By introducing expansion parameter R, the posterior modes in the original space are intersected with equal likelihood curves indexed by R in expanded space. Those curves facilitate traversal between posterior modes in the original space and encourage initial parameter estimates with appropriate sparse structure in the loading matrix (Rocková and George, 2015).

5.3 Computational complexity

The computational complexity of the block Gibbs sampler for the BASS model is demanding. Updating each loading row requires the inversion of a $k \times k$ matrix with $O(k^3)$ complexity and then calculating means with $O(k^2 n)$ complexity. The complexity of updating the full loading matrix repeats this calculation p times. Other updates are of lower order relative to updating the loading. Our Gibbs sampler has $O(k^3 p + k^2 pn)$ complexity per iteration, which makes MCMC difficult to apply when p is large.

In the BASS EM algorithm, the E-step has complexity $O(k^3)$ for a matrix inversion, complexity $O(k^2 p + kpn)$ for calculating the first moment, and complexity $O(k^2 n)$ for calculating the second moment. Calculations in the M-step are all of a lower order. Thus, the EM algorithm has complexity $O(k^3 + k^2 p + k^2 n + kpn)$ per iteration.

Our PX-EM algorithm for the BASS model requires an additional Cholesky decomposition with complexity $O(k^3)$ and a matrix multiplication with complexity $O(k^2 p)$ above the EM algorithm. The total complexity is therefore the same as the original EM algorithm, although in practice we note that the constants have a negative impact on the running time.

6. Simulations and comparisons

We demonstrate the performance of our model on simulated data in three settings: paired observations, four observations, and ten observations.

6.1 Simulations

We describe the details of the three types of simulations here.

6.1.1 Simulations with paired observations (CCA)

We simulated two data sets with $p_1 = 100$, $p_2 = 120$ in order to compare results from our method to results from state-of-the-art CCA methods. The number of samples in these simulations was $n = \{20, 30, 40, 50\}$, chosen to be smaller than both p_1 and p_2 to reflect the large p, small n regime (West, 2003) that motivated our structured approach. We first simulated observations with only sparse latent factors (*Sim1*). In particular, we set $k = 6$, where two sparse factors are shared by both observations (factors 1 and 2; Table 1), two sparse factors are specific to $\boldsymbol{y}^{(1)}$ (factors 3 and 4; Table 1), and two sparse factors are specific to $\boldsymbol{y}^{(2)}$ (factors 5 and 6; Table 1). The elements in the sparse loading matrix were randomly generated from a $\mathcal{N}(0, 4)$ Gaussian distribution, and sparsity was induced by setting 90% of the elements in each loading column to zero at random (Figure 3A). We zeroed values of the sparse loadings for which the absolute values were less than 0.5. Latent factors $\boldsymbol{x}$ were generated from $\mathcal{N}_6(0, \boldsymbol{I}_6)$. Residual error was generated by first generating the $p = p_1 + p_2$ diagonals on the residual covariance matrix $\boldsymbol{\Sigma}$ from a uniform distribution on $(0.5, 1.5)$, and then generating each column of the error matrix from $\mathcal{N}_p(\boldsymbol{0}, \boldsymbol{\Sigma})$.

We performed a second simulation that included both sparse and dense latent factors (*Sim2*). In particular, we extended *Sim1* to $k = 8$ latent factors, where one of the shared sparse factors is now dense, and two dense factors, each specific to one observation, were added. For all dense factors, each loading was generated according to a $\mathcal{N}(0, 4)$ Gaussian distribution (Table 1; Figure 3B).

	Sim1						*Sim2*							
Factors	1	2	3	4	5	6	1	2	3	4	5	6	7	8
$\boldsymbol{Y}^{(1)}$	S	S	S	S	-	-	S	D	S	S	D	-	-	-
$\boldsymbol{Y}^{(2)}$	S	S	-	-	S	S	S	D	-	-	-	S	S	D

Table 1: **Latent factors in *Sim1* and *Sim2* with two observation matrices.** S represents a sparse vector; D represents a dense vector; - represents no contribution to that observation from the factor.

6.1.2 Simulations with four observations (GFA)

We performed two simulations (*Sim3* and *Sim4*) including four observations with $p_1 = 70$, $p_2 = 60$, $p_3 = 50$ and $p_4 = 40$. The number of samples, as above, was set to $n = \{20, 30, 40, 50\}$. In *Sim3*, we let $k = 6$ and only simulated sparse factors: the first three factors were specific to $\boldsymbol{y}^{(1)}$, $\boldsymbol{y}^{(2)}$ and $\boldsymbol{y}^{(3)}$, respectively, and the last three corresponded to different subsets of the observations (Table 2). In *Sim4* we let $k = 8$, and, as with *Sim2*,

	Sim3						*Sim4*							
Factors	1	2	3	4	5	6	1	2	3	4	5	6	7	8
$\boldsymbol{Y}^{(1)}$	S	-	-	S	-	-	S	-	-	-	D	-	-	-
$\boldsymbol{Y}^{(2)}$	-	S	-	S	S	S	-	S	-	S	-	D	-	-
$\boldsymbol{Y}^{(3)}$	-	-	S	-	S	S	-	-	S	S	-	-	D	-
$\boldsymbol{Y}^{(4)}$	-	-	-	-	-	S	-	-	S	-	-	-	-	D

Table 2: **Latent factors in *Sim3* and *Sim4* with four observation matrices.** S represents a sparse vector; D represents a dense vector; - represents no contribution to that observation from the factor.

	Sim5								*Sim6*									
Factors	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	9	10
$\boldsymbol{Y}^{(1)}$	S	-	-	-	-	-	-	-	S	-	-	-	-	-	D	-	-	-
$\boldsymbol{Y}^{(2)}$	S	-	-	S	-	-	-	-	S	-	-	S	-	-	D	-	-	-
$\boldsymbol{Y}^{(3)}$	S	-	-	S	S	-	-	-	-	-	-	S	-	-	D	D	-	-
$\boldsymbol{Y}^{(4)}$	S	S	-	S	S	-	S	-	-	S	-	S	-	-	D	D	-	-
$\boldsymbol{Y}^{(5)}$	-	S	-	S	S	-	S	-	-	S	-	S	S	-	-	D	D	-
$\boldsymbol{Y}^{(6)}$	-	S	-	-	-	-	S	S	-	S	-	-	S	-	-	D	D	-
$\boldsymbol{Y}^{(7)}$	-	-	S	-	-	-	S	S	-	S	S	-	S	-	-	-	D	D
$\boldsymbol{Y}^{(8)}$	-	-	S	-	-	-	S	S	-	-	S	-	S	-	-	-	D	D
$\boldsymbol{Y}^{(9)}$	-	-	S	-	-	-	-	S	-	-	S	-	-	-	-	-	-	D
$\boldsymbol{Y}^{(10)}$	-	-	S	-	-	S	-	-	-	-	S	-	-	S	-	-	-	D

Table 3: **Latent factors in *Sim5* and *Sim6* with four observation matrices.** S represents a sparse vector; D represents a dense vector; - represents no contribution to that observation from the factor.

included both sparse and dense factors (Table 2). Samples from these two simulations were generated following the same procedure as the simulations with two observations.

6.1.3 Simulations with ten observations (GFA)

To further evaluate BASS on multiple observations, we performed two additional simulations (*Sim5* and *Sim6*) on ten coupled observations with $p_w = 50$ for $w = 1, \ldots, 10$. The number of samples was set to $n = \{20, 30, 40, 50\}$. In *Sim5*, we let $k = 8$ and only simulated sparse factors (Table 3). In *Sim6* we let $k = 10$ and simulated both sparse and dense factors (Table 3). Samples in these two simulations were generated following the same method as in the simulations with two observations.

6.2 Methods for comparison

We compared BASS to five available linear models that accept multiple observations: the Bayesian group factor analysis model with an ARD prior (GFA) (Klami et al., 2013), an extension of GFA that allows element-wise sparsity with independent ARD priors (sGFA) (Khan et al., 2014; Suvitaival et al., 2014), a regularized version of CCA (RCCA) (González et al., 2008), sparse CCA (SCCA) (Witten and Tibshirani, 2009), and Bayesian joint factor

analysis (JFA) (Ray et al., 2014). We also included the linear version of a flexible non-linear model, manifold relevance determination (MRD) (Damianou et al., 2012). To evaluate the sensitivity of BASS to initialization, we compared three different initialization methods: random initialization (EM), 50 iterations of MCMC (MCMC-EM), and 20 iterations of PX-EM (PX-EM); each of these were followed with EM until convergence, reached when both the number of non-zero loadings do not change for t iterations and the log likelihood changes $< 1 \times 10^{-5}$ within t iterations. We performed 20 runs for each version of inference in BASS: EM, MCMC-EM, and PX-EM. In *Sim1* and *Sim3*, we set the initial number of factors to $k = 10$. In *Sim2*, *Sim4*, *Sim5*, and *Sim6*, we set the initial number of factors to 15.

The GFA model (Klami et al., 2013) uses an ARD prior to encourage column-wise shrinkage of the loading matrix, but not sparsity within the loadings. The computational complexity of this GFA model with variational updates is $O(k^3m + k^2p + k^2n + kpn)$ per iteration, which is nearly identical to BASS but includes an additional factor m, the number of observations, scaling the k^3 term. In our simulations, we ran the GFA model with the factor number set to the correct value.

The sGFA model (Khan et al., 2014) encourages element-wise sparsity using independent ARD priors on loading elements. Loading columns are modeled with a spike-and-slab type mixture to encourage column-wise sparsity. Inference is performed with a Gibbs sampler without using block updates. Its complexity is $O(k^3 + k^2pn)$ per iteration, which, when k is large, will dominate the per-iteration complexity of BASS; furthermore, Gibbs samplers typically require greater numbers of iterations than EM-based methods. We ran the sGFA model with the correct number of factors in our six simulations.

We ran the regularized version of classical CCA (RCCA) for comparison in *Sim1* and *Sim2* (González et al., 2008). Classical CCA tries to find k canonical projection directions $\boldsymbol{u}_h$ and $\boldsymbol{v}_h$ ($h = 1, \ldots, k$) for $\boldsymbol{Y}^{(1)}$ and $\boldsymbol{Y}^{(2)}$ respectively such that i) the correlation between $\boldsymbol{u}_h^T\boldsymbol{Y}^{(1)}$ and $\boldsymbol{v}_h^T\boldsymbol{Y}^{(2)}$ is maximized for $h = 1, \ldots, k$; and ii) $\boldsymbol{u}_{h'}^T\boldsymbol{Y}^{(1)}$ is orthogonal to $\boldsymbol{u}_h^T\boldsymbol{Y}^{(1)}$ with $h' \neq h$, and similarly for $\boldsymbol{v}_h$ and $\boldsymbol{Y}^{(2)}$. Let these two projection matrices be denoted $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k] \in \mathbb{R}^{p_1 \times k}$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k] \in \mathbb{R}^{p_2 \times k}$. These matrices are the maximum likelihood estimates of the shared loading matrices in the Bayesian CCA model up to orthogonal transformations (Bach and Jordan, 2005). However, classical CCA requires the observation covariance matrices to be non-singular and thus is not applicable in the current simulations where $n < p_1, p_2$.

Here, we used a regularized version of CCA (RCCA) (González et al., 2008), which regularizes CCA using an ℓ_2-type penalty by adding $\lambda_1\boldsymbol{I}_{p_1}$ and $\lambda_2\boldsymbol{I}_{p_2}$ to the two sample covariance matrices. The effect of this penalty is not to induce sparsity but instead to allow application to $p \gg n$ data sets. The two regularization parameters (λ_1 and λ_2) were chosen according to leave-one-out cross-validation with the search space defined on a 11×11 grid from 0.0001 to 0.01. The projection directions $\boldsymbol{U}$ and $\boldsymbol{V}$ were estimated using the best regularization parameters. We let $\boldsymbol{\Lambda}' = [\boldsymbol{U}; \boldsymbol{V}]$; this matrix was comparable to the simulated loading matrix up to orthogonal transformations. We calculated the matrix $\boldsymbol{P}$ such that the Frobenius norm between $\boldsymbol{\Lambda}'\boldsymbol{P}^T$ and simulated $\boldsymbol{\Lambda}$ was minimized, with the constraint that $\boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}$. This was done by the constraint-preserving updates of the objective function (Wen and Yin, 2013). After finding the optimal orthogonal transformation matrix, we recovered $\boldsymbol{\Lambda}'\boldsymbol{P}^T$ as the estimated loading matrix. We set the number of projections to

6 and 8 in *Sim1* and *Sim2*, respectively, representing the true number of latent factors. RCCA does not apply to multiple coupled observations, and therefore it was not included in further simulations.

