



Spike and slab Bayesian sparse principal component analysis

Yu-Chien Bo Ning¹ · Ning Ning²

Received: 6 August 2023 / Accepted: 28 March 2024 / Published online: 13 May 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Sparse principal component analysis (SPCA) is a popular tool for dimensionality reduction in high-dimensional data. However, there is still a lack of theoretically justified Bayesian SPCA methods that can scale well computationally. One of the major challenges in Bayesian SPCA is selecting an appropriate prior for the loadings matrix, considering that principal components are mutually orthogonal. We propose a novel parameter-expanded coordinate ascent variational inference (PX-CAVI) algorithm. This algorithm utilizes a spike and slab prior, which incorporates parameter expansion to cope with the orthogonality constraint. Besides comparing to two popular SPCA approaches, we introduce the PX-EM algorithm as an EM analogue to the PX-CAVI algorithm for comparison. Through extensive numerical simulations, we demonstrate that the PX-CAVI algorithm outperforms these SPCA approaches, showcasing its superiority in terms of performance. We study the posterior contraction rate of the variational posterior, providing a novel contribution to the existing literature. The PX-CAVI algorithm is then applied to study a lung cancer gene expression dataset. The R package `VBsparsePCA` with an implementation of the algorithm is available on the Comprehensive R Archive Network (CRAN).

Keywords Bayesian SPCA · Spike and slab prior · Variational inference · Parameter expansion

Mathematics Subject Classification Primary 62C10 · 62H25 · 62J07

1 Introduction

Sparse Principal Component Analysis (SPCA), a contemporary variant of PCA, has gained popularity as a valuable tool for reducing the dimensions of high-dimensional data. Its applications span various fields, such as chemistry, where it aids in identifying crucial chemical components from spectra (Varmuza and Filzmoser 2009); genetics, where it helps discover significant genes and pathways (Li et al. 2017); and macroeconomics, where it plays a role in selecting dominant macro variables that earn substantial risk premiums (Rapach and Zhou 2019). The success of SPCA can be attributed to two main factors. Firstly, in typical high-dimensional datasets, the number of input variables p is greater than the

number of observations n . This condition poses challenges when using traditional PCA, as the leading eigenvector becomes inconsistently estimated when p/n does not converge to 0 (Paul 2007; Johnstone and Lu 2009). However, SPCA addresses and mitigates this issue effectively. Secondly, the principal components derived from SPCA are linear combinations of only a few important variables, making them highly interpretable in practical applications. This interpretability makes SPCA a valuable asset when dealing with complex data sets, enabling researchers and analysts to glean meaningful insights with ease.

Several SPCA algorithms have been proposed, and interested readers can refer to Zou and Xue (2018) for a comprehensive literature review on these algorithms. However, it's worth noting that the algorithms discussed in that review do not include Bayesian methods. Recently, two Bayesian SPCA approaches, introduced by Gao and Zhou (2015) and Xie et al. (2022), have emerged and demonstrated impressive advantages. Both approaches adopt the spiked covariance model, which conveniently represents a linear regression model where the loadings matrix serves as the coefficient and the design matrix follows a standard

✉ Ning Ning
patning@tamu.edu
Yu-Chien Bo Ning
bycning@hsph.harvard.edu

¹ Department of Epidemiology, Harvard T. H. Chan School of Public Health, 677 Huntington Ave, Boston, MA 02115, USA

² Department of Statistics, Texas A&M University, College Station, TX 77843, USA

multivariate normal distribution as the random component. A significant challenge in Bayesian SPCA lies in placing a prior on the coefficients while enforcing the orthogonality constraint, which requires the columns of the loadings matrix to be mutually orthogonal. This constraint needs to be incorporated through the prior distribution. Gao and Zhou (2015) tackled this challenge by constructing a prior that projects the nonzero coordinates onto a subspace spanned by a collection of mutually orthogonal unit vectors. However, their posterior becomes intractable and challenging to compute when the rank is greater than one. Xie et al. (2022) adopted a different approach by reparametrizing the likelihood, multiplying the loadings matrix with an orthogonal matrix to remove the orthogonal constraint. Their prior involves only Laplacian spike and slab densities while our density (see Eq. (3) in Sect. 2.2) is considerably more general. Additionally, this prior can introduce dependence while theirs demand prior independence.

In this paper, we present a novel prior for the coefficient of the spiked covariance model. In our approach, we apply a regular spike and slab prior on the parameter, which is the product of the loadings matrix (the coefficient) and an orthogonal matrix. The orthogonal matrix is then a latent variable in the prior. By marginalizing this joint density, we derive the prior of the coefficient. The spike and slab prior is a mixture of a continuous density and a Dirac measure centered at 0. By introducing an appropriate prior on the mixture weight, one can effectively impose sparsity on the coefficient. The spike and slab prior is widely recognized as one of the most prominent priors for Bayesian high-dimensional analysis and has received extensive study. Excellent works in this area include those by Johnstone and Silverman (2004), Ročková and George (2018), Castillo and Szabó (2020), Castillo and van der Vaart (2012), Castillo et al. (2015), Martin et al. (2017), Qiu et al. (2018), Jammalamadaka et al. (2019), Qiu et al. (2020), Ning (2023b), Jeong and Ghosal (2020), Ohn et al. (2023), Ning (2023a), and Ning et al. (2020). For a comprehensive overview of this topic, readers can refer to the review paper by Banerjee et al. (2021). It is important to note that in our spike and slab formulation, the slab density, which incorporates the latent variable, differs from that in traditional linear regression models. This distinction contributes to the uniqueness and effectiveness of our proposed approach.

We employ a variational approach to compute the posterior, a method that minimizes a chosen distance or divergence (e.g., Kullback-Leibler divergence) between a preselected probability measure, belonging to a rich and analytically tractable class of distributions, and the posterior distribution. This approach offers faster computational speed compared to sampling methods like the Markov chain Monte Carlo algorithm. Among the variational approaches, the coordinate ascent variational inference (CAVI) method stands out as

the most popular algorithm (Blei et al. 2017). Several CAVI methods have been developed for sparse linear regression models with the spike and slab prior (or the subset selection prior) such as Carbonetto and Stephens (2012), Huang et al. (2016), Ray and Szabo (2022), Yang et al. (2020) and sparse factor analysis such as Ghahramani and Beal (1999), Wang et al. (2020), Hansen et al. (2023). Researchers have also studied the theoretical properties of the variational posterior, such as the posterior contraction rate, as examined by Ray and Szabo (2022) and Yang et al. (2020). While variational Bayesian methods for SPCA have been developed by Guan and Dy (2009) and Bouveyron et al. (2018), they did not provide a theoretical justification for their posterior. Moreover, the priors used by Guan and Dy (2009) involving the Laplace distribution and Bouveyron et al. (2018)'s prior, similar to the spike and slab prior with a fixed mixture weight, are known not to yield the optimal (or near-optimal) posterior contraction rate.

In this paper, we show that the contraction rates of both the posterior and the variational posterior are nearly optimal. To the best of our knowledge, this is the first result for the variational Bayesian method applied to SPCA. Additionally, we develop an EM algorithm tailored for SPCA, in which the maximum of a posteriori estimator is obtained. The EM algorithm for Bayesian variable selection has been extensively studied for the sparse linear regression model by Ročková and George (2014), Ročková and George (2018). Similar algorithms have been developed for other high-dimensional models, such as the dynamic time series model (Ning et al. 2019) and the sparse factor model (Ročková and George 2016). For our EM algorithm to accommodate SPCA, we replace the Dirac measure in the spike and slab prior with a continuous density, resulting in the continuous spike and slab prior. Both the variational approach and the EM algorithm employ parameter expansion techniques on the likelihood function. Consequently, these algorithms are referred to as the PX-CAVI and the PX-EM algorithm respectively, where PX means parameter expanded. The parameter expansion approach was initially proposed by Liu et al. (1998) and has proven effective in accelerating the convergence speed of the EM algorithm. Additionally, we discovered that by selecting the expanded parameter as the orthogonal matrix, we can circumvent the need to handle the orthogonal constraint directly on the loading matrix. This approach allows us to first solve for the unconstrained matrix and subsequently apply singular value decomposition (SVD) to obtain an estimated value for the loadings matrix. This simplification streamlines the computation process and enhances the efficiency of our algorithms.

The remainder of this paper is structured as follows: Sect. 2 presents the model and the prior used in this study. Section 3 introduces the variational approach and outlines the development of the PX-CAVI algorithm. In Sect. 4, we delve into

the theoretical properties of both the posterior and the variational posterior. Section 5 presents the PX-EM algorithm we developed. To evaluate the performance of our algorithms, we conduct simulation studies in Sect. 6. Furthermore, in Sect. 7, we analyze a lung cancer gene dataset to illustrate the application of our approach in real-world scenarios. The appendix contains proofs of the equations presented in Sect. 3. Proofs of the theorems discussed in Sect. 4 and the batch PX-CAVI algorithm without relying on the jointly row-sparsity assumption are provided in the supplementary material. For readers interested in implementing our algorithms, we have made the VBsparsePCA package available on the comprehensive R archive network (CRAN). This package includes both the PX-CAVI algorithm and the batch PX-CAVI algorithm.

2 Model and priors

In this section, we begin by introducing the spiked covariance model, followed by the spike and slab prior applied.

2.1 The spiked covariance model

Consider the spiked covariance model

$$X_i = \theta w_i + \sigma \epsilon_i, \quad w_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_r), \quad \epsilon_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_p), \tag{1}$$

where X_i is a p -dimensional vector, θ is a $p \times r$ -dimensional loadings matrix, w_i is a r -dimensional vector, ϵ_i is a p -dimensional vector that is independent of w_i , and r is the rank. We denote θ_k as the k -th column of θ . The orthogonality constraint of θ requires that $\langle \theta_k, \theta_{k'} \rangle = 0$ for any $k \neq k', k, k' \in \{1, \dots, r\}$. The model is equivalent to $X_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma)$, where $\Sigma = \theta \theta' + \sigma^2 I_p$. Let $\theta = U \Lambda^{1/2}$, where $U = (\theta_{\cdot 1} / \|\theta_{\cdot 1}\|_2, \dots, \theta_{\cdot r} / \|\theta_{\cdot r}\|_2)$ is a $p \times r$ matrix containing the first r eigenvectors and $\Lambda = \text{diag}(\|\theta_{\cdot 1}\|_2^2, \dots, \|\theta_{\cdot r}\|_2^2)$ is an $r \times r$ diagonal matrix. Then, $\Sigma = U \Lambda U' + \sigma^2 I_p$. One can easily check that the k -th eigenvalue of Σ is $\|\theta_{\cdot k}\|_2^2 + \sigma^2$ if $k \leq r$ and is σ^2 if $k > r$. We assume $p \gg n$ (i.e. $n/p \rightarrow 0$) and θ is jointly row-sparse—that is, within the same row, the coordinates are either all zero or all non-zero. We define the rows containing non-zero entries as “non-zero rows” and the remaining rows as “zero rows.” With this assumption, the support of each column in θ remains the same and is denoted as $S = \{j \in \{1, \dots, p\} : \theta'_j \neq 0_r\}$ where 0_r represents r -dimensional zero vector. Adopting the row-sparsity assumption is convenient for practitioners, as the principal subspace is generated by the same sparse set of features. Additionally, we can simplify our main ideas and use more concise notations by adopting this assumption, as the support is consistent across all principal components. A more general

assumption that allows the support to vary across principal components, is covered in the supplementary material. Our R package VBsparsePCA can effectively handle both assumptions.

2.2 The spike and slab prior

We introduce our spike and slab prior, which is

$$\pi(\theta, \boldsymbol{\gamma} | \lambda_1, r) \propto \prod_{j=1}^p \left[\gamma_j \int_{A \in V_{r,r}} g(\theta_j | \lambda_1, A, r) \pi(A) dA + (1 - \gamma_j) \delta_0(\theta_j) \right], \tag{2}$$

where $V_{r,r} = \{A \in \mathbb{R}^{r \times r} : A' A = I_r\}$ is the Stiefel manifold of r -frames in \mathbb{R}^r and δ_0 is the Dirac measure at zero. Our idea of constructing the prior (2) is that since $\beta = \theta A$ does not have the orthogonality constraint, as A is an orthogonal matrix, we first apply the regular spike and slab prior on β which could be viewed as the joint distribution of θ and A . We then obtain the prior of θ by marginalizing the parameter A from the joint distribution of θ and A . Because of the latent variable A , this prior is different from those in the sparse linear regression models. We consider a general expression for the density g , which is

$$g(\theta_j | \lambda_1, A, r) = [C(\lambda_1)]^r \exp(-\lambda_1 \|\beta_j\|_q^m), \tag{3}$$

where $1 \leq q \leq 2$, $m \in \{1, 2\}$, and $[C(\lambda_1)]^r$ is the normalizing constant. This expression includes three common distributions as special cases. If $q = 1$ and $m = 1$, $C(\lambda_1) = \lambda_1/2$, then g is a product of r -independent Laplace densities. If $q = 2$ and $m = 2$, $C(\lambda_1) = \sqrt{\lambda_1/(2\pi)}$, then it is the multivariate normal density. If $q = 2$ and $m = 1$, $C(\lambda_1) = \lambda_1/a_r$ where

$$a_r = \sqrt{\pi} (\Gamma(r + 1) / \Gamma(r/2 + 1))^{1/r} > 2,$$

then it is the density part of the prior introduced by Ning et al. (2020) for group sparsity. The priors for the remaining parameters are given as follows: $\pi(A) \propto 1$ and for each j ,

$$\gamma_j | \kappa \sim \text{Bernoulli}(\kappa), \quad \kappa \sim \text{Beta}(\alpha_1, \alpha_2). \tag{4}$$

If σ^2 and r are unknown, we let $\sigma^2 \sim \text{InverseGamma}(\sigma_a, \sigma_b)$ and $r \sim \text{Poisson}(x)$. Assuming r is fixed, then the joint posterior distribution of $(\theta, \boldsymbol{\gamma}, \sigma^2)$ is

$$\pi(\theta, \boldsymbol{\gamma}, \sigma^2 | X) \propto \prod_{i=1}^n f(X_i | \theta, \sigma^2, r) \prod_{j=1}^p \pi(\theta_j | \gamma_j, r) \times \left(\int \pi(\gamma_j | \kappa) \pi(\kappa) d\kappa \right) \pi(\sigma^2), \tag{5}$$

where $X = (X_1, \dots, X_n)$ with each X_i being a p -dimensional vector.

3 Variational inference

In this section, we propose a variational approach for SPCA using the posterior (5). We introduce a mean-field variational class to obtain the variational posterior, and then develop the PX-CAVI algorithm to efficiently compute it.

3.1 The variational posterior and the evidence lower bound

To obtain the variational posterior, we adopt the mean-field variational approximation, which decomposes the posterior into several independent components, with the parameter in each component being independent of the others. The variational class is defined as follows:

$$\mathcal{P}^{\text{MF}} = \left\{ P(\theta) := \prod_{j=1}^p \left[z_j \mathcal{N}(\mu_j, \sigma^2 M_j) + (1 - z_j) \delta_0 \right], \right. \\ \left. \begin{aligned} \mu_j \in \mathbb{R}^r, \langle \mu_{\cdot k}, \mu_{\cdot k'} \rangle = 0, \forall k \neq k', \\ M_j \in \mathbb{M}^{r \times r}, z_j \in [0, 1] \end{aligned} \right\}, \tag{6}$$

where $\mathbb{M}^{r \times r}$ stands for the space of $r \times r$ positive definite matrices. For any $P(\theta) \in \mathcal{P}^{\text{MF}}$, it is a product of p independent densities, each of which is a mixture of two distributions—a multivariate normal (or a normal density when $r = 1$) and the Dirac measure at zero. The mixture weight z_j is the corresponding inclusion probability. The variational posterior is obtained by minimizing the Kullback-Leibler divergence between all $P(\theta) \in \mathcal{P}^{\text{MF}}$ and the posterior, i.e.,

$$\widehat{P}(\theta) = \arg \min_{P(\theta) \in \mathcal{P}^{\text{MF}}} KL(P(\theta), \pi(\theta|X)), \tag{7}$$

which can be also written as

$$\begin{aligned} \widehat{P}(\theta) &= \arg \min_{P(\theta) \in \mathcal{P}^{\text{MF}}} \left(\mathbb{E}_P \log P(\theta) - \mathbb{E}_P \log \pi(\theta|X) \right) \\ &= \arg \min_{P(\theta) \in \mathcal{P}^{\text{MF}}} \left(\mathbb{E}_P \log P(\theta) - \mathbb{E}_P \log \pi(\theta, X) \right. \\ &\quad \left. + \log \pi(X) \right). \end{aligned} \tag{8}$$

As the expression of $\log \pi(X)$ in (8) is intractable, we define the evidence lower bound (ELBO), which is the lower bound of $\log \pi(X)$ as follows:

$$\text{ELBO}(\theta) = \mathbb{E}_P \log \pi(\theta, X) - \mathbb{E}_P \log P(\theta). \tag{9}$$

and solve $\widehat{P}(\theta_j) = \arg \max_{P(\theta_j) \in \mathcal{P}^{\text{MF}}} \text{ELBO}(\theta)$. Since

$$P(\theta) = \prod_{j=1}^p P(\theta_j) \quad \text{and} \quad \pi(\theta, X) = \prod_{j=1}^p \pi(\theta_j, X),$$

the ELBO can be also written as follows:

$$\text{ELBO}(\theta) = \sum_{j=1}^p \left(\mathbb{E}_P \log \pi(\theta_j, X) - \mathbb{E}_P \log P(\theta_j) \right).$$

From the last display, we can solve each $\widehat{P}(\theta_j)$ independently and then obtain the variational posterior from $\widehat{P}(\theta) = \prod_{j=1}^p \widehat{P}(\theta_j)$.

3.2 The PX-CAVI algorithm

The PX-CAVI algorithm is an iterative method where, in each iteration, it optimizes each of the unknown variables by conditioning on the rest. Our algorithm incorporates two key differences from the conventional CAVI algorithm. Firstly, we include an expectation step, similar to that used in the EM algorithm, since $w = (w_1, \dots, w_n)$ is a random variable. Secondly, we apply parameter expansion to the likelihood, which enables us to handle the orthogonality constraint and accelerate the convergence speed of our algorithm. Now, let's provide a step-by-step derivation of the PX-CAVI algorithm, where $M = (M_1, \dots, M_p)$ and $z = (z_1, \dots, z_p)$.