The sparse CCA (SCCA) method (Witten and Tibshirani, 2009) maximizes correlation between two observations after projecting the original space with a sparsity-inducing penalty onto the latent components, producing sparse matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. This method is encoded in the R package `PMA` (Witten et al., 2013). For *Sim1* and *Sim2*, as with RCCA, we found an optimal orthogonal transformation matrix $\boldsymbol{P}$ such that the Frobenius norm between $\boldsymbol{\Lambda}_S\boldsymbol{P}^T$ and simulated $\boldsymbol{\Lambda}$ was minimized, where $\boldsymbol{\Lambda}_S$ was the vertical concatenation of the recovered sparse $\boldsymbol{U}$ and $\boldsymbol{V}$. We chose 6 and 8 sparse projections in *Sim1* and *Sim2*, respectively, representing the true number of linear factors. Because both RCCA and SCCA are both deterministic and greedy, the results for $k < 6$ are all implicitly available by subsetting the factors in the $k = 6$ results.

An extension of SCCA allows for multiple observations (Witten and Tibshirani, 2009). For *Sim3* and *Sim4*, we recovered four sparse projection matrices $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \boldsymbol{U}^{(3)}, \boldsymbol{U}^{(4)}$, and for *Sim5* and *Sim6*, we recovered ten projection matrices. $\boldsymbol{\Lambda}_S$ was calculated with the concatenation of those projection matrices. Then the orthogonal transformation matrix $\boldsymbol{P}$ was calculated similarly by minimizing the Frobenius norm between $\boldsymbol{\Lambda}_S\boldsymbol{P}^T$ and the true loading matrix $\boldsymbol{\Lambda}$. The number of canonical projections was set to 6 in *Sim3*, 8 in *Sim4* and *Sim5*, and 10 in *Sim6*, corresponding to the true number of latent factors.

The Bayesian joint factor analysis model (JFA) (Ray et al., 2014) puts an Indian buffet process (IBP) prior (Griffiths and Ghahramani, 2011) on the factors, inducing element-wise sparsity, and an ARD prior on the variance of the loadings. The idea of putting an IBP on a latent factor model, which gives desirable nonparametric behavior in the number of latent factors and also produces element-wise sparsity in the loading matrix, was described for the Nonparametric Sparse Factor Analysis (NSFA) model (Knowles and Ghahramani, 2011). Similarly, in JFA, element-wise sparsity is encouraged both in the factors and in the loadings. JFA partitions latent factors into a fixed number of observation-specific factors and factors shared by all observations, and does not include column-wise sparsity. Its complexity is $O(k^3 + k^2pn)$ per iteration of the Gibbs sampler. We ran JFA on our simulations with the number of factors set to the correct values. Because the JFA model uses a sparsity-inducing prior instead of an independent Gaussian prior on the latent factors, the resulting model does not have a closed form posterior predictive distribution (Equation 13); therefore, we excluded the JFA model from prediction results.

The non-linear manifold relevance determination (MRD) model (Damianou et al., 2012) extends the notable Gaussian process latent variable (GPLVM) model (Lawrence, 2005) to include multiple observations. A GPLVM puts a Gaussian process prior on the latent variable space. GPLVM has an interpretation of a dual probabilistic PCA model that marginalizes loading columns using Gaussian priors. MRD extends GPLVM by putting multiple weight vectors on the latent variables using a Gaussian process kernel. Each of the weight vectors corresponds to one observation, therefore they determine a soft partition of latent variable space. The complexity of MRD is quadratic in the number of samples n per iteration using a sparse Gaussian process. Posterior inference and prediction using the MRD model was performed with Matlab package `vargplvm` (Damianou et al., 2012). We used the linear kernel with feature selection (i.e., `Linard2` kernel), meaning that we

used the linear version of this model for a fair comparison. We ran the MRD model on our simulated data with the correct number of factors.

We summarize the parameter choices for all methods here:

sGFA: We used the `getDefaultOpts` function in the sGFA package to set the default parameters. In particular, the ARD prior was set to $Ga(10^{-3}, 10^{-3})$. The prior on the inclusion probabilities was set to $beta(1, 1)$. *Total MCMC iterations* were set to 10^5 with `sampling iterations` set to $1,000$ and `thinning steps` set to 5.

GFA: We used the `getDefaultOpts()` function in the GFA package to set the default parameters. In particular, the ARD prior for both loading and error variance was set to $Ga(10^{-14}, 10^{-14})$. The `maximum iteration` parameter was set to 10^5, and the "L-BFGS" optimization method was used.

RCCA: The regularization parameter was chosen using leave-one-out cross-validation on an 11×11 grid from 0.0001 to 0.01 using the function `estim.regul` in the `CCA` package.

SCCA: We used the `PMA` package with Lasso penalty (the `typex` and `typez` parameters in the function `CCA` were set to "standard"). This corresponds to setting the ℓ_1 bound of the projection vector to $0.3\sqrt{p_w}$ for $w = 1, 2$.

JFA: The ARD priors for both the loading and factor scores were set to $Ga(10^{-5}, 10^{-5})$. The parameters of the beta process prior were set to $\alpha = 0.1$ and $c = 10^4$. The MCMC iterations were set to $1,000$ with 200 iterations of burn-in. As is the default settings, we did not thin the chain.

MRD: We used the `svargplvm_init` function in the `GPLVM` package to initialize parameters. The `linar2` kernel was chosen for all observations. Latent variables were initialized by concatenating the observation matrices first (the 'concatenated' option) and then performing PCA. Other parameters were set by `svargplvm_init` with default options.

6.3 Metrics for comparison

To compare the results of BASS with the alternative methods, we used the sparse and dense stability indices (Gao et al., 2013) to quantify the distance between the simulated loadings and the recovered loadings. The sparse stability index (SSI) measures the similarity between columns of sparse matrices. SSI is invariant to column scale and label switching, but it penalizes factor splitting and matrix rotation; larger values of SSI indicate better recovery. Let $\boldsymbol{C} \in \mathbb{R}^{k_1 \times k_2}$ be the absolute correlation matrix of columns of two sparse loading matrices. Then SSI is calculated by

$$SSI = \frac{1}{2k_1} \sum_{h_1=1}^{k_1} \left(\max(\boldsymbol{c}_{h_1,\cdot}) - \frac{\sum_{h_2=1}^{k_2} I(c_{h_1,h_2} > \overline{\boldsymbol{c}}_{h_1,\cdot}) c_{h_1,h_2}}{k_2 - 1} \right) + \frac{1}{2k_2} \sum_{h_2=1}^{k_2} \left(\max(\boldsymbol{c}_{\cdot,h_2}) - \frac{\sum_{h_1=1}^{k_1} I(c_{h_1,h_2} > \overline{\boldsymbol{c}}_{\cdot,h_2}) c_{h_1,h_2}}{k_1 - 1} \right).$$

The dense stability index (DSI) quantifies the difference between dense matrix columns, and is invariant to orthogonal matrix rotation, factor switching, and scale; DSI values closer to zero indicate better recovery. Let $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ be the dense matrices. DSI is calculated by

$$DSI = \frac{1}{p^2} tr(\boldsymbol{M}_1 \boldsymbol{M}_1^T - \boldsymbol{M}_2 \boldsymbol{M}_2^T).$$

We extended the stability indices to allow multiple coupled observations as in our simulations. In *Sim1*, *Sim3*, and *Sim5*, all factors are sparse, and SSIs were calculated between the true sparse loading matrices and recovered sparse loading matrices. In *Sim2*, *Sim4*, and *Sim6*, because none of the methods other than BASS explicitly distinguished sparse and dense factors, we categorized each recovered factor as follows. We first selected a global sparsity threshold on the elements of the combined loading matrix; here we set that value to 0.15. Elements below this threshold were set to zero in the loading matrix. Then we chose the first five loading columns with the fewest non-zero elements as the sparse loadings in *Sim2*, first four such loadings as the sparse loadings in *Sim4*, and first six such loadings as sparse in *Sim6*. The remaining loading columns were considered dense loadings and were not zeroed according to the global sparsity threshold. We found that varying the sparsity threshold did not affect the separation of sparse and dense loadings significantly across methods. SSIs were then calculated for the true sparse loading matrix and the recovered sparse loadings across methods.

To calculate DSIs, we treated the loading matrices $\boldsymbol{\Lambda}^{(w)}$ for each observation separately, and calculated the DSI for the recovered dense components of each observation. The DSI for each method was the sum of the m separate DSIs. Because the loading matrix is marginalized out in MRD (Lawrence, 2005), we excluded MRD from this comparison.

We further evaluated the prediction performance of BASS and other methods. In the BASS model (Equation 6), the joint distribution of any one observation $\boldsymbol{y}_i^{(w)}$ and all other observations $\boldsymbol{y}_i^{(-w)}$ can be written as

$$\begin{pmatrix} \boldsymbol{y}_i^{(w)} \\ \boldsymbol{y}_i^{(-w)} \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Lambda}^{(w)}(\boldsymbol{\Lambda}^{(w)})^T + \boldsymbol{\Sigma}^{(w)} & \boldsymbol{\Lambda}^{(w)}(\boldsymbol{\Lambda}^{(-w)})^T \\ \boldsymbol{\Lambda}^{(-w)}(\boldsymbol{\Lambda}^{(w)})^T & \boldsymbol{\Lambda}^{(-w)}(\boldsymbol{\Lambda}^{(-w)})^T + \boldsymbol{\Sigma}^{(-w)} \end{pmatrix} \right],$$

where $\boldsymbol{\Lambda}^{(-w)}$ and $\boldsymbol{\Sigma}^{(-w)}$ are the loading matrix and residual covariance excluding the w^{th} observation. Therefore, the conditional distribution of $\boldsymbol{y}_i^{(w)}$ is a multivariate response in a multivariate linear regression model, where $\boldsymbol{y}_i^{(-w)}$ are the predictors; the mean term takes the form

$$\begin{aligned} \mathbb{E}(\boldsymbol{y}_i^{(w)} | \boldsymbol{y}_i^{(-w)}) &= \boldsymbol{\Lambda}^{(w)}(\boldsymbol{\Lambda}^{(-w)})^T \big(\boldsymbol{\Lambda}^{(-w)}(\boldsymbol{\Lambda}^{(-w)})^T + \boldsymbol{\Sigma}^{(-w)}\big)^{-1} \boldsymbol{y}_i^{(-w)} \\ &= \sum_{h=1}^{k} \boldsymbol{\lambda}_{\cdot h}^{(w)} (\boldsymbol{\lambda}_{\cdot h}^{(-w)})^T \big(\boldsymbol{\Lambda}^{(-w)}(\boldsymbol{\Lambda}^{(-w)})^T + \boldsymbol{\Sigma}^{(-w)}\big)^{-1} \boldsymbol{y}_i^{(-w)}. \end{aligned} \tag{13}$$

We used this conditional distribution to predict specific observations given others. For the six simulations, we used the simulated data as training data for training sample sizes $n_t = \{30, 50\}$, and, additionally, simulated data sets with training sample sizes $n_t =$

	EM	MCMC-EM	PX-EM
Sim1	79.17%	99.17%	91.67%
Sim2	61.25%	93.75%	85.62%
Sim3	50.00%	78.57%	73.57%
Sim4	62.78%	86.11%	82.78%
Sim5	17.22%	86.67%	66.67%
Sim6	13.64%	60.45%	62.73%

Table 4: **Percentage of latent factors correctly identified across** 20 **runs with** $n = 40$**.** The columns represent the runs of EM, EM initialized with MCMC (MCMC-EM), and EM initialized with PX-EM.

$\{10, 100, 200\}$. Then, we generated $n_s = 200$ samples as test data using the true model parameters, simulating the corresponding test data factors $\boldsymbol{X} \sim \mathcal{N}(0, 1)$. For each simulation study, we chose at least one observation in the test data as the response and used the other observations and model parameters estimated from the training data to perform prediction. Mean squared error (MSE) was used to evaluate the prediction performance. For *Sim1* and *Sim2*, $\boldsymbol{y}_i^{(2)}$ was the response; for *Sim3* and *Sim4*, $\boldsymbol{y}_i^{(3)}$ was the response; and for *Sim5* and *Sim6*, $\boldsymbol{y}_i^{(8)}$, $\boldsymbol{y}_i^{(9)}$ and $\boldsymbol{y}_i^{(10)}$ were the responses.

6.4 Results of the simulation comparison

We first evaluated the performance of BASS and the other methods in terms of recovering the correct number of sparse and dense factors in the six simulations (Figures S3-S8). We calculated the percentage of correctly identified factors across 20 runs in the simulations with $n = 40$ (Table 4). Qualitatively, BASS recovered the closest matches to the simulated loading matrices across all methods (Figures 3, S1, S2). The correctly estimated loading matrices by the three different BASS initializations produced similar results; we only plot matrices from the PX-EM method.