1. E-step

In this step, the full model posterior is $\pi(\theta, w, X)$. Let $\Theta^{(t)}$ be the estimated value of $\Theta = (\mu, M, z)$ from the t -th iteration, we obtain

$$\begin{aligned} \widehat{P}(w_i|\Theta^{(t)}) &= \mathcal{N}(\widetilde{\omega}_i, \widetilde{V}_w), \\ \widetilde{\omega}_i &= \frac{1}{\sigma^2} \widetilde{V}_w \sum_{j=1}^p z_j^{(t)} [\mu_j^{(t)}]' X_{ij}, \\ \widetilde{V}_w &= \left(\frac{1}{\sigma^2} \sum_{j=1}^p z_j^{(t)} \left([\mu_j^{(t)}]' \mu_j^{(t)} + \sigma^2 M_j^{(t)} \right) + I_r \right)^{-1}. \end{aligned} \tag{10}$$

Then, the objective function is given by

$$Q(\theta|\Theta^{(t)}) = \mathbb{E}_{w|\Theta^{(t)}} \log \pi(\theta, w, X).$$

We obtain

$$\begin{aligned} \widehat{P}(\theta) &= \arg \max_{P(\theta) \in \mathcal{P}^{\text{MF}}} \sum_{j=1}^p \left(\mathbb{E}_P Q(\theta_j|\Theta^{(t)}) - \mathbb{E}_P \log P(\theta_j) \right). \\ &= \arg \max_{P(\theta) \in \mathcal{P}^{\text{MF}}} \sum_{j=1}^p \left(\mathbb{E}_P \left[\mathbb{E}_{w|\Theta^{(t)}} \log \pi(\theta_j, w, X) \right] - \mathbb{E}_P \log P(\theta_j) \right). \end{aligned}$$

2. Parameter expansion

To obtain $\hat{P}(\theta)$, special attention must be given to the orthogonality constraint of μ as defined in (6). This is where the parameter expansion technique is used. Let A be the expanded parameter and denote $\beta = \theta A$, the likelihood after the parameter expansion becomes $X_i = \beta w_i + \sigma^2 \epsilon_i$, as $A w_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_r)$ follows the same distribution as w_i . Then, our spike and slab prior is directly applied on β . We do not require the prior to be invariant under the transformation of the parameter. After solving β , one can obtain θ using the singular value decomposition (SVD). To accelerate the convergence speed of the algorithm, we apply parameter expansion again. At this time, the expanded parameter is chosen to be a positive definite matrix, say D . We denote $\tilde{\beta} = \beta D$. The likelihood after this parameter expansion becomes $X_i = \tilde{\beta} \tilde{w}_i + \sigma \epsilon_i$, where $\tilde{w}_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, D)$ and $\tilde{\beta} = \tilde{\theta} A D_L^{-1}$, D_L is the lower triangular matrix obtained using SVD. Our spike and slab prior is then directly putting on $\tilde{\beta}$. To summarize, parameter expansion is used twice in the PX-CAVI algorithm. The first time is primary used to deal with the orthogonality constraint, and the second time is to accelerate its convergence speed. We denote \tilde{u} and \tilde{M} as the mean and the covariance of $P(\tilde{\beta})$. This leads us to instead maximize $\mathbb{E}_P Q(\tilde{\Theta} | \tilde{\Theta}^{(t)}) - \mathbb{E}_q \log P(\tilde{\beta})$, where $\tilde{\Theta} = (\tilde{u}, \tilde{M}, z)$ and

$$P(\tilde{\beta}) = \prod_{j=1}^p \left[z_j \mathcal{N}(\tilde{u}_j, \sigma^2 \tilde{M}_j) + (1 - z_j) \delta_0 \right]. \tag{11}$$

One can quickly check that $\tilde{u} = \mu A D_L^{-1}$ and $\tilde{M}_j = D_L^{-1} M_j D_L^{-1}$. Note that since we assume θ is jointly row-sparse, the support of β and it of $\tilde{\beta}$ are the same. Thus z_j in (11) is the same as it in (6).

To solve for \tilde{u} and \tilde{M} , we explore the following two choices of the density g in (3):

- When $q = 1$ and $m = 1$, it yields a product of r -independent Laplace densities. Details are given in Appendix A. In summary, denoting $H_i = \tilde{\omega}_i \tilde{\omega}_i' + \tilde{V}_w$, we obtain

$$\hat{\tilde{u}}_j = \min_{\tilde{u}_j} \left[\frac{1}{2\sigma^2} \sum_{i=1}^n (\tilde{u}_j H_i \tilde{u}_j' - 2X_{ij} \tilde{u}_j \tilde{\omega}_i) + \lambda_1 \sum_{k=1}^r f(\tilde{u}_{jk}, \sigma^2 \tilde{M}_{j,kk}) \right] \tag{12}$$

$$\hat{\tilde{M}}_j = \min_{\tilde{M}_j} \left[\frac{1}{2} \sum_{i=1}^n \text{Tr}(\tilde{M}_j H_i) - \frac{\log \det(\tilde{M}_j)}{2} + \lambda_1 \sum_{k=1}^r f(\tilde{u}_{jk}, \sigma^2 \tilde{M}_{j,kk}) \right], \tag{13}$$

where $f(\tilde{u}_{jk}, \sigma^2 \tilde{M}_{j,kk})$ is the mean of the folded normal distribution,

$$f(\tilde{u}_{jk}, \sigma^2 \tilde{M}_{j,kk}) = \sqrt{\frac{2\sigma^2 \tilde{M}_{j,kk}}{\pi}} \exp\left(-\frac{\tilde{u}_{jk}^2}{2\sigma^2 \tilde{M}_{j,kk}}\right) + \tilde{u}_{jk} \left(1 - 2\Phi\left(-\frac{\tilde{u}_{jk}}{\sqrt{\sigma^2 \tilde{M}_{j,kk}}}\right)\right),$$

with Φ being the cumulative distribution function of a standard normal distribution. Here, $\det(B)$ and $\text{Tr}(B)$ stands for the determinant and the trace of the matrix B .

- When $q = 2$ and $m = 2$, it results in a multivariate normal density. If g is the multivariate normal density, we use $\mathcal{N}(0, \sigma^2 I_r / \lambda_1)$ instead, as the solution for σ^2 is simpler. One can consider we choose the tuning parameter to be λ_1 / σ^2 instead of λ_1 . Then, we obtain

$$\hat{\tilde{u}}_j = \hat{\tilde{M}}_j^{-1} \sum_{i=1}^n X_{ij} \tilde{\omega}_i' \quad \text{and} \quad \hat{\tilde{M}}_j = \left(\sum_{i=1}^n (\tilde{\omega}_i \tilde{\omega}_i' + \tilde{V}_w) + \lambda_1 I_r \right)^{-1}. \tag{14}$$

3. Updating z

To solve z , we need to obtain $\hat{h} = (\hat{h}_1, \dots, \hat{h}_p)$, where for each $j \in \{1, \dots, p\}$, $\hat{h}_j = \log(\hat{z}_j / (1 - \hat{z}_j))$. In Appendix A, we derive the solution for \hat{h}_j . If g is the product of r independent Laplace density, then

$$\begin{aligned} \hat{h}_j &= \log\left(\frac{\alpha_1}{\alpha_2}\right) + r \log\left(\frac{\sqrt{\pi} \sigma \lambda_1}{\sqrt{2}}\right) \\ &\quad - \lambda_1 \sum_{k=1}^r f(\tilde{u}_{jk}, \sigma^2 \tilde{M}_{j,kk}) + \frac{\log \det(\tilde{M}_j) + 1}{2} \\ &\quad - \frac{1}{2\sigma^2} \sum_{i=1}^n \left[-2X_{ij} \tilde{u}_j \tilde{\omega}_i + \tilde{u}_j H_i \tilde{u}_j' + \text{Tr}(\sigma^2 \tilde{M}_j H_i) \right]. \end{aligned} \tag{15}$$

If g is the multivariate normal density, then

$$\begin{aligned} \hat{h}_j &= \log\left(\frac{\alpha_1}{\alpha_2}\right) + \frac{r \log \lambda_1}{2} - \frac{\lambda_1}{2} (\tilde{u}_j \tilde{u}_j' + \sigma^2 \text{Tr}(\tilde{M}_j)) \\ &\quad + \frac{\log \det(\tilde{M}_j) + 1}{2} - \frac{1}{2\sigma^2} \sum_{i=1}^n (-2X_{ij} \tilde{u}_j \tilde{\omega}_i' + \tilde{u}_j H_i \tilde{u}_j' \\ &\quad + \text{Tr}(\sigma^2 \tilde{M}_j H_i)). \end{aligned} \tag{16}$$

4. Updating $\hat{\mu}$ and \hat{M}

As we obtained $\hat{\tilde{u}}$ and $\hat{\tilde{M}}$, then $\hat{\mu}$ and \hat{M} can be solved accordingly. Note that $\tilde{w}_i \sim \mathcal{N}(0, D)$. In the E-step, we also obtained $\tilde{\omega}_i$ and V_w . Thus, D can be solved using

$\widehat{D} = \frac{1}{n} \sum_{i=1}^n \widetilde{\omega}_i \widetilde{\omega}'_i + \widetilde{V}_\omega$, and $\widehat{\mu}$ can be obtained by first solving $\widehat{u} = \widehat{u} \widehat{D}_L$. Next, we apply the SVD to obtain \widehat{A} . Last, we obtain μ using $\widehat{\mu} = \widehat{u} \widehat{A}'$. \widehat{M} can be obtained similarly, i.e., $\widehat{M}_j = \widehat{D}_L \widehat{M}_j \widehat{D}'_L$.

5. Updating σ^2

Recall that the prior $\sigma^2 \sim \text{InverseGamma}(\sigma_a, \sigma_b)$. If g is the product of r independent Laplace density, we obtain

$$\begin{aligned} \widehat{\sigma}^2 = \arg \min_{\sigma^2 \in (0, \infty)} & \left[\sum_{j=1}^p z_j \left\{ \frac{1}{2\sigma^2} \sum_{i=1}^n (\widetilde{u}_j H_i \widetilde{u}'_j - 2X_{ij} \widetilde{u}_j \widetilde{\omega}_i) - \frac{r \log \sigma^2}{2} \right. \right. \\ & \left. \left. + \lambda_1 \sum_{k=1}^r f(\widetilde{u}_{jk}, \sigma^2 \widetilde{M}_{j,kk}) \right\} \right. \\ & \left. + \frac{(np + 2\sigma_a + 2) \log \sigma^2}{2} \right. \\ & \left. + \frac{\text{Tr}(X'X) + 2\sigma_b}{2\sigma^2} \right]. \end{aligned} \tag{17}$$

If g is the multivariate normal density, we obtain

$$\widehat{\sigma}^2 = \frac{\text{Tr}(X'X) + \sum_{j=1}^p z_j \sum_{i=1}^n (\widetilde{u}_j H_i \widetilde{u}'_j - 2X_{ij} \widetilde{u}_j \widetilde{\omega}_i + \lambda_1 \widetilde{u}_j \widetilde{u}'_j) + 2\sigma_b}{np + 2(\sigma_a + 1)}. \tag{18}$$

Now, we summarize the PX-CAVI algorithm.

Algorithm 1: The PX-CAVI algorithm

Data: X , a $p \times n$ matrix, scaled and centered

Input: $\widehat{\mu}^{(0)}$, $\widehat{M}^{(0)}$, $\widehat{z}^{(0)}$, $\widehat{\sigma}^{(0)}$, r , number of total iterations T , and the threshold δ

For $t = 0, \dots, T - 1$:

- Update $\widehat{\omega}^{(t+1)}$ and $\widetilde{V}_\omega^{(t+1)}$ using (10)
- If g is the product of r independent Laplace density
 - Update $\widetilde{u}^{(t+1)}$ and $\widetilde{M}^{(t+1)}$ using (12) and (13)
 - Update $h^{(t+1)}$ using (15) and then obtain $\widehat{z}^{(t+1)}$
 - Update $\sigma^{(t+1)}$ from (17)
- If g is the multivariate normal density
 - Update $\widetilde{u}^{(t+1)}$ and $\widetilde{M}^{(t+1)}$ using (14)
 - Update $h^{(t+1)}$ using (16) and then obtain $\widehat{z}^{(t+1)}$
 - Update $\sigma^{(t+1)}$ using (18)
- Obtain $D^{(t+1)}$, $u^{(t+1)}$, and $\widehat{M}^{(t+1)}$
- Apply SVD to obtain $A^{(t+1)}$ and then obtain $\mu^{(t+1)}$

Stop: If $\max \left(\|\mu^{(t+1)} \mu^{(t+1)'} - \mu^{(t)} \mu^{(t)'}\|_F, \|z^{(t+1)} - z^{(t)}\|_1 \right) \leq \delta$

Output: $\widehat{P}(\theta)$.

4 Asymptotic properties

This section studies the asymptotical properties of the posterior in (5) and the variational posterior in (7). We work with the subset selection prior, which includes the spike and slab prior in (2) as a special case, which is constructed as follows: First, a number s is chosen from a prior π on the set $\{0, \dots, p\}$. Next, a set S is chosen uniformly from the set $\{1, \dots, p\}$ such that its cardinality $|S| = s$. Last, conditional on S , if $j \in S$, then the prior for θ_j is chosen to be $\int_{A \in V_{r,r}} g(\theta_j | \lambda_1, A) d\Pi(A)$; if $j \notin S$, then θ_j is set to $0'_r$. The prior is given as follows:

$$\pi(\theta, S | \lambda_1) \propto \pi(|S|) \frac{1}{\binom{p}{|S|}} \prod_{j \in S} \int_{A \in V_{r,r}} g(\theta_j | \lambda_1, A) \pi(A) dA \prod_{j \notin S} \delta_0(\theta_j). \tag{19}$$

Note that (2) is a special case of (19) when $\pi(|S|)$ is the beta-binomial distribution. That is, $s | \kappa \sim \text{binomial}(p, \kappa)$ and $\kappa \sim \text{Beta}(\alpha_1, \alpha_2)$.

In the next subsection, we will study the theoretical properties of the posterior with the subset selection prior. Before we proceed, some notations need to be introduced. Let \lesssim (resp. \gtrsim) stand for inequalities up (resp. down) to a constant, $a \asymp b$ stand for $C_1 a \leq b \leq C_2 a$ with positive constants $C_1 < C_2$, and $a \ll b$ stand for $a/b \rightarrow 0$. We denote $\|b\|_2$ as the ℓ_2 -norm of a vector b and $\|B\|$ as the spectrum norm of a matrix B . The true value of an unknown parameter ϑ is denoted by ϑ^* .

4.1 Contraction rate of the posterior

We study the dimensionality and the contraction rate of the posterior distribution. In this study, we assume r is unknown and σ^2 is fixed. Three assumptions are needed to obtain the rate.

Assumption 1 (Priors for s and r) For positive constants a_1, a_2, a_3 , and a_4 , assume

$$\begin{aligned} p^{-a_1} & \lesssim \pi(s) / \pi(s - 1) \lesssim p^{-a_2} \\ \exp(-a_3 r) & \lesssim \pi(r) \lesssim \exp(-a_4 r). \end{aligned}$$

The above assumption impose conditions on the tails of the priors $\pi(s)$ and $\pi(r)$. The first condition also appears in the study of the sparse linear regression model (e.g. Castillo et al. 2015; Martin et al. 2017; Ning et al. 2020). It assumes that the logarithm of the ratio between $\pi(s + h)$ and $\pi(s)$ is in the same magnitude as $-h \log p$. When h increases, the assigned probability on $s + h$ decays exponentially fast. The beta-binomial prior mentioned above satisfies this condition if one chooses, for example, $\alpha_1 = 1$ and $\alpha_2 = p^\nu + 1$ for any $\nu > \log \log p / \log p$. The second condition is similar to that in Pati et al. (2014). It assumes the tail of $\pi(r)$ should

decay exponentially fast; the Poisson distribution satisfies this condition.

Assumption 2 (*Bounds for λ_1*) For positive constants b_1, b_2 , and b_3 , assume

$$b_1 \sqrt{\frac{n}{p^{b_2/r^*}}} \leq \lambda_1 \leq b_3 \sqrt{n \log p}.$$

Assumption 2 provides the permissible region for λ_1 . If λ_1 is too large, it introduces an extra shrinkage effect on large signals; if it is too small, the posterior will contract at a slower rate. Our upper bound is of the same order as that in Castillo et al. (2015), where they studied the sparse linear regression model. But the lower bounds are different. Ours is bigger; it can go to 0 very slowly if r^* is close to $\log p / \log n$.

Assumption 3 (*Bounds for r^* and θ^**) For some positive constant b_2, b_4 , and b_5 , $r^* \leq b_2 \log p / \log n$, $\|\theta^*\| \geq b_4$ and $\|\theta^*\|_{1,1} \leq b_5 s^* \log p / \lambda_1$ if $m = 1$ and $1 \leq q \leq 2$ and $\|\theta^*\|^2 \leq b_5 s^* \log p / \lambda_1$ if $m = 2$ and $q = 2$.

Assumption 3 requires the true values of θ and r being bounded. r^* cannot be too large. If $\log p / \log n \lesssim r^* \lesssim \log p$, then the rate obtained in Theorem 4.1 will be slower, i.e., $O(\sqrt{r^* s^* \log p / n})$. The bounds for $\|\theta^*\|$ essentially control the largest eigenvalue, as $\|\theta^*\|^2 + \sigma^2$ is the largest eigenvalue of Σ^* . It cannot be either too big or too small.

We now present the main theorem.

Theorem 4.1 *For the model in (1) and the subset selection prior in (19), if Assumptions 1–3 hold, then for sufficiently large constants M_1, M_2 , and $M_3 \geq M_2/b_4$, as n goes to infinity,*

$$\mathbb{E}_{f^*} \Pi(\theta : |S| > M_1 s^* | X) \rightarrow 0, \tag{20}$$

$$\mathbb{E}_{f^*} \Pi(\|\Sigma - \Sigma^*\| \geq M_2 \epsilon_n | X) \rightarrow 0, \tag{21}$$

$$\mathbb{E}_{f^*} \Pi(\|UU' - U^*U^{*'}\| \geq M_3 \epsilon_n | X) \rightarrow 0, \tag{22}$$

where $\epsilon_n = \sqrt{s^* \log p / n}$.

In Theorem 4.1, we derive the posterior contraction rate under the spectrum loss. The minimax rates for using the spectrum loss have been studied by Cai et al. (2015). Consider the parameter space

$$\Theta_0(s, p, r, \bar{\rho}) = \left\{ \Sigma : 0 \leq \|\theta_{\cdot r}\|_2^2 \leq \|\theta_{\cdot 1}\|_2^2 \leq \bar{\rho}, U \in V_{r,r}, |S| \leq s \right\},$$

the minimax rate of estimating Σ for $r \leq s \leq p$ is $\sqrt{\frac{(\bar{\rho}+1)s}{n} \log\left(\frac{ep}{s}\right) + \frac{\bar{\rho}^2 r}{n}} \wedge \bar{\rho}$. Comparing it to the rate we obtained, assuming $\bar{\rho}$ is fixed and $s \geq r$, our rate is sub-optimal as the log factor in our rate is $\log p$ but in the

minimax rate, it is $\log(p/s)$. Cai et al. (2015) also provided the minimax rate for the projection matrix. Assuming a more restrictive parameter space $\Theta_1(s, p, r, \bar{\rho}, \tau)$,

$$\Theta_1(s, p, r, \bar{\rho}, \tau) = \left\{ \Sigma : \bar{\rho}/\tau \leq \|\theta_{\cdot r}\|_2^2 \leq \|\theta_{\cdot 1}\|_2^2 \leq \bar{\rho}, U \in V_{r,r}, |S| \leq s \right\},$$

the minimax rate is $\sqrt{\frac{(\bar{\rho}+1)s}{n\bar{\rho}^2} \log\left(\frac{ep}{s}\right)} \wedge 1$. Again, if $\bar{\rho}$ is fixed, the rate we obtained is suboptimal.

One may ask if we could obtain the same rate as that in Theorem 4.1 if we use the Frobenius norm as the loss function (in short, Frobenius loss). This is in fact possible, and the proof can simply follow the argument in Gao and Zhou (2015). However, one needs to impose a lower bound for $\|\theta \cdot r\|_2^2$. Although in practice, the lower bound can be introduced through the prior, e.g., using a truncated prior, the exact value is hard to determine. Thus, we did not choose this prior.