6.4.1 Results on simulations with two observations (CCA)

Comparing results with two observations (*Sim1* and *Sim2*), our model produced the best SSIs and DSIs among all methods across all sample sizes (Figures 4). sGFA's performance was limited for these simulations because the ARD prior does not produce sufficient element-wise sparsity, resulting in low SSIs (Figure 4). As a consequence of not matching sparse loadings well, sGFA had difficulty recovering dense loadings, especially with small sample sizes (Figure 4). GFA had difficulty recovering sparse loadings because of column-wise ARD priors with the same limitation (Figure 3, Figure 4). Its dense loadings were indirectly affected by the lack of sufficient sparsity for small sample sizes (Figure 4). RCCA also had difficulty in the two simulations because the recovered loadings were not sufficiently sparse using the ℓ_2-type penalty (Figure 3).

SCCA recovered shared sparse loadings well in *Sim1* (Figure 3). However SCCA does not model local covariance structure, and therefore was unable to recover the sparse loadings specific to either of the observations in *Sim1* (Figure 3A) resulting in poor SSIs (Figure

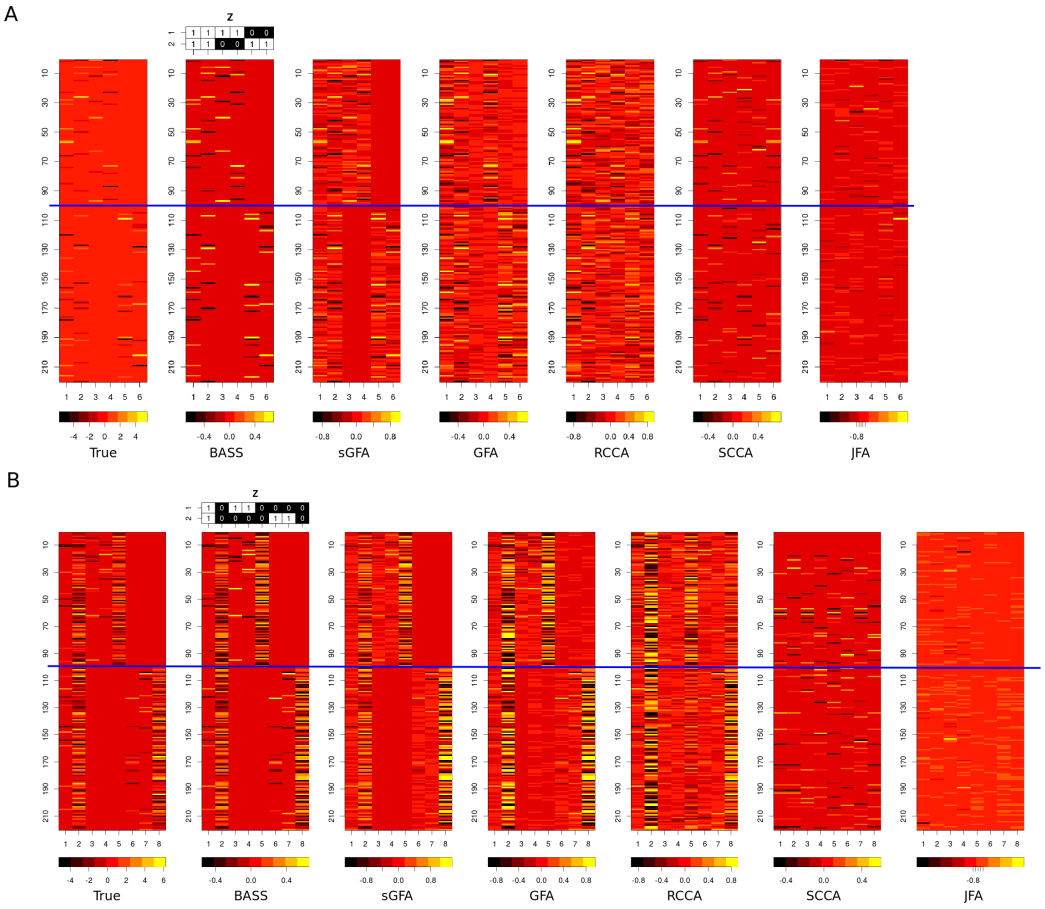

Figure 3: **Simulation results with two paired observations.** We reordered the columns of the recovered matrices and, where necessary, multiplied columns by -1 for easier visual comparisons. Horizontal lines separate the two observations. Panel A: Comparison of the recovered loading matrices using different models on *Sim1*. Panel B: Comparison of the recovered loading matrices using different models on *Sim2*.

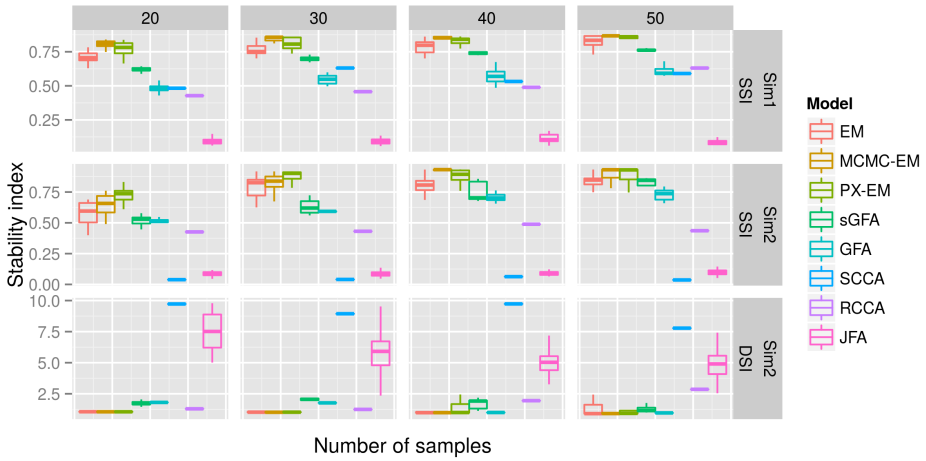

Figure 4: **Comparison of stability indices on recovered loading matrices with two observations.** Each stability index is plotted across 20 runs. For SSI, a larger value indicates better recovery; for DSI, a smaller value indicates better recovery. The boundaries of the box are the first and third quartiles. The line extends to the highest and lowest observations that are within 1.5 times the distance of the first and third quartiles beyond the box boundaries.

4). Adding dense loadings deteriorated the performance of SCCA (Figures 3B, 4). The JFA model did not recover the true loadings matrix well because of insufficient sparsity in the loadings and additional sparsity in the factors (Figure 3). The SSIs and DSIs for JFA reflect this data-model mismatch (Figure 4).

We next evaluated the predictive performance of these methods for two observations. In *Sim1*, SCCA achieved the best prediction accuracy in three training sample sizes (Table 5). We attribute this to SCCA recovering well the shared sparse loadings (Figure 3) because the prediction accuracy is only a function of the shared loadings. Note (Equation 13) that zero columns in either $\mathbf{\Lambda}^{(w)}$ or $\mathbf{\Lambda}^{(-w)}$ decouple the contribution of the corresponding factors to the prediction of $\boldsymbol{y}_i^{(w)}$. In *Sim2*, shared sparse and dense factors contribute to the prediction performance, and BASS achieved the best prediction accuracy (Table 5).

6.4.2 Results on simulations with four observations (GFA)

For simulations with four observations (*Sim3* and *Sim4*), BASS correctly recovered sparse and dense factors and their active observations (Figure S1). sGFA achieved column-wise sparsity for two observations; however, sparsity levels within factors were insufficient to match the simulations. GFA results produced insufficient column-wise sparsity: columns with zero values were not effectively removed (Figure S1B). Element-wise shrinkage in GFA was less effective than either BASS or sGFA (Figure S1). The results of SCCA and JFA did not match the true loading matrices for the same reasons as in *Sim1* and *Sim2* (Figure S1). The results using stability indices showed that BASS produced the best SSIs and DSIs across models and almost all sample sizes (Figure 5). sGFA achieved similar SSI values in *Sim3* with $n = 40$ compared to BASS EM, but showed worse performance for BASS

		BASS													
		EM		MCMC-EM		PX-EM		sGFA		GFA		SCCA	RCCA	MRD-lin	
	n_t	Err	SD	Err	SD	Err	SD	Err	SD	Err	SD	Err	Err	Err	SD
Sim1	10	1.00	0.024	1.03	0.024	1.02	0.028	1.00	<1e-3	0.98	0.002	**0.88**	1.01	1.08	0.024
	30	0.90	0.022	**0.88**	0.001	0.88	0.003	0.92	0.005	0.93	0.002	0.88	0.97	1.00	0.016
	50	0.88	0.011	**0.87**	0.003	0.88	0.014	0.90	0.004	0.92	0.002	0.88	0.92	0.98	0.028
	100	0.88	0.010	**0.87**	0.001	0.87	0.005	0.89	0.003	0.89	<1e-3	0.87	0.91	0.97	0.016
	200	0.88	0.007	**0.87**	0.004	0.87	0.005	0.88	0.001	0.88	<1e-3	0.87	0.95	1.16	0.202
Sim2	10	0.80	0.161	0.82	0.162	**0.68**	0.003	0.74	0.043	0.89	0.023	0.86	0.72	1.14	0.002
	30	0.72	0.092	0.72	0.097	0.67	0.016	0.67	0.014	**0.66**	0.006	0.86	0.70	1.15	0.034
	50	0.71	0.155	0.70	0.155	0.65	0.105	**0.63**	0.009	0.67	<1e-3	0.85	0.72	1.17	0.009
	100	0.63	0.066	**0.61**	0.013	0.62	0.013	0.62	0.005	0.61	0.001	0.85	0.75	1.13	0.013
	200	0.65	0.099	**0.61**	0.012	0.63	0.020	0.62	0.007	0.61	0.002	0.85	0.81	1.55	0.591

Table 5: **Prediction accuracy with two observations on $n_s = 200$ test samples.** Test samples $\boldsymbol{y}_i^{(2)}$ are treated as the response, and training samples $\boldsymbol{y}_i^{(1)}$ are used to estimate parameters in order to predict the response. Prediction accuracy is measured by mean squared error (MSE) between simulated $\boldsymbol{y}_i^{(1)}$ and $\mathbb{E}(\boldsymbol{y}_i^{(1)}|\boldsymbol{y}_i^{(2)})$. Values presented are the mean MSE (Err) and standard deviation (SD) across 20 runs of each method. If SD is missing for a method, then that method was deterministic.

		BASS												
		EM		MCMC-EM		PX-EM		sGFA		GFA		SCCA	MRD-lin	
	n_t	Err	SD	Err	SD	Err	SD	Err	SD	Err	SD	Err	Err	SD
Sim3	10	1.03	0.044	1.02	0.019	1.01	0.010	1.00	<1e-3	**0.97**	0.001	1.00	1.00	<1e-3
	30	0.91	0.049	**0.87**	0.016	0.88	0.007	0.90	0.007	0.93	0.003	1.00	0.99	0.021
	50	**0.85**	0.019	0.85	<1e-3	0.87	0.038	0.87	0.005	0.88	0.002	1.01	1.04	0.095
	100	0.85	0.019	**0.84**	0.002	0.84	0.003	0.86	0.004	0.87	0.001	1.11	0.92	0.014
	200	0.84	0.001	0.84	<1e-3	0.84	0.004	0.84	0.001	**0.83**	0.001	1.13	1.16	0.140
Sim4	10	1.05	0.095	1.03	0.094	1.10	0.138	**1.00**	<1e-3	1.32	0.029	1.35	1.98	0.067
	30	0.97	0.020	**0.95**	0.015	0.96	0.013	0.97	0.007	1.03	0.003	1.40	1.50	0.090
	50	0.94	0.013	**0.93**	0.005	0.94	0.012	0.95	0.005	1.02	0.017	1.40	1.50	0.084
	100	**0.93**	0.015	0.93	0.007	0.93	0.010	0.94	0.003	0.96	<1e-3	1.51	1.47	0.088
	200	0.91	0.029	0.92	0.022	**0.89**	0.047	0.93	0.001	0.89	0.001	1.77	1.58	0.132

Table 6: **Prediction accuracy with four observations on $n_s = 200$ test samples.** Test samples $\boldsymbol{y}_i^{(3)}$ are treated as the response, and training samples $\boldsymbol{y}_i^{(1)}$, $\boldsymbol{y}_i^{(2)}$, and $\boldsymbol{y}_i^{(4)}$ are used to estimate parameters in order to predict the response. Prediction accuracy is measured by mean squared error (MSE) between simulated $\boldsymbol{y}_i^{(3)}$ and $\mathbb{E}(\boldsymbol{y}_i^{(3)}|\boldsymbol{y}_i^{(1)}, \boldsymbol{y}_i^{(2)}, \boldsymbol{y}_i^{(4)})$. Values presented are the mean MSE (Err) and standard deviation (SD) across 20 runs of each method. Standard deviation (SD) is missing for SCCA because the method is deterministic.