4.2 Contraction rate of the variational posterior

We study the contraction rate of the variational posterior in (7). Recent studies on this topic have provided exciting results of the variational method and developed useful tools for studying their theoretical properties (e.g. Ray and Szabo 2022; Wang and Blei 2019; Yang et al. 2020; Zhang and Gao 2020). Ray and Szabo (2022) and Yang et al. (2020) studied the spike and slab posterior with the linear regression model and obtained a (near-)optimal rate for their posterior. Zhang and Gao (2020) proposed a general framework for deriving the contraction rate of a variational posterior. We derive the rate by directly applying this general framework, as our variational posterior is intractable, and using a direct argument (e.g., those in the linear regression model) is impossible. Theorem 4.2 shows that the rate of the variational posterior is also ϵ_n (but with a larger constant). Proofs of the theorem are provided in the supplemental material.

Theorem 4.2 *With the model (1) and the subset selection prior (19), if $\widehat{P}(\theta) \in \mathcal{P}^{MF}$ and Assumptions 1–3 hold, then for large constants M_4 and M_5 , as n goes to infinity,*

$$\widehat{Q}(\|\Sigma - \Sigma^*\| \geq M_4 \epsilon_n | X) \rightarrow 0, \tag{23}$$

$$\widehat{Q}(\|UU' - U^*U^{*'}\| \geq M_5 \epsilon_n | X) \rightarrow 0. \tag{24}$$

5 The PX-EM algorithm

The EM algorithm is another popular algorithm that is used in Bayesian high-dimensional analysis. In this section, to understand the strength of PX-CAVI, we also develop its EM

analog, referred to as the PX-EM algorithm. The parameter expansion steps for the PX-EM algorithm mirror those used in the PX-CAVI algorithm. The PX-EM algorithm requires us to use the continuous spike and slab prior, which is

$$\pi(\theta, S|\lambda_1, \lambda_0) \propto \int_A \prod_{j=1}^p [\gamma_j g(\theta_j|\lambda_1, A, r) + (1 - \gamma_j)g(\theta_j|\lambda_0, A, r)] \pi(A) dA, \tag{25}$$

where $\lambda_0 \gg \lambda_1$. By comparing to (2), the Dirac measure is replaced by the continuous density with a large variance. The priors for the rest parameters remain the same.

Our PX-EM algorithm contains two steps: E-step and M-step. In the E-step, expectations are taken with respect to both w and γ . We then obtain

$$w_i|\theta^{(t)}, X \sim \mathcal{N}(\omega_i, V_w), \tag{26}$$

$$\gamma_j \sim \text{Bernoulli}(\tilde{\gamma}_j), \tag{27}$$

where $\theta^{(t)}$ and $\kappa^{(t)}$ are the estimated values of θ and κ from the t -th iteration and

$$V_w = \sigma^2(\theta^{(t)'}\theta^{(t)} + \sigma^2 I_r)^{-1}, \quad \omega_i = \sigma^{-2} V_w \theta^{(t)'} X_i, \tag{28}$$

$$\tilde{\gamma}_j^{(t)} = P(\gamma_j = 1|\theta^{(t)}, \kappa^{(t)}, X) = \frac{a_j^{(t)}}{a_j^{(t)} + b_j^{(t)}}, \tag{29}$$

where $a_j^{(t)} = \exp(-\lambda_1 \|\theta_j^{(t)}\|_q^m + \log \kappa^{(t)})$ and $b_j^{(t)} = \exp(-\lambda_0 \|\theta_j^{(t)}\|_q^m + \log(1 - \kappa^{(t)}))$.

To obtain the objective function, we first apply parameter expansion to the likelihood, same as that in the PX-CAVI algorithm. The expanded parameter becomes $\tilde{\beta} = \beta D = \theta AD$. The spike and slab prior is then directly applied on $\tilde{\beta}$. The objective function is given by $Q(\tilde{\beta}, \kappa|\theta^{(t)}, A^{(t)}, D^{(t)}, \kappa^{(t)})$, where

$$\begin{aligned} Q &= \mathbb{E}_{w, \gamma|\theta^{(t)}, \kappa^{(t)}} \log \pi(\tilde{\beta}, w|X) \\ &= C - \sum_{j=1}^p \left(\frac{1}{2\sigma^2} \left\| M_L \tilde{\beta}'_j - d_j \right\|_2^2 \right. \\ &\quad \left. + (\tilde{\gamma}_j \lambda_1 + (1 - \tilde{\gamma}_j) \lambda_0) \|\tilde{\beta}_j\|_q^m \right) \\ &\quad + (\|\tilde{\gamma}\|_1 + \alpha_1 - 1) \log \kappa + (p - \|\tilde{\gamma}\|_1 + \alpha_2 - 1) \log(1 - \kappa), \end{aligned} \tag{30}$$

where C is a constant, M_L is the lower triangular part from the Cholesky decomposition, $M = \sum_{i=1}^n \omega_i \omega_i' + nV_w$, and $d_j = M_L^{-1} \sum_{i=1}^n \omega_i X_{ij}$.

In the M-step, we maximize the objective function and obtain

$$\hat{\tilde{\beta}}_j = \arg \min_{\tilde{\beta}_j} \left\{ \frac{1}{2\sigma^2} \left\| M_L \tilde{\beta}'_j - d_j \right\|_2^2 + \text{pen}_j \|\tilde{\beta}_j\|_q^m \right\}, \tag{31}$$

$$\hat{\kappa} = \frac{\alpha_1 + \|\tilde{\gamma}\|_1 - 1}{p + \alpha_1 + \alpha_2 - 2}, \tag{32}$$

where $\text{pen}_j = \tilde{\gamma}_j \lambda_1 + (1 - \tilde{\gamma}_j) \lambda_0$. Then $\hat{\theta}$ is obtained using $\tilde{\beta} = \theta AD_L$, where $\hat{D} = \frac{1}{n} \sum_{i=1}^n \omega_i \omega_i' + V_w$ and \hat{A} is obtained by applying the SVD on the matrix $\hat{\tilde{\beta}} \hat{D}_L^{-1}$.

In (31), we choose $m = 1$ and let $q = 1$ and 2. When $q = 1$, the expression is similar to that of the adaptive lasso (Zou et al. 2006). When $q = 2$, the penalty term is then similar to it in the group lasso method (Yuan and Lin 2006). Despite those similarities, the tuning parameter in (31) can be updated during each EM iteration; however, in both of the two aforementioned literature, their tuning parameters are chosen to be fixed values. The benefit of allowing the tuning parameter to update is explored by Ročková (2018), which studied the sparse normal mean model.

Last, we obtain

$$\hat{\sigma}^2 = \frac{\text{Tr}(X'X) - 2 \sum_{j=1}^p d_j M_L \theta'_j + \sum_{j=1}^p \theta_j M \theta'_j + 2\sigma_b}{np + 2(\sigma_a + 1)}. \tag{33}$$

Algorithm 2: The PX-EM algorithm

Data: X , a $p \times n$ matrix, centered and scaled

Input: $\theta^{(0)}, \sigma^{(0)}, r$, number of total iterations T , and the threshold δ

For $t = 0, \dots, T - 1$, **repeat:**

- Update $\omega^{(t+1)}$ and $V_w^{(t+1)}$ from (26) and $\tilde{\gamma}^{(t+1)}$ from (27);
- Update $\tilde{\beta}_j^{(t+1)}$ from (31)
- Update $\kappa^{(t+1)}$ from (32)
- Update $D^{(t+1)}$ and $A^{(t+1)}$ and then obtain $\theta^{(t+1)}$ and $U^{(t+1)}$
- Update $\sigma^{(t+1)}$ from (33)
- Evaluate the objective function $Q^{(t+1)}$ in (30)

Stop: if $|\log Q^{(t+1)} - \log Q^{(t)}| \leq \delta$

Output: $\hat{\theta} = \theta^{(t+1)}, U = \hat{U}^{(t+1)}, \hat{\gamma} = \tilde{\gamma}^{(t+1)}$, and $\hat{\sigma} = \sigma^{(t+1)}$.

We conclude this section by offering theoretical justification for utilizing parameter expansion to accelerate the convergence speed of the EM algorithm. We observed that the convergence speed improves with both parameter expansions. Intuitively, by Dempster et al. (1977), the speed of convergence is determined by the largest eigenvalue of

$S(\Delta) = I_{com}^{-1}(\Delta)I_{obs}(\Delta)$, where

$$I_{obs}(\Delta) = - \left. \frac{\partial^2 \log(\Delta|X)}{\partial \Delta \partial \Delta'} \right|_{\Delta=\Delta^*} \quad \text{and}$$

$$I_{com}(\Delta) = - \left. \frac{\partial^2 Q(\Delta|\Delta)}{\partial \Delta \partial \Delta'} \right|_{\Delta=\Delta^*}. \tag{34}$$

We denote Δ as the collection of all the unknown parameters and Δ^* as the true values. Let Ψ be the expanded parameter and $\tilde{\Delta} = (\Delta, \Psi)$. We found that the largest eigenvalue of $S(\tilde{\Delta})$ is bigger than that of $S(\Delta)$. Thus, the convergence speed is increased. In Lemma 5.1, we provide a formal statement of this result. Proof of Lemma 5.1 is provided in the supplementary material.

Lemma 5.1 *Given that the PX-EM algorithm converges to the posterior mode, both parameter expansions speed up the convergence of the original EM algorithm.*

6 Simulation study

In this section, we conduct four simulation studies to evaluate the performance of our proposed PX-CAVI algorithm. Firstly, we compare the use of a product of Laplace density (i.e., $q = 1$ and $m = 1$ in g (3)) with the multivariate normal density (i.e., $q = 2$ and $m = 2$ in g (3)) within the PX-CAVI algorithm. Next, we compare the PX-CAVI algorithm with the PX-EM algorithm. Additionally, we introduce the batch PX-CAVI algorithm, which does not require θ to be jointly row-sparse, and compare it with two other penalty methods for SPCA and the conventional PCA. In the final study, we assume that r is unknown and demonstrate that the algorithm is less sensitive to the choice of r . Throughout all the studies, we set σ^2 to be fixed. However, in the R package we provided, it has the capability to estimate σ^2 automatically.