MCMC-EM and PX-EM. The advantage of BASS relative to the other methods is apparent in these SSI comparisons, which specifically highlight interpretability and robust recovery of this type of latent structure (Figure 5).

In the context of prediction using four observation matrices, BASS achieved the best prediction performance with $\boldsymbol{y}_i^{(3)}$ as the response and the remaining observations as predictors (Table 6). In particular, the MCMC-initialized EM approach had the best overall prediction performance across methods for these two simulations.

6.4.3 Results on simulations with ten observations (GFA)

When we increased the number of observations to ten (*Sim5* and *Sim6*), BASS still correctly recovered the sparse and dense factors and their active observations (Figure S2). sGFA effectively performed column-wise selection although element-wise sparsity remained inadequate (Figure S2). GFA did not recover sufficient column-wise or element-wise sparsity (Figure S2). SCCA and JFA both failed to recover the true loading matrices (Figure

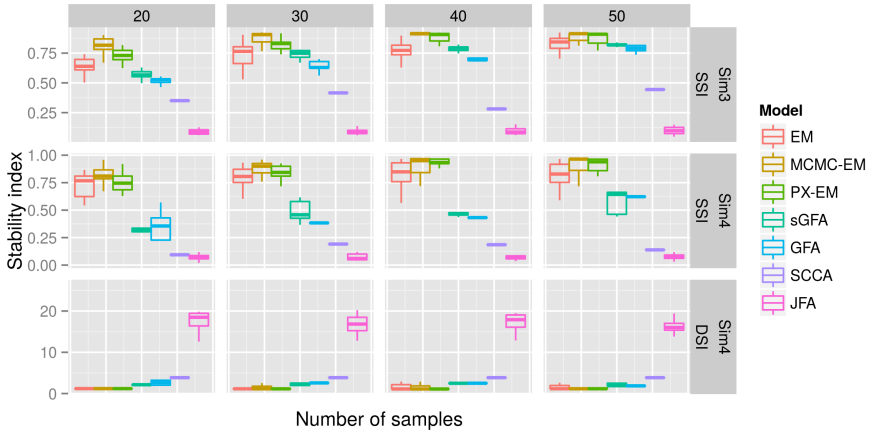

Figure 5: **Comparison of stability indices on recovered loading matrices with four observations.** Each stability index is plotted across 20 runs. For SSI, a larger value indicates better recovery; for DSI, a smaller value indicates better recovery. The boundaries of the box are the first and third quartiles. The line extends to the highest and lowest values that are within 1.5 times the distance of the first and third quartiles beyond the box boundaries.

		BASS												
		EM		MCMC-EM		PX-EM		sGFA		GFA		SCCA	MRD-lin	
	n_t	Err	SD	Err	SD	Err	SD	Err	SD	Err	SD	Err	Err	SD
Sim5	10	1.01	0.020	1.00	0.011	1.00	0.007	**0.99**	0.008	1.00	0.002	0.99	1.49	0.001
	30	0.88	0.031	**0.86**	0.018	0.87	0.028	0.89	0.005	0.90	0.002	0.99	1.01	0.035
	50	0.86	0.023	**0.85**	<1e-3	0.86	0.022	0.87	0.003	0.88	0.001	0.99	0.97	0.020
	100	**0.85**	0.007	0.85	<1e-3	0.85	0.002	0.86	0.003	0.87	0.001	1.01	0.92	0.039
	200	0.85	0.006	0.84	<1e-3	0.84	<1e-3	0.84	0.001	**0.83**	0.001	0.96	1.06	0.105
Sim6	10	0.61	0.164	0.57	0.116	**0.51**	0.031	0.58	0.012	0.75	0.011	0.97	1.00	<1e-3
	30	0.49	0.160	0.40	0.093	**0.38**	0.007	0.43	0.006	0.40	0.005	0.98	0.46	0.006
	50	0.44	0.099	**0.39**	0.011	0.39	0.004	0.41	0.002	0.40	0.001	1.01	0.42	0.009
	100	**0.39**	0.033	0.39	0.004	0.39	0.011	0.39	0.002	0.39	0.001	0.97	0.52	0.249
	200	**0.38**	0.003	0.38	0.001	0.38	0.001	0.39	0.001	0.39	0.001	1.01	0.40	0.020

Table 7: **Prediction mean squared error with ten observations on $n_s = 200$ test samples.** Test samples $\boldsymbol{y}_i^{(8)}, \boldsymbol{y}_i^{(9)}$ and $\boldsymbol{y}_i^{(10)}$ are treated as the response and the rest of the observations are used as the training data to estimate parameters used to predict the response. Prediction accuracy is measured by mean squared error (MSE) between simulated responses and predicted responses. Values presented are the mean MSE (Err) and standard deviation (SD) across 20 runs of each method. Standard deviation (SD) is missing for SCCA because the method is deterministic.

S2). For the stability indices, BASS with MCMC-EM and PX-EM produced the best SSIs in *Sim5* across all methods and for almost all sample sizes (Figures 6). Here sGFA achieved equal or better SSIs than BASS EM, highlighting the sensitivity of BASS EM to initializations. GFA had equivalent or worse SSIs than BASS EM. In this pair of simulations, the advantages of BASS for flexible and robust column-wise and element-wise shrinkage are apparent (Figures 6). BASS also achieved the best prediction performance in *Sim5* and *Sim6* with ten observations (Table 6).

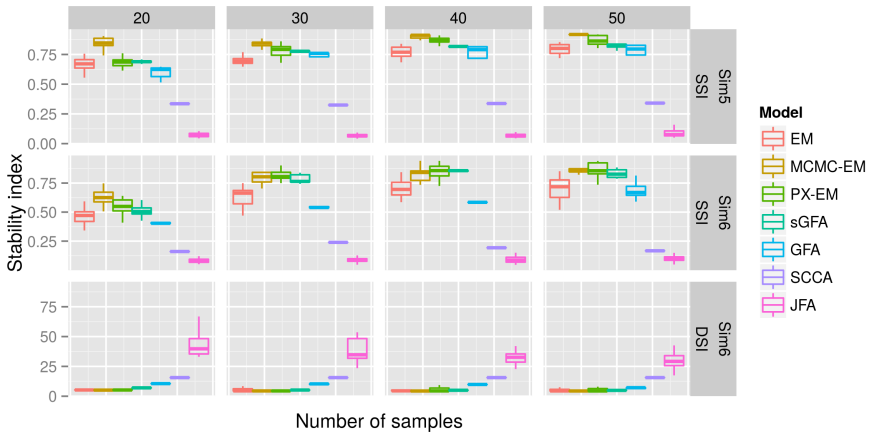

Figure 6: **Comparison of stability indices on recovered loading matrices with ten observations.** Each stability index is plotted across 20 runs. For SSI, a larger value indicates better recovery; for DSI, a smaller value indicates better recovery. The boundaries of the box are the first and third quartiles. The line extends to the highest and lowest value within 1.5 times the distance of the first and third quartiles beyond the box boundaries.

Across the three BASS methods, MCMC-EM had the most accurate performance across nearly all simulation settings. However, this performance boost comes with the price of running a small number of Gibbs sampling iterations with complexity of $O(k^3p + k^2pn)$ per iteration. When p is large, even a few iterations are computationally infeasible. PX-EM, on the other hand, has the same complexity as EM, and showed robust and accurate simulation results relative to EM. In the following real applications, we used BASS EM initialized with a small number of iterations of PX-EM.

7. Applying BASS to Mulan Library, genomics data, and text analysis

In this section we considered three real data applications of BASS. In the first application, we evaluated the prediction performance for multiple correlated response variables in the Mulan Library (Tsoumakas et al., 2011). In the second application, we applied BASS to gene expression data from the Cholesterol and Pharmacogenomic (CAP) study. The data consist of expression measurements for about ten thousands genes in 480 lymphoblastoid cell lines (LCLs) under two experimental conditions (Mangravite et al., 2013; Brown et al., 2013). BASS was used to detect sparse covariance structures specific to each experimental condition. In the third application, we applied BASS to approximately $20,000$ newsgroup posts to 20 newsgroups (Joachims, 1997) in order to perform multiclass classification.

7.1 Multivariate response prediction: The Mulan Library

The Mulan Library consists of multiple data sets collected for the purpose of evaluating multi-label predictions (Tsoumakas et al., 2011). This library was used to test the Bayesian CCA model (GFA in our simulations) for multi-label prediction vectors converted to multiple binary label vectors (one-hot encoding) (Klami et al., 2013). There are two observations

($m = 2$): the matrix of labels were treated as one observation ($\boldsymbol{Y}^{(1)}$) and the features were treated as another ($\boldsymbol{Y}^{(2)}$). Recently Mulan added multiple regression data sets with continuous variables. We chose ten benchmark data sets from the Mulan Library. Four of them (`bibtex, delicious, mediamill, scene`) have binary responses and were studied previously (Klami et al., 2013). Another six data sets (`rf1, rf2, scm1d, scm20d, atp1d, atp7d`) have continuous responses (Table 8). For all data sets, we removed features with identical values for all samples in the training set as uninformative. For the continuous response data sets, for each value, we subtracted the mean and divided by the standard deviation of each feature.

We ran BASS, sGFA, GFA, and MRD-lin on the ten data sets, and compared the results using prediction accuracy. For data sets with binary labels, we quantified prediction error using the Hamming loss between the predicted labels and true labels. The predicted labels on the test samples were calculated using the same thresholding rules as in earlier work (Klami et al., 2013). The value of the threshold was chosen so that the Hamming loss between the estimated labels and the true labels in the training set was minimized. We used the R package `PresenceAbsence` and Matlab function `perfcurve` to find the thresholds to produce binary classifications from continuous predictions. In particular, the R package `PresenceAbsence` selects the threshold by maximizing the percent correctly classified, which corresponds to minimizing the Hamming loss. For continuous variables, mean squared error (MSE) was used to evaluate prediction accuracy. We initialized BASS with 500 factors and 50 PX-EM iterations. The other models were set to the default parameters with the number of factors set to $\min(p_1, p_2, 50)$ (see Simulations for details). All methods were run 20 times, and minimum errors were reported (Tables S1-S11).

BASS achieved the best prediction accuracy in five of the ten data sets (Table 8). For the data sets with a binary response, sGFA produced the best performance compared with other methods, achieving the smallest MSE in all four data sets. GFA had the most stable results in terms of SD in the four data sets. For the continuous response, BASS outperformed the other models in four out of six data sets. GFA again had the most stable MSE compared with other methods. The good performance of BASS on the data sets with continuous response variables may be attributed to the structured sparsity on the loading matrix, achieving the intended gains in generalization error from flexible regularization. Although the ARD prior used in GFA did not produce consistently sparse loadings, this model generated the most stable predictive results.

7.2 Gene expression data analysis

We applied our BASS model to gene expression data from the Cholesterol and Pharmacogenomic (CAP) study, consisting of expression measurements for $10,195$ genes in 480 lymphoblastoid cell lines (LCLs) after 24-hour exposure to either a control buffer ($\boldsymbol{Y}^{(1)}$) or $2\mu M$ simvastatin acid ($\boldsymbol{Y}^{(2)}$) (Mangravite et al., 2013; Brown et al., 2013). In this example, the number of observations ($m = 2$) represents gene expression levels on the same samples and genes after the two different exposures. The expression levels were preprocessed to adjust for experimental traits (batch effects and cell growth rate) and clinical traits of donors (age, BMI, smoking status, and sex). We projected the adjusted expression levels to the quantiles of a standard normal within gene to control for outlier effects and applied BASS

Data Set	p_1	p_2	n_t	n_s	BASS		sGFA		GFA		MRD-lin	
					Err	SD	Err	SD	Err	SD	Err	SD
bibtex	1836	159	4880	2515	0.014	0.001	0.014	0.001	**0.014**	<1e-3	0.014	0.001
delicious	983	500	12920	3185	0.016	0.001	**0.016**	<1e-3	0.017	<1e-3	0.020	<1e-3
mediamill	120	101	30993	12914	**0.032**	0.001	0.032	0.005	0.034	<1e-3	0.043	<1e-3
scene	294	6	1211	1196	0.131	0.016	**0.123**	0.029	0.130	0.002	0.138	0.026
rf1	64	8	4108	5017	**0.292**	0.050	0.390	0.008	0.309	<1e-3	0.370	0.146
rf2	576	8	4108	5017	**0.271**	0.027	0.478	0.004	0.427	0.001	0.438	0.160
scm1d	280	16	8145	1658	**0.211**	0.005	0.225	0.028	0.213	<1e-3	0.212	0.163
scm20d	61	16	7463	1503	0.650	0.015	**0.538**	0.006	0.720	0.002	0.608	0.033
atp1d	370	6	237	100	**0.176**	0.032	0.208	0.006	0.201	0.001	0.219	0.113
atp7d	370	6	196	100	0.597	0.063	0.537	0.015	**0.537**	0.003	0.545	0.049

Table 8: **Multi-variate response prediction in the Mulan library.** p_1: the number of features; p_2: the number of responses; n_t: the number of training samples; n_s: the number of test samples. The first four data sets have binary responses, and the final six are continuous responses. For binary responses, error (Err) is evaluated using Hamming loss between predicted labels and test labels in test samples. For continuous responses, mean squared error (MSE) is used to quantify error. Values shown are the minimum Hamming loss or MSE across 20 runs, and the standard deviation (SD).

with the initial number of factors set to $k = 2,000$. We performed parameter estimation 100 times on these data with 100 iterations of PX-EM to initialize EM. Across these 100 runs, the estimated number of recovered factors was approximately 870 (Table S2), with only a few dense factors (Table S12) likely due to the adjustments made in the preprocessing step. The total percentage of variance explained (PVE) by the recovered latent structure was 14.73%, leaving 85.27% of the total variance to be captured in the residual error.

We computed the PVE of the sparse factors alone (Figure S9A). The PVE for the h^{th} factor was calculated as the variance explained by the h^{th} factor divided by the total variance: $tr(\boldsymbol{\lambda}_{\cdot h}\boldsymbol{\lambda}_{\cdot h}^T)/tr(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Sigma})$. Shared sparse factors explained more variance than observation-specific sparse factors, suggesting that variation in expression levels across genes was driven by structure shared across the exposures to a greater degree than by exposure-specific structure. Moreover, 87.5% of the observation-specific sparse factors contained fewer than 100 genes, and 0.7% had more than 500 genes. The shared sparse factors had, on average, more genes than the observation-specific factors: 72% shared sparse factors had fewer than 100 genes, and 4.5% had more than 500 genes. (Figure S9B).

The sparse factors specific to each observation characterized the local sparse covariance estimates. As we pursue more carefully elsewhere (Gao et al., 2014), we used observation-specific sparse factors to a construct a gene co-expression network that is uniquely found in the samples from that exposure while explicitly controlling for shared covariance across exposures (Zou et al., 2013). The problem of constructing condition specific co-expression networks has been studied by both machine learning and computational biology communities (Li, 2002; Ma et al., 2011). BASS provides an alternative approach to solve this problem. We denote $\boldsymbol{B}_s^{(w)}$ as the sparse loadings in $\boldsymbol{B}^{(w)}$ ($w \in \{1, 2\}$) and $\boldsymbol{X}_s^{(w)}$ as the factors corresponding to sparse loadings for observation w. Then, $\boldsymbol{\Omega}_s^{(w)} = \boldsymbol{B}_s^{(w)} Var(\boldsymbol{X}_s^{(w)})(\boldsymbol{B}_s^{(w)})^T + \boldsymbol{\Sigma}^{(w)}$ represents the regularized estimate of the covariance matrix specific to each observation after controlling for the contributions of the dense factors.

In our model, $Var(\boldsymbol{X}_s^{(w)}) = \boldsymbol{I}$, and so the covariance matrix becomes $\boldsymbol{\Omega}_s^{(w)} = \boldsymbol{B}_s^{(w)}(\boldsymbol{B}_s^{(w)})^T + \boldsymbol{\Sigma}^{(w)}$. We inverted this positive definite covariance matrix to get a precision matrix $\boldsymbol{R}^{(w)} = (\boldsymbol{\Omega}_s^{(w)})^{-1}$. The partial correlation between gene j_1 and j_2, representing the correlation between the two features conditioned on the remaining features, is then calculated by normalizing each entry in the precision matrix (Edwards, 2000; Schäfer and Strimmer, 2005):

$$\rho_{j_1 j_2}^{(w)} = -\frac{r_{j_1 j_2}^{(w)}}{\sqrt{r_{j_1 j_1}^{(w)} r_{j_2 j_2}^{(w)}}}.$$

A partial correlation that is (near) zero for two genes (j_1, j_2) suggests that they are conditionally independent; non-zero partial correlation implies a direct relationship between two genes, and a network edge is added between the genes. The resulting undirected network is an instance of a Gaussian Markov random field, also known as a Gaussian graphical model (Edwards, 2000; Koller and Friedman, 2009). We note that BASS was the only method that enables construction of a condition specific network: sGFA could not be applied to data of this magnitude, GFA did not shrink the column selection sufficiently to recover sparsity in the condition specific covariance matrix, and SCCA only recovers shared sparse projections.

We used the following method to combine the results of 100 runs to construct a single observation-specific gene co-expression network for each observation. For each run, we first constructed a network by connecting genes with partial correlation greater than a threshold (0.01). Then we combined the 100 run-specific networks to construct a single network by removing all network edges that appeared in fewer than 50 (50%) of the networks. The two observation-specific gene co-expression networks contained 160 genes and $1,244$ edges (buffer treated, Figure 7A), and 154 genes and $1,030$ edges (statin-treated, Figure 7B), respectively.

7.3 Twenty newgroups analysis

In this application, we used BASS and related methods for multiclass classification in the 20 Newsgroups data (Joachims, 1997). The documents were processed so that duplicates and headers were removed, resulting $18,846$ documents. The data were downloaded using the `scikit-learn` Python package (Pedregosa et al., 2011). We converted the raw data into TF-IDF feature vectors and selected 319 words using SVM feature selection from `scikit-learn`. One document had a zero vector across the subset of vocabulary words and was removed. We held out 10 documents at random from each newsgroup as test data (Table S14).

We applied BASS to the transposed data matrices with the 20 newsgroups as 20 observations. We set the initial number of factors to $k = 1,000$ and ran EM 100 times from random starting points, each with 100 initial PX-EM iterations. There were on average 820 factors recovered across the runs.

To analyze the newsgroup-specific words, we calculated the Pearson correlation of each estimated loading and newsgroup indicator vectors consisting of ones for all of the documents in one newsgroup and zeros for documents in the other groups. Then, for each newsgroup, the loadings with the ten largest absolute value correlation coefficients were used to find the

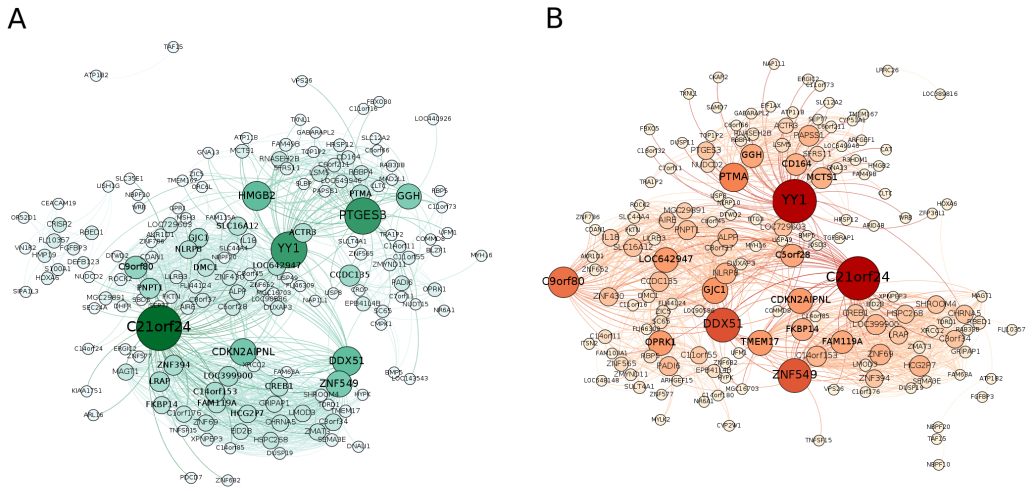

Figure 7: **Observation-specific gene co-expression networks from the CAP data.** The two networks represent the co-expressed genes specific to buffer-treated samples (Panel A) and statin-treated samples (Panel B). The node size is scaled according to the number of shortest paths from all vertices to all others that pass through that node (*betweenness centrality*).

ten words with the largest absolute value factor scores. The results from one run include, for example, the `rec.autos` newsgroup with 'car', 'dealer' and 'oil,' as top words, and the `rec.sport.baseball` newsgroup with 'baseball', 'braves,' and 'runs' as top words (Table 9).

We further partitioned the newsgroups into six classes according to subject matter to analyze the top words across newsgroups subgroups (Table 10). As above, we calculated the Pearson correlation with the binary indicator vectors for documents in newsgroup subgroups, and we analyzed the top ten words in the ten factors with largest absolute value correlation coefficients with these subsets of newsgroups (Table 10). We found, for example, that the newsgroups `talk.religion.misc`, `alt.atheism` and `soc.religion.christian` had 'god', 'bible' and 'christian' as top shared words. Examining one of the selected shared loadings for this newsgroup subgroup (Figure 8A), we noticed that documents outside of these three newsgroups, for the most part, have negligible loadings. This analysis highlights the ability of BASS to recover meaningful shared structure among 20 observations.

alt.atheism		comp.graphics		comp.os.ms-windows.misc		comp.sys.ibm.pc.hardware		comp.sys.mac.hardware	
islam	atheism	graphics	polygon	windows	file	ide	drive	mac	powerbook
keith	mathew	3d	gif	thanks	go	scsi	motherboard	apple	quadra
okcforum	atheists	tiff	images	of	dos	controller	thanks	quadra	iisi
atheism	livesey	image	format	cica	microsoft	vlb	ide	duo	centris
livesey	of	image	pov	dos	the	bios	isa	centris	mac
comp.windows.x		misc.forsale		rec.autos		rec.motorcycles		rec.sport.baseball	
window	mit	sale	offer	car	dealer	dod	bmw	baseball	hitter
motif	lcs	sale	forsale	cars	oil	bike	riding	braves	ball
server	motif	for	the	engine	toyota	motorcycle	bikes	runs	year
widget	xterm	sell	shipping	ford	eliot	ride	dod	phillies	players
lcs	code	condition	offer	cars	cars	bike	bike	sox	players
rec.sport.hockey		sci.crypt		sci.electronics		sci.med		sci.space	
hockey	bruins	encryption	crypto	circuit	radio	geb	msg	it	people
nhl	pens	clipper	nsa	voltage	copy	medical	doctor	space	orbit
game	detroit	chip	nsa	amp	battery	diet	disease	for	henry
team	season	key	pgp	electronics	tv	cancer	geb	digex	moon
leafs	espn	des	tapped	audio	power	photography	doctor	for	shuttle
soc.religion.christian		talk.politics.guns		talk.politics.mideast		talk.politics.misc		talk.religion.misc	
god	sin	atf	fbi	israeli	israeli	cramer	government	sandvik	morality
clh	bible	firearms	stratus	jews	armenians	optilink	drugs	koresh	jesus
church	petch	guns	batf	israel	armenian	kaldis	president	sandvik	religion
christian	mary	gun	stratus	arab	jake	clinton	br	bible	god
heaven	church	handheld	waco	armenians	jewish	cramer	tax	christian	objective

Table 9: **Most significant words in the newsgroup-specific factors for 20 newsgroups.** For each newsgroup, we include the top ten words in the newsgroup-specific components.

Newsgroup classes	Top ten shared words		Newsgroup classes	Top ten shared words	
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	windows	dos	misc.forsale	sale	shipping
	thanks	mac		sell	ca
	graphics	go		condition	wanted
	file	scsi		offer	thanks
	window	server		forsale	edu
rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	dod	baseball	talk.politics.misc talk.politics.guns talk.politics.mideast	government	it
	car	ride		israeli	israel
	bike	cars		jews	gun
	motorcycle	bmw		atf	guns
	game	team		firearms	batf
sci.crypt sci.electronics sci.med sci.space	clipper	henry	talk.religion.misc alt.atheism soc.religion.christian	god	bible
	encryption	orbit		bible	heaven
	space	people		christian	sandvik
	chip	circuit		clh	faith
	digex	voltage		jesus	church

Table 10: **Top ten words in the factors shared among specific subgroups of newsgroups.** In the shared recovered components corresponding to subsets of newsgroups, we show the ten most significant words in these shared components for six different subsets of newsgroups.