The dataset is generated as follows: First, given r^* , s^* , and p , we generate U^* using the `randortho` function in R. Next, we set $\sigma^2 = 0.1$ and choose the diagonal values of Λ^* to be an equally spaced sequence from 10 to 20 (i.e., the largest value is 20 and the smallest value is 10); however, in the first study, we will choose different values for Λ^* ; see Sect. 6.2 for details. Last, we obtain $\Sigma^* = U^* \Lambda^* U^{*'} + \sigma^2 I_p$ and generate $n = 200$ independent samples from $\mathcal{N}(0, \Sigma^*)$. Then, the dataset is an $n \times p$ matrix. For each simulated dataset, we obtain the following quantities: the Frobenius loss of the projection matrix $\|\widehat{U}\widehat{U}' - U^*U^{*'}\|_F$, the percentage of misclassification also known as the average Hamming distance $\|\widehat{z} - \mathbf{y}^*\|_1/p$, the false discovery rate (FDR), and the false negative rate (FNR). The hyperparameters in the prior are chosen as follows: $\lambda_1 = 1, \alpha_1 = 1, \alpha_2 = p + 1, \sigma_a = 1$, and $\sigma_b = 2$. Also, we set the total iterations $T = 100, \iota = 0.1$,

and the threshold $\delta = 10^{-4}$. To determine whether $\gamma_j = 1$ or 0, we choose the threshold to be 0.5.

6.1 On choosing the initial values for PX-CAVI and PX-EM

Before presenting the simulation results, let us discuss how we obtained the initial values for the PX-CAVI algorithm, as well as the batch PX-CAVI algorithm, and the initial values for the PX-EM algorithm. We carefully explored different choices of initial values and found that the PX-CAVI algorithm exhibits robustness against variations in the initial values. Consequently, the algorithm is not overly sensitive to the specific choices of initializations. Therefore, we estimated $\widehat{\mu}^{(0)}$ using the conventional PCA and set $\widehat{z}^{(0)} = \mathbf{1}'_p$. For $\widehat{M}_j^{(0)}$, we let it be an identity matrix times a small value (i.e., 10^{-3}). Finally, for $(\widehat{\sigma}^{(0)})^2$, we chose it to be the smallest eigenvalue of the Gramian matrix $X'X/(n - 1)$.

The PX-EM algorithm is more sensitive to poor initializations than the PX-CAVI algorithm. To address this concern, we employed two strategies aimed at alleviating this issue. The first one is through prior elicitation, which is proposed by Ročková and Lesaffre (2014). We replaced (27) with its tempered version given by

$$\widehat{\gamma}_j^{(\iota)} = P(\gamma_j = 1 | \theta^{(\iota)}, X) = \frac{(a_j^{(\iota)})^\iota}{(a_j^{(\iota)})^\iota + (b_j^{(\iota)})^\iota}. \tag{35}$$

where $\iota < 1$ is fixed. In the simulation study, we fix $\iota = 0.1$.

Another strategy is to choose the “best-guess” of initial values using the path-following strategy proposed by Ročková and George (2016). First, we chose a vector containing a sequence of values of $\lambda_0, \{\lambda_0^{(1)}, \dots, \lambda_0^{(I)}\}$, where $\lambda_0^{(1)} = \lambda_1 + 2\sqrt{\rho_{\min}}$ with ρ_{\min} being the smallest eigenvalue of $X'X/(n - 1)$, and $\lambda_0^{(I)} = p^2 \log p$. Next, we obtained an initial value of θ using the conventional PCA and repeated the following process: At i -th step, set $\lambda_0 = \lambda_0^{(i)}$ and chose the input values as their output values obtained from the $(i - 1)$ -th step. We repeated this I times until all the values in that sequence of λ_0 are used. Finally, the values output from the last step are used as the initial values for the PX-EM algorithm. As can be seen, comparing to the PX-CAVI algorithm, obtaining the initial values of the PX-EM algorithm takes a much longer time.

6.2 Laplace density vs normal density

Let $r^* = 1$, then g is the Laplace distribution ($m = 1, q = 1$) and the normal distribution ($m = 2, q = 2$). We conducted simulation studies of the PX-CAVI algorithm and compared the use of two distributions. We chose $\|\theta^*\|^2 \in$

{1, 3, 5, 10, 20} and $p \in \{100, 1000\}$. For each setting, 1000 datasets are generated. Simulation results are provided in Table 1.

From Table 1, we observed the following results: For $p = 100$, there is no significant difference between using the normal and the Laplace densities, as their results are similar. However, when $p = 1000$, using the normal density yields better results, as indicated by the smaller average value of the Frobenius loss of the projection matrix. In the case of $p = 1000$, the normal density outperforms the Laplace density in estimating weaker signals (e.g., observed in the Frobenius loss when $|\theta^*| = 1$). The computational speed using the normal density is faster than the Laplace density. This is because when choosing the Laplace density, the algorithm needs to solve the two nonlinear functions (12) and (13) in each iteration. The computational speed notably increases, particularly when $r > 2$ using the Laplace density, and solving the two Eqs. (12) and (13) becomes more challenging. Based on these findings, we recommend using the multivariate normal density, especially when the rank r is large, as it provides improved performance and computational efficiency in comparison to the Laplace density.

6.3 Comparison between PX-CAVI and PX-EM

In this study, we compare the PX-CAVI algorithm with the PX-EM algorithm. Two options for q in (31) are considered for the PX-EM algorithm: $q = 1$ representing the ℓ_1 -norm, and $q = 2$ representing the ℓ_2 -norm. We observed that the algorithm using the ℓ_1 -norm outperforms the one using the ℓ_2 -norm in terms of parameter estimation (see the simulation result in the Supplementary Material). Henceforth, we utilized the ℓ_1 -norm. The true parameter values were chosen as follows: We fixed $s^* = 20$, and $r^* = 2$ and chose $q = 1$, $s^* \in \{10, 40, 70, 150\}$, $r^* \in \{1, 3, 5\}$, and $p \in \{500, 1000, 2000, 4000\}$. We ran both the PX-CAVI and the PX-EM algorithms. The results are given in Table 2. As we mentioned in Sect. 6.1, choosing the initial values for the PX-EM algorithm takes a longer time, and thus, we were only able to run 100 simulations. For the PX-CAVI, the result is based on 1000 simulations.

We remark two findings in Table 2. First, in general, the PX-CAVI algorithm is better than the PX-EM algorithm in both parameter estimation and variable selection. When s^* and r^* are large, the PX-CAVI algorithm is more accurate.

Although it seems that when s^* and r^* are small (e.g., $s^* = 10$ and $r^* = 1$ and $s^* = 40$ and $r^* = 1$), the Frobenius loss and the percentage of misclassification are bigger in the PX-CAVI algorithm than the PX-EM algorithm. However, the standard errors associate with the Frobenius loss when $s^* = 10$ and $r^* = 1$ is 0.011 and $s^* = 40$ and $r^* = 1$ is 0.015. For the percentage of misclassification, the standard errors are 0.1 when $s^* = 10$ and $r^* = 1$ and 0.2 when $s^* = 40$ and $r^* = 1$. Consequently, the observed differences between the two algorithms are insignificant. Our second notable finding is that both algorithms effectively control the FDR. However, the PX-CAVI algorithm exhibits better control over the FNR, resulting in more accurate and desirable variable selection outcomes.

6.4 The batch PX-CAVI vs other SPCA algorithms

The PX-CAVI algorithm assumes θ to be jointly row-sparse. In the Supplementary Material, we propose the batch PX-CAVI algorithm, which relaxes this assumption, allowing each principal component to have identical support. The batch PX-CAVI algorithm updates the coordinates belonging to the same row simultaneously. To evaluate the performance of the batch PX-CAVI algorithm, we compare it with two other popular algorithms for SPCA: the elastic net method proposed by Zou et al. (2006) and the robust SPCA method proposed by Erichson et al. (2020). Both of these methods are penalty-based approaches, and their tuning parameters are fixed (unlike the PX-EM algorithm). They are often used in practice, and their R packages `elasticnet` and `sparsePCA` are available on CRAN.

To determine the optimal values of the tuning parameters for each algorithm, we consider a vector containing 100 values and estimate the Frobenius loss of the projection matrix for each value in ascending order. The tuning parameter that results in the smallest Frobenius loss value is selected as the optimal value. The results are presented in Table 3. Notably, we observed that the batch PX-CAVI algorithm outperforms the other three algorithms listed in the table with the small-

Table 1 Simulation results of the PX-CAVI algorithm using the Laplace and Normal densities

	$\ \theta^*\ ^2$	Prior	$p = 100$				$p = 1000$			
			Frob	Misc(%)	FDR	FNR	Frob	Misc(%)	FDR	FNR
1		Normal	0.156	2.4	0.000	0.026	0.174	0.3	0.000	0.003
		Laplace	0.156	2.4	0.000	0.026	0.190	0.3	0.000	0.003
3		Normal	0.076	1.4	0.000	0.015	0.082	0.1	0.000	0.001
		Laplace	0.076	1.4	0.000	0.015	0.088	0.2	0.000	0.002
5		Normal	0.055	1.0	0.000	0.012	0.059	0.1	0.000	0.001
		Laplace	0.055	1.0	0.000	0.012	0.062	0.1	0.000	0.001
10		Normal	0.036	0.8	0.000	0.008	0.038	0.1	0.000	0.001
		Laplace	0.036	0.8	0.000	0.008	0.039	0.1	0.000	0.001
20		Normal	0.024	0.5	0.000	0.006	0.026	0.1	0.000	0.001
		Laplace	0.024	0.5	0.000	0.006	0.026	0.1	0.000	0.001

We fixed $n = 200$, $s^* = 20$ and $r^* = 1$ and chose $p \in \{100, 1000\}$ and $\|\theta^*\|^2 \in \{1, 3, 5, 10, 20\}$. For each setting, we ran 1000 simulations and computed the average values of the Frobenius loss of the projection matrix, the percentage of misclassification, FDR, and FNR

Table 2 Simulation results of the PX-CAVI and the PX-EM algorithms. We fixed $n = 200$ and chose $s^* \in \{10, 20, 40, 70, 150\}$, $r^* \in \{1, 2, 3, 5\}$, and $p \in \{500, 1000, 2000, 4000\}$

p	s^*	r^*	Frobenius loss		Misc (%)		FDR		FNR	
			PX-CAVI	PX-EM	PX-CAVI	PX-EM	PX-CAVI	PX-EM	PX-CAVI	PX-EM
1000	10	1	0.025	0.024	0.1	0.1	0.000	0.001	0.001	0.001
1000	10	3	0.039	0.040	0.1	0.0	0.000	0.000	0.000	0.000
1000	10	5	0.043	0.043	0.1	0.0	0.000	0.000	0.000	0.000
1000	40	1	0.061	0.054	0.5	0.4	0.000	0.001	0.006	0.004
1000	40	3	0.092	0.128	0.0	0.1	0.000	0.000	0.000	0.001
1000	40	5	0.113	0.128	0.0	0.1	0.000	0.000	0.000	0.001
1000	70	1	0.089	0.093	1.2	1.2	0.000	0.000	0.016	0.013
1000	70	3	0.126	0.214	0.1	0.4	0.000	0.000	0.001	0.005
1000	70	5	0.155	0.212	0.0	0.1	0.000	0.000	0.000	0.002
1000	150	1	0.145	0.155	3.5	3.7	0.000	0.000	0.052	0.042
1000	150	3	0.194	0.463	0.3	2.6	0.000	0.000	0.010	0.029
1000	150	5	0.231	0.520	0.0	1.5	0.000	0.000	0.002	0.017
500	20	2	0.054	0.067	0.0	0.1	0.000	0.000	0.001	0.001
1000	20	2	0.054	0.072	0.0	0.1	0.000	0.000	0.000	0.001
2000	20	2	0.055	0.063	0.0	0.0	0.000	0.000	0.000	0.001
4000	20	2	0.055	0.057	0.0	0.0	0.000	0.000	0.000	0.001

For each setting, we ran 100 simulations for the PX-EM and 1000 simulations for the PX-CAVI. We computed the average values of the Frobenius loss of the projection matrix, the percentage of misclassification, FDR, and FNR

est estimation and selection errors, regardless of the values of p , s^* , and r^* . Furthermore, the algorithm proposed by Zou et al. (2006) shows better performance than Erichson et al. (2020)'s method when r^* is large. As expected, all three algorithms (batch PX-CAVI and two penalty methods) outperform the conventional PCA method. The program was executed on a MacBook Pro laptop with a 2.9 GHz Intel Core i7 processor and 16 GB memory. With specific parameters set at $n = 200$, $p = 1000$, $s = 10$, and $r = 2$, the average running time for a single simulation were found to be 14.96 s

for the PX-CAVI algorithm, 0.281 s for the method proposed by Zou et al. (2006), and 0.498 s for the method introduced by Erichson et al. (2020). The running time of our PX-CAVI algorithm can be further accelerated by using parallel computing to update \hat{u}_j in (12) and (14) for each $j = 1, \dots, p$ in Algorithm 1.

Table 3 Simulation results of the batch PX-CAVI (bPX-CAVI) algorithm, the two SPCA algorithms proposed by Zou et al. (2006) and Erichson et al. (2020) (namely, sPCA1 and sPCA2), and PCA

p	s^*	r^*	Frobenius loss				Misclassification (%)		
			bPX-CAVI	sPCA1	sPCA2	PCA	bPX-CAVI	sPCA1	sPCA2
1000	10	1	0.025	0.073	0.066	0.222	0.1	0.1	0.1
1000	10	3	0.046	0.169	0.359	0.461	0.1	0.3	10.5
1000	10	5	0.052	0.207	0.607	0.593	0.1	0.6	26.5
1000	40	1	0.061	0.115	0.114	0.222	0.5	0.8	0.5
1000	40	3	0.131	0.261	0.348	0.462	0.6	2.7	8.3
1000	40	5	0.164	0.342	0.565	0.593	0.6	2.4	26.7
1000	70	1	0.089	0.145	0.151	0.222	1.2	1.8	1.2
1000	70	3	0.193	0.316	0.368	0.462	1.3	3.4	9.1
1000	70	5	0.247	0.417	0.565	0.593	1.3	14.5	27.3
1000	150	1	0.145	0.189	0.203	0.223	3.5	4.2	3.5
1000	150	3	0.321	0.405	0.422	0.463	4.1	8.4	12.3
1000	150	5	0.405	0.526	0.594	0.593	4.1	16.3	28.7
500	20	2	0.068	0.183	0.257	0.273	1.3	2.3	1.2
1000	20	2	0.068	0.175	0.268	0.383	0.1	1.8	0.1
2000	20	2	0.070	0.215	0.285	0.532	0.1	0.6	0.2
4000	20	2	0.072	0.332	0.304	0.725	0.1	0.1	0.0

We fixed $n = 200$ and chose $s^* \in \{10, 20, 40, 70, 150\}$, $r^* \in \{1, 2, 3, 5\}$, and $p \in \{500, 1000, 2000, 4000\}$. For each setting, we ran 1000 simulations and obtained the average values of the Frobenius loss of the projection matrix and the percentage of misclassification

Table 4 Simulations for the PX-CAVI algorithm choosing different input values for r

	$ \langle \widehat{U}_{\cdot 1}, U_{\cdot 1}^* \rangle $	$ \langle \widehat{U}_{\cdot 2}, U_{\cdot 2}^* \rangle $	$ \langle \widehat{U}_{\cdot 3}, U_{\cdot 3}^* \rangle $	$ \langle \widehat{U}_{\cdot 4}, U_{\cdot 4}^* \rangle $	Misc (%)
$r = 1$	0.868 (0.180)				1.1 (0.3)
$r = 2$	0.864 (0.188)	0.798 (0.217)			0.2 (0.1)
$r = 3$	0.866 (0.186)	0.801 (0.216)	0.855 (0.170)		0.0 (0.1)
$r = 4$	0.869 (0.183)	0.803 (0.214)	0.855 (0.171)	0.932 (0.104)	0.0 (0.0)
$r = 5$	0.868 (0.183)	0.803 (0.214)	0.855 (0.170)	0.933 (0.104)	0.0 (0.0)
$r = 20$	0.881 (0.173)	0.811 (0.213)	0.850 (0.176)	0.933 (0.099)	0.0 (0.0)

Let $r^* = 4$, $n = 200$, $p = 1000$, and $s^* = 70$, and we generated 1000 datasets. For each value $r \in \{1, 2, 3, 4, 5, 20\}$, we calculated the average values (and the standard errors) of the quantity, $|\langle \widehat{U}_{\cdot k}, U_{\cdot k}^* \rangle|$, and the percentage of misclassification

Table 5 The results of the top 10 references IDs of genes and the total numbers of active genes of the first two principal components estimated by the PX-CAVI and the batch PX-CAVI (bPX-CAVI) algorithms and PCA

Ranking	1st principal component			2nd principal component		
	PX-CAVI	bPX-CAVI	PCA	PX-CAVI	bPX-CAVI	PCA
1	38691_s_at	38691_s_at	38691_s_at	41209_at	41209_at	39220_at
2	37004_at	37004_at	37004_at	39,220_at	39220_at	41,209_at
3	33383_f_at	33383_f_at	33383_f_at	38430_at	38430_at	38430_at
4	35926_s_at	35926_s_at	35926_s_at	34708_at	34708_at	34708_at
5	37864_s_at	37864_s_at	37864_s_at	33377_at	40607_at	40607_at
6	41723_s_at	41723_s_at	41723_s_at	40607_at	33377_at	33377_at
7	38096_f_at	38096_f_at	38096_f_at	36780_at	36119_at	36119_at
8	38194_s_at	38194_s_at	38194_s_at	36119_at	36780_at	36780_at
9	33274_f_at	33274_f_at	33274_f_at	32452_at	32452_at	32452_at
10	33500_i_at	33500_i_at	33500_i_at	32052_at	35730_at	35730_at
# of nonzeros	1183	1469	5000	1183	795	5000