To assess prediction quality, we used the factors estimated from the training set to classify documents in the test set into one of 20 newsgroups. To estimate the loadings in the test set, we left-multiplied the test data matrix by the Moore-Penrose pseudoinverse of factors estimated from training data. This gave a rough estimate of the loading matrix for test data. Then test labels were predicted using the ten nearest neighbors in the loading rows estimated for the training documents. For the 200 test documents, BASS achieved 58.3% accuracy (Hamming loss; Figure 8B). Because some of the newsgroups were closely related to each other with respect to topic, we partitioned the 20 newsgroups into six topics according to subject matter. Then, the ten nearest neighbors were used to predict the topic of the test data. In this experiment, BASS achieved approximately 74.12% accuracy (Hamming loss; Figure 8C; Table S3).

8. Discussion

There exists a rich set of methods to explore latent structure in paired or multiple observations jointly (e.g., Parkhomenko et al., 2009; Witten and Tibshirani, 2009; Zhao and Li, 2012, among others). The multiple trajectories of interpretation of these approaches as linear factor analysis models includes the original inter-battery and multi-battery models (Browne, 1979, 1980), the probabilistic CCA model (Bach and Jordan, 2005), the sparse probabilistic projection (Archambeau and Bach, 2009), and, most recently, the Bayesian CCA model (Klami et al., 2013) and GFA model (Klami et al., 2014b). Only recently has the idea of column-wise shrinkage, or group-wise sparsity, been applied to develop useful

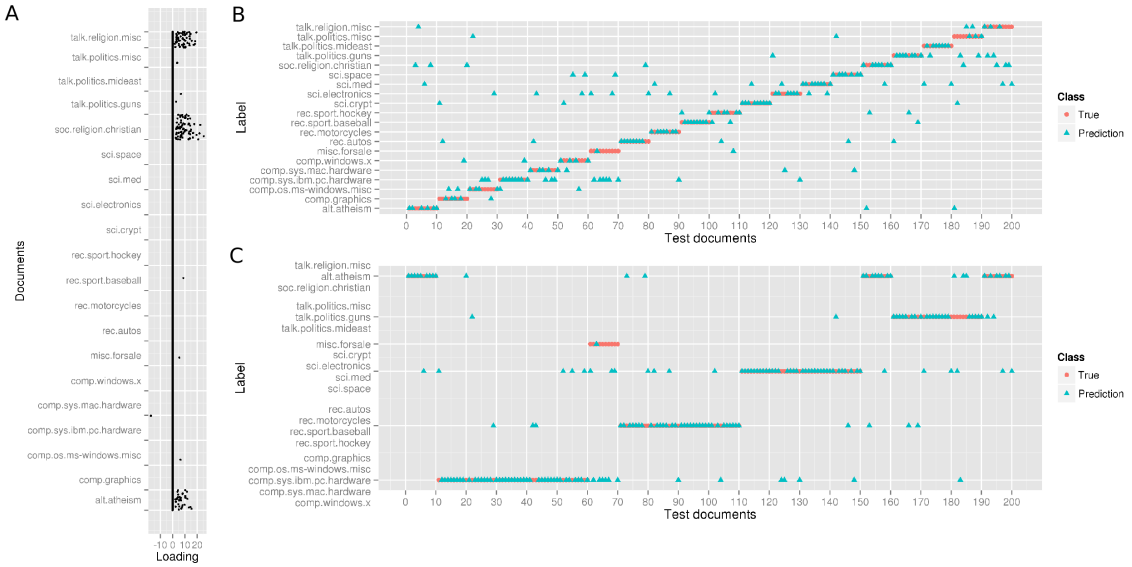

Figure 8: **Newsgroup prediction on** 200 **test documents.** Panel A: One factor loading selected as shared by three newsgroups (`talk.religion.misc`, `alt.atheism` and `soc.religion.christian`). Panel B: 20 Newsgroups predictions on 200 test documents using ten nearest neighbors from loadings estimated from the training data. Panel C: Document subgroup predictions based on six groups of similar newsgroups using ten nearest neighbors based on loadings estimated from the training data.

models for this problem. The advantage of column-wise shrinkage is to decouple portions of the latent space from specific observations and adaptively select the number of factors.

While the innovation of column-wise sparsity is primarily due to the ideas developed in the Bayesian CCA model (Virtanen et al., 2011), additional layers of shrinkage were required to create both column-wise and element-wise sparsity as is essential in real data analyses. The most recent attempt to develop such combined effects is the sGFA model (Khan et al., 2014) using a combination of an element-wise ARD prior with spike-and-slab prior for column selection. In our work here, we developed the necessary Bayesian prior and methodology framework to realize these advantages for the analysis of large data sets. In particular, we developed a structured sparse prior using three hierarchical layers of the three parameter beta ($\mathcal{TPB}$) distribution. This carefully formulated prior combines both column-wise and element-wise shrinkage with global shrinkage to adapt the level of sparsity—both column-wise and element-wise—to the underlying data, creating robustness to parameter settings that cannot be achieved using a single-layer ARD prior. The resulting BASS model also allows sparse and dense factor loadings, which proved essential for data scenarios that have this low-rank and sparse structure and has been pursued in classical statistics (Chandrasekaran et al., 2009; Candès et al., 2011; Zhou et al., 2011). We showed in the simulations that this regularization is essential for problems in the $p \gg n$ data scenario, which motivated this work. With the assumption of full column rank of dense loadings and one single observation, our model provides a Bayesian solution to the sparse and low-rank decomposition problem.

Column-wise shrinkage in BASS was achieved using the observation-specific global and column-specific $\mathcal{TPB}$ priors. With current parameter settings, it is equivalent to the horseshoe prior put on the entire column. The horseshoe prior has been shown to induce better shrinkage effects compared to the ARD prior, the Laplace prior (Bayesian lasso), and other similar shrinkage priors while remaining computationally tractable (Carvalho et al., 2010). In addition, our local shrinkage encourages element-wise sparsity. A two component mixture allows both dense and sparse factors to be recovered for any subset of observations. These shared factors have an interpretation as a supervised low-rank projection when one observation is supervised labels (e.g., the Mulan Library data). To the best of our knowledge, the BASS model is the first model in either the Bayesian or classical statistical literature that is able to capture low-rank and sparse decompositions among multiple observations.

We developed three algorithms that estimate the posterior distribution of our model or MAP parameter values. We found that EM with random initialization would occasionally get stuck in poor local optima. This motivated the development of a fast and robust PX-EM algorithm by introducing an auxiliary rotation matrix (Rocková and George, 2015). Initializing EM with PX-EM enabled EM to escape from poor initializations, illustrated in simulations. Our PX-EM and EM algorithms have better computational complexity than two competing approaches, GFA and sGFA, allowing for large-scale data application.

Extending multiple observation linear factor models to non-linear or non-Gaussian models has been studied recently (Salomatin et al., 2009; Damianou et al., 2012; Klami et al., 2014a; Klami, 2014). The ideas in this paper of inducing structured sparsity in the loadings has parallels in both of these settings. For example, we may consider structured Gaussian process kernels in the non-linear setting, where structure corresponds to known shared and observation-specific structure. A number of issues remain, including robustness of the recovered sparse factors across runs, scaling these methods to current studies in genomics, neuroscience, or text analysis, allowing for missing data, and developing approaches to include domain-specific structure across samples or features.

Acknowledgments

The authors would like to thank David Dunson and Sanvesh Srivastava for helpful discussions. The authors also appreciate constructive comments from Arto Klami and three anonymous reviewers. BEE, CG, and SZ were funded by NIH R00 HG006265 and NIH R01 MH101822. SZ was also funded in part by NSF DMS-1418261 and a Graduate Fellowship from Duke University. SM was supported in part by NSF DMS-1418261, NSF DMS-1209155, NSF IIS-1320357, and AFOSR under Grant FA9550-10-1-0436. All code and data are publicly available. The software for BASS is available at `https://github.com/judyboon/BASS`. The gene expression data were acquired through Gene Expression Omnibus (GEO) Accession number GSE36868. We acknowledge the PARC investigators and research team, supported by NHLBI, for collection of data from the Cholesterol and Pharmacogenetics clinical trial.

Appendix A. Markov chain Monte Carlo (MCMC) algorithm for posterior inference

We first derive the MCMC algorithm with Gibbs sampling steps for BASS. We write the joint distribution of the full model as

$$\begin{aligned} p(\boldsymbol{Y}, \boldsymbol{X}, &\boldsymbol{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{Z}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \\ &= p(\boldsymbol{Y}|\boldsymbol{\Lambda}, \boldsymbol{X}, \boldsymbol{\Sigma}) p(\boldsymbol{X}) \\ &\quad \times p(\boldsymbol{\Lambda}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\boldsymbol{\Delta}, \boldsymbol{Z}, \boldsymbol{\Phi}) p(\boldsymbol{\Delta}|\boldsymbol{\Phi}) p(\boldsymbol{\Phi}|\boldsymbol{T}) p(\boldsymbol{T}|\boldsymbol{\eta}) p(\boldsymbol{\eta}|\boldsymbol{\gamma}) \\ &\quad \times p(\boldsymbol{\Sigma}) p(\boldsymbol{Z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}), \end{aligned}$$

where $\boldsymbol{\Theta} = \{\theta_{jh}^{(w)}\}$, $\boldsymbol{\Delta} = \{\delta_{jh}^{(w)}\}$, $\boldsymbol{\Phi} = \{\phi_h^{(w)}\}$, $\boldsymbol{T} = \{\tau_h^{(w)}\}$, $\boldsymbol{\eta} = \{\eta^{(w)}\}$ and $\boldsymbol{\gamma} = \{\gamma^{(w)}\}$ are the collections of global-factor-local $\mathcal{TPB}$ prior parameters.

The full conditional distribution for latent factor $\boldsymbol{x}_i$ is

$$\boldsymbol{x}_i|- \sim \mathcal{N}_k\bigg((\boldsymbol{\Lambda}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda} + \boldsymbol{I})^{-1}\boldsymbol{\Lambda}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{y}_i, (\boldsymbol{\Lambda}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda} + \boldsymbol{I})^{-1}\bigg), \tag{14}$$

for $i = 1, \ldots, n$.

For $\boldsymbol{\Lambda}$, we derive the full conditional distributions of its p rows, $\boldsymbol{\lambda}_{j\cdot}$ for $j = 1, \ldots, p$,

$$\boldsymbol{\lambda}_{j\cdot}^T|- \sim \mathcal{N}_k\bigg((\sigma_j^{-2}\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{D}_j^{-1})^{-1}\sigma_j^{-2}\boldsymbol{X}\boldsymbol{y}_{j\cdot}^T, (\sigma_j^{-2}\boldsymbol{X}\boldsymbol{X}^T + \boldsymbol{D}_j^{-1})^{-1}\bigg),$$

where

$$\boldsymbol{D}_j^{-1} = \mathrm{diag}\bigg((\theta_{j1}^{(w_j)})^{I(z_1^{(w_j)}=1)}(\phi_1^{(w_j)})^{I(z_1^{(w_j)}=0)}, \ldots, (\theta_{jk}^{(w_j)})^{I(z_k^{(w_j)}=1)}(\phi_k^{(w_j)})^{I(z_k^{(w_j)}=0)}\bigg),$$

and w_j represents the observation that the j^{th} row belongs to.

The full conditional distributions of $\theta_{jh}^{(w)}$, $\delta_{jh}^{(w)}$ and $\phi_h^{(w)}$ with $z_h^{(w)} = 1$ are

$$\begin{aligned} \theta_{jh}^{(w)}|- &\sim \mathcal{GIG}\left(a - 1/2, 2\delta_{jh}^{(w)}, (\lambda_{jh}^{(w)})^2\right), \\ \delta_{jh}^{(w)}|- &\sim Ga\left(a + b, \phi_h^{(w)} + \theta_{jh}^{(w)}\right), \\ \phi_h^{(w)}|- &\sim Ga\left(p_w b + c, \sum_{j=1}^{p_w}\delta_{jh}^{(w)} + \tau_h^{(w)}\right), \end{aligned}$$

where $\mathcal{GIG}$ is the generalized inverse Gaussian distribution.

The full conditional distribution of $\phi_h^{(w)}$ with $z_h^{(w)} = 0$ is

$$\phi_h^{(w)}|- \sim \mathcal{GIG}\left(c - p_w/2, 2\tau_h^{(w)}, \sum_{j=1}^{p_w}(\lambda_{jh}^{(w)})^2\right).$$

The full conditional distributions of the remaining parameters are

$$\tau_h^{(w)}|- \sim Ga(c + d, \phi_h^{(w)} + \eta^{(w)}),$$

$$\eta^{(w)}|- \sim Ga\left(kd+e, \gamma^{(w)} + \sum_{h=1}^{k} \tau_h^{(w)}\right),$$
$$\gamma^{(w)}|- \sim Ga(e+f, \eta^{(w)} + \nu),$$
$$\pi^{(w)}|- \sim beta\left(1 + \sum_{h=1}^{k} z_h^{(w)}, 1 + k - \sum_{h=1}^{k} z_h^{(w)}\right).$$

The full conditional distribution of $z_h^{(w)}$ is

$$\Pr(z_h^{(w)} = 1|-) \propto \pi^{(w)} \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) Ga(\theta_{jh}^{(w)}; a, \delta_{jh}^{(w)}) Ga(\delta_{jh}^{(w)}; b, \phi_h^{(w)}),$$
$$\Pr(z_h^{(w)} = 0|-) \propto (1-\pi^{(w)}) \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \phi_h^{(w)}).$$

We further integrate out $\delta_{jh}^{(w)}$ in $\Pr(z_h^{(w)} = 1|-)$:

$$\begin{aligned}\Pr(z_h^{(w)} = 1|-) &\propto \pi^{(w)} \prod_{j=1}^{p_w} \int \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) Ga(\theta_{jh}^{(w)}; a, \delta_{jh}^{(w)}) Ga(\delta_{jh}^{(w)}; b, \phi_h^{(w)}) d\delta_{jh}^{(w)} \\ &= \pi^{(w)} \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{(\theta_{jh}^{(w)})^{a-1} (\theta_h^{(w)})^b}{(\theta_{jh}^{(w)} + \phi_h^{(w)})^{a+b}}.\end{aligned}$$

The full conditional distribution of σ_j^{-2} for $j = 1, \ldots, p$ is

$$\sigma_j^{-2}|- \sim Ga\left(n/2 + a_\sigma, \frac{1}{2}(\boldsymbol{y}_{j\cdot} - \boldsymbol{\lambda}_{j\cdot}\boldsymbol{X})(\boldsymbol{y}_{j\cdot} - \boldsymbol{\lambda}_{j\cdot}\boldsymbol{X})^T + b_\sigma\right).$$

Appendix B. Variational expectation maximization (EM) algorithm for MAP estimates

Expectation Step: Given model parameters, the distribution of latent factor $\boldsymbol{X}$ was written in Appendix A (Equation 14). The expected sufficient statistics of $\boldsymbol{X}$ is

$$\langle \boldsymbol{x}_{\cdot i} \rangle = (\boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda} + \boldsymbol{I})^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{y}_{\cdot i}, \tag{15}$$
$$\langle \boldsymbol{x}_{\cdot i} \boldsymbol{x}_{\cdot i}^T \rangle = \langle \boldsymbol{x}_{\cdot i} \rangle \langle \boldsymbol{x}_{\cdot i} \rangle^T + (\boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda} + \boldsymbol{I})^{-1}. \tag{16}$$

The expectation of the indicator variable $\rho_h^{(w)} = \langle z_h^{(w)} \rangle$ is

$$\rho_h^{(w)} = \frac{\pi^{(w)} \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) Ga(\theta_{jh}^{(w)}; a, \delta_{jh}^{(w)}) Ga(\delta_{jh}^{(w)}; b, \phi_h^{(w)})}{(1-\pi^{(w)}) \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \phi_h^{(w)}) + \pi^{(w)} \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) Ga(\theta_{jh}^{(w)}; a, \delta_{jh}^{(w)}) Ga(\delta_{jh}^{(w)}; b, \phi_h^{(w)})}.$$

Maximization Step: The log posterior of $\boldsymbol{\Lambda}$ is written as

$$\log(p(\boldsymbol{\Lambda}|-)) \propto \mathrm{tr}\left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda} \boldsymbol{S}^{XY}\right) - \frac{1}{2} \mathrm{tr}\left(\boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda} \boldsymbol{S}^{XX}\right) - \frac{1}{2} \sum_{h=1}^{k} \boldsymbol{\lambda}_{\cdot h}^T \boldsymbol{D}_h \boldsymbol{\lambda}_{\cdot h},$$

where

$$\boldsymbol{D}_h = \text{diag}\left(\frac{\rho_h^{(1)}}{\theta_{1h}^{(1)}} + \frac{1-\rho_h^{(1)}}{\phi_h^{(1)}}, \ldots, \frac{\rho_h^{(m)}}{\theta_{p_m h}^{(m)}} + \frac{1-\rho_h^{(m)}}{\phi_h^{(m)}}\right),$$
$$\boldsymbol{S}^{XY} = \sum_{i=1}^{n} \langle \boldsymbol{x}_{\cdot i} \rangle \boldsymbol{y}_{\cdot i}^T, \text{ and } \boldsymbol{S}^{XX} = \sum_{i=1}^{n} \langle \boldsymbol{x}_{\cdot i} \boldsymbol{x}_{\cdot i}^T \rangle.$$

We take the derivative with respect to the loading column $\boldsymbol{\lambda}_{\cdot h}$ to get the MAP estimate. The derivative of first part in the right hand side is

$$\begin{aligned}\frac{\partial \text{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\boldsymbol{S}^{XY})}{\partial \boldsymbol{\lambda}_{\cdot h}} &= (\mathbf{1}_k^h \otimes \boldsymbol{I}_p) \times \text{vec}[\boldsymbol{\Sigma}^{-1}\boldsymbol{S}^{YX}] = \text{vec}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{S}^{YX}\mathbf{1}_k^h\right) \\ &= \boldsymbol{\Sigma}^{-1}\boldsymbol{S}^{YX}\mathbf{1}_k^h,\end{aligned}$$

where vec is the vectorization of a matrix, $\mathbf{1}_k^h \in \mathbb{R}^{k\times 1}$ is a zero vector with a single 1 in the h^{th} element, and $\boldsymbol{S}^{YX} = (\boldsymbol{S}^{XY})^T$. For the second part

$$\begin{aligned}\frac{\partial \text{tr}(\boldsymbol{\Lambda}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\boldsymbol{S}^{XX})}{\partial \boldsymbol{\lambda}_{\cdot h}} &= 2(\mathbf{1}_k^h \otimes \boldsymbol{I}_p) \times \text{vec}[\boldsymbol{\Sigma}^{-1}\Lambda\boldsymbol{S}^{XX}] = 2 \times \text{vec}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\boldsymbol{S}^{XX}\mathbf{1}_k^h\right) \\ &= 2\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\boldsymbol{S}^{XX}\mathbf{1}_k^h.\end{aligned}$$

For the third part, the derivative is $\boldsymbol{D}_h\boldsymbol{\lambda}_{\cdot h}$. The MAP estimates for $\boldsymbol{\lambda}_{\cdot h}$ are found by setting the derivative to zero:

$$\hat{\boldsymbol{\lambda}}_{\cdot h} = [\boldsymbol{S}_{hh}^{XX}\boldsymbol{I}_p + \boldsymbol{\Sigma}\boldsymbol{D}_h]^{-1}\left(\boldsymbol{S}_{\cdot h}^{YX} - \sum_{h' \neq h} \boldsymbol{\lambda}_{\cdot h'}\boldsymbol{S}_{h'h}^{XX}\right),$$

where $\boldsymbol{S}_{ij}^{XX}$ is the $(i,j)^{th}$ element of $\boldsymbol{S}^{XX}$, and $\boldsymbol{S}_{\cdot h}^{YX}$ is the h^{th} column of $\boldsymbol{S}^{YX}$. The matrix inverse is for a diagonal matrix; thus $\hat{\boldsymbol{\lambda}}_{\cdot h}$ can be calculated efficiently. The MAP estimate for the other model parameters are found from their full conditional distributions with the latent variables replaced by their expectations. We list the parameter updates for those variables here

$$\begin{aligned}\hat{\theta}_{jh}^{(w)} &= \frac{2a - 3 + \sqrt{(2a-3)^2 + 8(\lambda_{jh}^{(w)})^2\delta_{jh}^{(w)}}}{4\delta_{jh}^{(w)}}, \\ \hat{\delta}_{jh}^{(w)} &= \frac{a+b}{\theta_{jh}^{(w)} + \phi_h^{(w)}}, \\ \hat{\phi}_h^{(w)} &= \frac{p' - 1 + \sqrt{(p'-1)^2 + a'b'}}{a'}, \text{with} \\ p' &= \rho_h^{(w)} p_w b - (1-\rho_h^{(w)})p_w/2 + c, \\ a' &= 2(\rho_h^{(w)} \sum_{j=1}^{p_w} \delta_{jh}^{(w)} + \tau_h^{(w)}),\end{aligned}$$

$$
\begin{aligned}
b' &= (1-\rho_h^{(w)})\sum_{j=1}^{p_w}(\lambda_{jh}^{(w)})^2 \\
\hat{\tau}_h^{(w)} &= \frac{c+d}{\phi_h^{(w)}+\eta^{(w)}}, \\
\hat{\eta}^{(w)} &= \frac{dk+e}{\gamma^{(w)}+\sum_{h=1}^{k}\tau_h^{(w)}}, \\
\hat{\gamma}^{(w)} &= \frac{e+f}{\eta^{(w)}+\nu}, \\
\hat{\pi}^{(w)} &= \frac{\sum_{h=1}^{k}\rho_h^{(w)}}{k}, \\
\hat{\sigma}_j^{-2} &= \frac{n/2+a_\sigma-1}{1/2(\boldsymbol{y}_{j\cdot}-\boldsymbol{\lambda}_{j\cdot}\langle\boldsymbol{X}\rangle)(\boldsymbol{y}_{j\cdot}-\boldsymbol{\lambda}_{j\cdot}\langle\boldsymbol{X}\rangle)^T+b_\sigma}.
\end{aligned}
$$

Appendix C. Parameter-expanded EM (PX-EM) algorithm for robust MAP estimates

We introduce a positive semidefinite matrix $\boldsymbol{R}$ in our original model to obtain a parameter-expanded version:

$$
\begin{aligned}
\boldsymbol{y}_i &= \boldsymbol{\Lambda}\boldsymbol{R}_L^{-1}\boldsymbol{x}_i+\boldsymbol{\epsilon}_i, \\
\boldsymbol{x}_i &\sim \mathcal{N}_k(\boldsymbol{0},\boldsymbol{R}), \\
\boldsymbol{\epsilon}_i &\sim \mathcal{N}_k(\boldsymbol{0},\boldsymbol{\Sigma}).
\end{aligned}
$$

Here, $\boldsymbol{R}_L$ is the lower triangular part of the Cholesky decomposition of $\boldsymbol{R}$. Marginally, the covariance matrix is still $\boldsymbol{\Omega}=\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T+\boldsymbol{\Sigma}$, as this additional parameter keeps the likelihood invariant. This additional parameter reduces the coupling effects between the updates of loading matrix and latent factors (Liu et al., 1998; Dyk and Meng, 2001) and serves to connect different posterior modes with equal likelihood curves indexed by $\boldsymbol{R}$ (Rocková and George, 2015).

Let $\boldsymbol{\Lambda}^\star=\boldsymbol{\Lambda}\boldsymbol{R}_L^{-1}$ and $\boldsymbol{\Xi}^\star=\{\boldsymbol{\Lambda}^\star,\boldsymbol{\Theta},\boldsymbol{\Delta},\boldsymbol{\Phi},\boldsymbol{T},\boldsymbol{\eta},\boldsymbol{\gamma},\boldsymbol{\pi},\boldsymbol{\Sigma}\}$. Then the parameters of our expanded model are $\{\boldsymbol{\Xi}^\star\cup\boldsymbol{R}\}$. We assign our structured prior on $\boldsymbol{\Lambda}^\star$. Thus, the updates of $\boldsymbol{\Xi}^\star$ are unchanged given the estimates of the first and second moments of $\boldsymbol{X}$. The estimates of $\langle\boldsymbol{X}\rangle$ and $\langle\boldsymbol{X}\boldsymbol{X}^T\rangle$ are calculated using Equations (15 and 16) in Appendix B after mapping the loading matrix back to the original matrix: $\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^\star\boldsymbol{R}_L$. It remains to estimate $\boldsymbol{R}$.

Write the expected complete log likelihood in the expanded model as

$$
Q(\boldsymbol{\Xi}^\star,\boldsymbol{R}|\boldsymbol{\Xi}_{(s)})=\mathbb{E}_{\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\Xi}_{(s)},\boldsymbol{Y},\boldsymbol{R}_0}\log\big(p(\boldsymbol{\Xi}^\star,\boldsymbol{R},\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{Y})\big).
$$

The only term involving $\boldsymbol{R}$ is $p(\boldsymbol{X})$. Therefore, the $\boldsymbol{R}$ that maximizes this function is

$$
\boldsymbol{R}_{(s)}=\arg\max_{\boldsymbol{R}}Q(\boldsymbol{\Xi}^\star,\boldsymbol{R}|\boldsymbol{\Xi}_{(s)})=\arg\max_{\boldsymbol{R}}\left(\text{const}-\frac{n}{2}\log|\boldsymbol{R}|-\frac{1}{2}\text{tr}\big(\boldsymbol{R}^{-1}\boldsymbol{S}^{XX}\big)\right).
$$

The solution is $\boldsymbol{R}_{(s)}=\frac{1}{n}\boldsymbol{S}^{XX}$.

The EM algorithm in this parameter-expanded space generates the sequence $\{\boldsymbol{\Xi}^{\star}{}_{(1)} \cup \boldsymbol{R}_{(1)}, \boldsymbol{\Xi}^{\star}{}_{(2)} \cup \boldsymbol{R}_{(2)}, \dots\}$. This sequence corresponds to a sequence of parameter estimates in the original space $\{\boldsymbol{\Xi}_{(1)}, \boldsymbol{\Xi}_{(2)}, \dots\}$, where $\boldsymbol{\Lambda}$ in the original space is equal to $\boldsymbol{\Lambda}^{\star}\boldsymbol{R}_L$ (Rocková and George, 2015). We initialize $\boldsymbol{R}_{(0)} = \boldsymbol{I}_k$.

References

Cédric Archambeau and Francis R. Bach. Sparse probabilistic projections. In *Advances in Neural Information Processing Systems 21*, pages 73–80, 2009.

Artin Armagan, Merlise Clyde, and David B. Dunson. Generalized beta mixtures of Gaussians. In *Advances in Neural Information Processing Systems 24*, pages 523–531, 2011.

Artin Armagan, David B. Dunson, and Jaeyong Lee. Generalized double Pareto shrinkage. *Statistica Sinica*, 23(1):119, 2013.

Francis R. Bach and Michael I. Jordan. A probabilistic interpretation of canonical correlation analysis. *Technical Report 688, Department of Statistics, University of California, Berkeley*, 2005.

Anirban Bhattacharya and David B. Dunson. Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306, 2011.

Anirban Bhattacharya, Debdeep Pati, Natesh S. Pillai, and David B. Dunson. Dirichlet-Laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, Accepted for publication, 2014.

Christopher D. Brown, Lara M. Mangravite, and Barbara E. Engelhardt. Integrative modeling of eQTLs and cis-regulatory elements suggests mechanisms underlying cell type specificity of eQTLs. *PLoS Genetics*, 9(8):e1003649, 2013.

Michael W. Browne. The maximum-likelihood solution in inter-battery factor analysis. *British Journal of Mathematical and Statistical Psychology*, 32(1):75–86, 1979.

Michael W. Browne. Factor analysis of multiple batteries by maximum likelihood. *British Journal of Mathematical and Statistical Psychology*, 33(2):184–199, 1980.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.

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), 2008.

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling sparsity via the horseshoe. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 73–80, 2009.

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Sparse and low-rank matrix decompositions. In *47th Annual Allerton Conference on Communication, Control, and Computing*, pages 962–967, 2009.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2): 572–596, 2011.

Pierre Comon. Independent component analysis, A new concept? *Signal Processing*, 36(3): 287–314, 1994.

John P Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16, 2015.

Andreas Damianou, Carl Ek, Michalis Titsias, and Neil Lawrence. Manifold relevance determination. In *29th International Conference on Machine Learning*, pages 145–152, 2012.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.

David A. van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

David Edwards. *Introduction to Graphical Modelling*. Springer, New York, 2nd edition, June 2000. ISBN 9780387950549.

Carl Henrik Ek, Jon Rihan, Philip H.S. Torr, Grégory Rogez, and Neil D. Lawrence. Ambiguity modeling in latent spaces. In *Machine Learning for Multimodal Interaction*, pages 62–73. Springer, 2008.

Barbara E. Engelhardt and Ryan P. Adams. Bayesian structured sparsity from Gaussian fields. *arXiv:1407.2235*, 2014.

Barbara E. Engelhardt and Matthew Stephens. Analysis of population structure: A unifying framework and novel methods based on sparse factor analysis. *PLoS Genetics*, 6(9): e1001117, 2010.

Chuan Gao, Christopher D. Brown, and Barbara E. Engelhardt. A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects. *arXiv:1310.4792*, 2013.

Chuan Gao, Shiwen Zhao, Ian C. McDowell, Christopher D. Brown, and Barbara E. Engelhardt. Differential gene co-expression networks via Bayesian biclustering models. *arXiv:1411.1997*, 2014.

Ignacio González, Sébastien Déjean, Pascal G.P. Martin, and Alain Baccini. CCA: An R package to extend canonical correlation analysis. *Journal of Statistical Software*, 23(12): 1–14, 2008.

Thomas L. Griffiths and Zoubin Ghahramani. The Indian buffet process: An introduction and review. *The Journal of Machine Learning Research*, 12:1185–1224, 2011.

Chris Hans. Bayesian lasso regression. *Biometrika*, 96(4):835–845, 2009.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.

Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 366–373, 2010.

Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 12:2777–2824, 2011.

Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Factorized latent spaces with structured sparsity. In *Advances in Neural Information Processing Systems 23*, pages 982–990, 2010.

Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TF-IDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 143–151, 1997.

Suleiman A. Khan, Seppo Virtanen, Olli P. Kallioniemi, Krister Wennerberg, Antti Poso, and Samuel Kaski. Identification of structural features in chemicals associated with cancer drug response: A systematic data-driven analysis. *Bioinformatics*, 30(17):i497–i504, 2014.

Arto Klami. Polya-gamma augmentations for factor models. In *The 6th Asian Conference on Machine Learning*, pages 112–128, 2014.

Arto Klami and Samuel Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72(1):39–46, 2008.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14:965–1003, 2013.

Arto Klami, Guillaume Bouchard, and Abhishek Tripathi. Group-sparse embeddings in collective matrix factorization. In *International Conference on Learning Representations*, 2014a.

Arto Klami, Seppo Virtanen, Eemeli Leppaaho, and Samuel Kaski. Group factor analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):2136–2147, 2014b.

David Knowles and Zoubin Ghahramani. Nonparametric Bayesian sparse factor models with application to gene expression modeling. *The Annals of Applied Statistics*, 5(2B): 1534–1552, 2011.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, first edition, July 2009.

Matthieu Kowalski. Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis*, 27(3):303–324, 2009.

Matthieu Kowalski and Bruno Torrésani. Structured sparsity: From mixed norms to structured shrinkage. In *Processing with Adaptive Sparse Structured Representations*, 2009.

Minjung Kyung, Jeff Gill, Malay Ghosh, and George Casella. Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis*, 5(2):369–411, 2010.

Neil Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.

Jeffrey T. Leek, Robert B. Scharpf, Hector Corrada Bravo, David Simcha, Benjamin Langmead, W. Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010.

Ker-Chau Li. Genome-wide coexpression dynamics: Theory and application. *Proceedings of the National Academy of Sciences*, 99(26):16875–16880, 2002.

Chuanhai Liu, Donald B. Rubin, and Ying Nian Wu. Parameter expansion to accelerate EM: The PX-EM algorithm. *Biometrika*, 85(4):755–770, 1998.

Joseph E. Lucas, Hsiu-Ni Kung, and Jen-Tsan A. Chi. Latent factor analysis to discover pathway-associated putative segmental aneuploidies in human cancers. *PLoS Computational Biology*, 6(9):e1000920, 2010.

Haisu Ma, Eric E. Schadt, Lee M. Kaplan, and Hongyu Zhao. COSINE: Condition-specific sub-network identification using a global optimization method. *Bioinformatics*, 27(9): 1290–1298, 2011.

Lara M. Mangravite, Barbara E. Engelhardt, Marisa W. Medina, Joshua D. Smith, Christopher D. Brown, Daniel I. Chasman, Brigham H. Mecham, Bryan Howie, Heejung Shim, Devesh Naidoo, et al. A statin-dependent QTL for *GATM* expression is associated with statin-induced myopathy. *Nature*, 502(7471):377–380, 2013.

Roderick P. McDonald. Three common factor models for groups of variables. *Psychometrika*, 35(1):111–128, 1970.

Toby J. Mitchell and John J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

Radford M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

Anthony O'Hagan. On outlier rejection phenomena in Bayes inference. *Journal of the Royal Statistical Society. Series B*, 41(3):358–367, 1979.

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

Elena Parkhomenko, David Tritchler, and Joseph Beyene. Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology*, 8(1):1–34, 2009.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

Nicholas G. Polson and James G. Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. In *Bayesian Statistics 9, eds. J.M. Bernardo et al.*, pages 501–538. Oxford University Press, 2011.

Iosifina Pournara and Lorenz Wernisch. Factor analysis for gene regulatory networks and transcription factor activity profiles. *BMC Bioinformatics*, 8:61, 2007.

Iulian Pruteanu-Malinici, Daniel L. Mace, and Uwe Ohler. Automatic annotation of spatial expression patterns via Bayesian factor models. *PLoS Computational Biology*, 7(7): e1002098, 2011.

Xinquan Qu and Xinlei Chen. Sparse structured probabilistic projections for factorized latent spaces. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1389–1394, 2011.

Priyadip Ray, Lingling Zheng, Joseph Lucas, and Lawrence Carin. Bayesian joint analysis of heterogeneous genomics data. *Bioinformatics*, 30(10):1370–1376, 2014.

Veronika Rocková and Edward I. George. Fast Bayesian factor analysis via automatic rotations to sparsity. *Journal of the American Statistical Association*, 2015.

Justin K. Romberg, Hyeokho Choi, and Richard G. Baraniuk. Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. *IEEE Transactions on Image Processing*, 10(7):1056–1068, 2001.

Sam Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems 10*, pages 626–632, 1998.

Konstantin Salomatin, Yiming Yang, and Abhimanyu Lad. Multi-field correlated topic modeling. In *SIAM International Conference on Data Mining*, pages 628–637, 2009.

Mathieu Salzmann, Carl H. Ek, Raquel Urtasun, and Trevor Darrell. Factorized orthogonal latent spaces. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 701–708, 2010.

Juliane Schäfer and Korbinian Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

Aaron Shon, Keith Grochow, Aaron Hertzmann, and Rajesh P. Rao. Learning shared latent structure for image synthesis and robotic imitation. In *Advances in Neural Information Processing Systems 18*, pages 1233–1240, 2005.

Tommi Suvitaival, Juuso A. Parkkinen, Seppo Virtanen, and Samuel Kaski. Cross-organism toxicogenomics with group factor analysis. *Systems Biomedicine*, 2:e29291, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

Michael E. Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

Michael E. Tipping and Christopher M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999a.

Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B*, 61(3):611–622, 1999b.

Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. Mulan: A Java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

Seppo Virtanen, Arto Klami, and Samuel Kaski. Bayesian CCA via group sparsity. In *Proceedings of the 28th International Conference on Machine Learning*, pages 457–464, 2011.

Seppo Virtanen, Arto Klami, Suleiman A. Khan, and Samuel Kaski. Bayesian group factor analysis. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22, pages 1269–1277, 2012.

Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

Mike West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 1987.

Mike West. Bayesian factor regression models in the "large p, small n" paradigm. In *Bayesian Statistics 7, eds. J.M. Bernardo et al.*, pages 723–732. Oxford University Press, 2003.

Daniela Witten, Rob Tibshirani, Sam Gross, and Balasubramanian Narasimhan. *PMA: Penalized Multivariate Analysis*, 2013. URL `http://CRAN.R-project.org/package=PMA`. R package version 1.0.9.

Daniela M. Witten and Robert J. Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genetics and Molecular Biology*, 8(1):1–27, 2009.

Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

Anqi Wu, Mijung Park, Oluwasanmi O Koyejo, and Jonathan W Pillow. Sparse Bayesian structure learning with "dependent relevance determination priors. In *Advances in Neural Information Processing Systems*, pages 1628–1636, 2014.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.

Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 37(6A):3468–3497, 2009.

Shiwen Zhao and Shao Li. A co-module approach for elucidating drug-disease associations and revealing their molecular basis. *Bioinformatics*, 28(7):955–961, 2012.

Tianyi Zhou, Dacheng Tao, and Xindong Wu. Manifold elastic net: A unified framework for sparse dimension reduction. *Data Mining and Knowledge Discovery*, 22(3):340–371, 2011.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B*, 67(2):301–320, 2005.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

James Y Zou, Daniel J Hsu, David C Parkes, and Ryan P Adams. Contrastive learning using spectral methods. In *Advances in Neural Information Processing Systems*, pages 2238–2246, 2013.