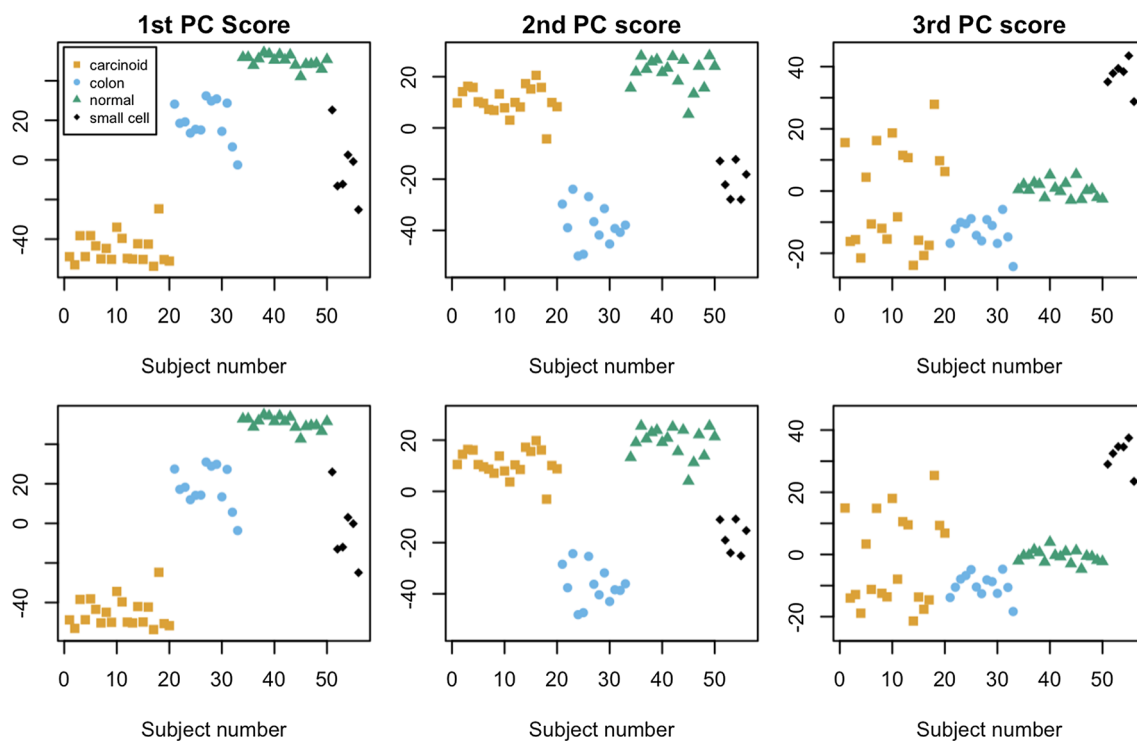


Fig. 1 The three plots in the first row are the first three principal component scores estimated using the PX-CAVI algorithm and the three plots at the bottom are the same score functions estimated using the batch PX-CAVI algorithm

6.5 Unknown r

The previous three studies assumed that r is known. In this study, we investigate the scenario where r is unknown. Rather than modifying our algorithms to estimate r directly—which could increase computation time and introduce complexity in choosing initial values—we propose a practical approach of plugging in a value for r before conducting the analysis. This plugged-in value can be obtained through other algorithms or based on prior studies. Importantly, the accuracy of the plugged-in value is not critical. This study is designed as follows: We set $r^* = 4$, $n = 200$, $p = 1000$, and $s^* = 70$. The input value of r is chosen to be $r = 1, 2, 3, 4, 5, 20$. For each value, we ran the PX-CAVI algorithm and obtained the average values of $|\langle \widehat{U}_k, U_k^* \rangle|$ and the percentage of misclassification from 1000 simulations. Note that \widehat{U}_k is the k -th eigenvector from $\widehat{\mu}$; \widehat{U}_k and U_k^* are close if $|\langle \widehat{U}_k, U_k^* \rangle|$ is close 1. The results are provided in Table 4. From that table, we found that regardless of the input value r , even when $r = 20$, the results are similar. Additionally, we noticed that as the rank increases, the accuracy of variable selection improves.

7 A real data study

This section applies the PX-CAVI and batch PX-CAVI algorithms to analyze a lung cancer dataset. This a gene expression dataset, accessible through the R package `sparseBC`, which comprises expression levels of 5000 genes and 56 subjects. These subjects encompass 20 pulmonary carcinoid subjects (carcinoid), 13 colon cancer metastasis subjects (colon), 17 normal lung subjects (normal), and 6 small cell lung subjects (small cell). The primary objective is to identify biologically relevant genes correlated with lung cancer and distinguish the four different cancer types.

To prepare the data for analysis, we center and scale it before running each algorithm. In this study, we set the rank $r = 8$, as it captures over 70% variability. Furthermore, we are particularly interested in the first three principal components (PCs). Therefore, selecting $r = 8$ serves the purpose well. Table 5 presents the top 10 reference IDs of genes identified from the first and second PCs. Each reference ID corresponds to a specific gene, and this correspondence can be validated using the NCBI website. For instance, the reference ID ‘38,691_s_at’ represents the gene 6440 (see <https://www.ncbi.nlm.nih.gov/geo/profiles/62830018>).

From Table 5, we made the following observations. The top ten genes of the first principal component obtained from all three algorithms are the same. In the second PC, the order might vary slightly, but overall, the results are similar. We

conducted a gene count analysis to determine the number of genes with nonzero loading values for each PC. For PCA, which does not impose sparsity on the loadings matrix, the total number of nonzeros is equal to the total number of genes. The PX-CAVI algorithm ensures that all PCs have the same number of nonzero loadings by the jointly row-sparsity assumption. This property leads to easier interpretation, as there is no concern about specific genes being selected in the first PC but not in second PC. The batch PX-CAVI algorithm employs fewer genes than the PX-CAVI algorithm to construct the second PC. By comparing their score functions in Fig. 1, we observed that using either 1183 genes or 795 genes to represent PCs does not result in significant differences. This demonstrates the advantage of the batch PX-CAVI algorithm in utilizing fewer genes to construct PCs while maintaining comparable performance. Additionally, we provided the first three PC scores estimated by both algorithms and highlighted the four different cancer types using different colors. As shown in the PC scores, the four cancer types are well-separated, indicating the effectiveness of our algorithms in distinguishing between the different types of lung cancer.

8 Conclusion and discussion

In this paper, we proposed the PX-CAVI algorithm (also the batch PX-CAVI algorithm) and its EM analogue the PX-EM algorithm for Bayesian SPCA. These algorithms utilized parameter expansion to effectively handle the orthogonality constraint imposed by the loading matrix and enhance their convergence speeds. We demonstrated that the PX-CAVI algorithm outperforms all other algorithms discussed in the paper, showcasing its superiority. Furthermore, we studied the posterior contraction rate of the variational posterior, providing a novel contribution to the existing literature. Additionally, our findings revealed that choosing the normal (or multivariate normal) density for g yielded better results compared to the heavier-tailed Laplace density.

Future studies include understanding why the Laplace density of the current prior fails to yield smaller estimation errors even in the rank one case and choosing other shrinkage priors such as the product moment prior in Johnson and Rossell (2010) and the non-local priors considered in Avalos-Pacheco et al. (2022). Additionally, the uncertainty quantification problem of SPCA remains unexplored, despite the rich literature on this topic for the sparse linear regression model (see van der Pas et al. 2017; Belitser and Ghosal 2020; Castillo and Roquain 2020; Martin and Ning 2020). Moreover, gaining a deeper understanding of the variational posterior, i.e. its conditions for achieving variable selection consistency would be valuable. Lastly, an interesting avenue for exploration involves extending our proposed method to

the unsupervised learning setting, as explored in She (2017). In this context, the row-wise sparsity and row-rank restrictions can be imposed through priors that is similar to the those considered in the current paper. Our R package VBsparsePCA for the PX-CAVI and batch PX-CAVI algorithms is available on CRAN, offering a practical tool for researchers to apply these algorithms in their analyses.

Supplementary Material

Supplement to “Spike and slab Bayesian sparse principal component analysis”

(). In this supplementary material, we present the batch PX-CAVI algorithm, include the simulation results of the the PX-EM algorithm by choosing ℓ_1 -norm and ℓ_2 -norm in its penalty term, give the proofs of Theorems 4.1 and 4.2 and Lemma 5.1, and provide some auxiliary lemmas.

Appendix A: Derivation of (12)–(18)

First, we need the following result:

$$\begin{aligned} \mathbb{E}_{w|\Theta^{(t)}} \left[\frac{1}{2\sigma^2} \sum_{i=1}^n (X_{ij} - \tilde{\beta}_j w_i)^2 \right] \\ = \frac{1}{2\sigma^2} \sum_{i=1}^n \left(X_{ij}^2 - 2X_{ij} \tilde{\beta}_j \tilde{w}_i + \tilde{\beta}_j H_i \tilde{\beta}_j' \right), \end{aligned} \quad (36)$$

where $H_i = \tilde{w}_i \tilde{w}_i' + \tilde{V}_w$ and the expressions of \tilde{w}_i and \tilde{V}_w are given in (10).

Since the ELBO is a summation of p terms, we solve u_j and M_j for each j . As the posterior conditional on $\gamma_j = 0$ is singular to the Dirac measure, we only need to consider the case $\gamma_j = 1$. This leads to minimize the function

$$\begin{aligned} \mathbb{E}_{\tilde{u}_j, \tilde{M}_j, z_j | \gamma_j = 1} \left[\frac{1}{2\sigma^2} \sum_{i=1}^n \left(-2X_{ij} \tilde{\beta}_j \tilde{w}_i + \tilde{\beta}_j H_i \tilde{\beta}_j' \right) \right. \\ \left. + \log \frac{N(\tilde{u}_j, \sigma^2 \tilde{M}_j)}{\kappa_j^\circ g(\tilde{\beta}_j | \lambda_1)} \right] \\ = C - \frac{1}{\sigma^2} \sum_{i=1}^n X_{ij} \tilde{u}_j \tilde{w}_i \\ + \frac{1}{2\sigma^2} \sum_{i=1}^n \left(\tilde{u}_j H_i \tilde{u}_j + \text{Tr} \left(\sigma^2 \tilde{M}_j H_i \right) \right) \\ + \lambda_1 \sum_{k=1}^r f(\tilde{u}_{jk}, \tilde{M}_{j,kk}), \end{aligned}$$

where $\kappa_j^\circ = \int \pi(\gamma_j | \kappa) d\Pi(\kappa)$. Then we take the derivative of \tilde{u}_j and \tilde{M}_j to obtain (12) and (13). The solutions in (14)

are obtained by changing $\lambda_1 \sum_{k=1}^r f(\tilde{u}_{jk}, \tilde{M}_{j,kk})$ in the last display with $\frac{\lambda_1}{2\sigma^2} (\tilde{u}_j \tilde{u}'_j + \sigma^2 \text{Tr}(M_j))$.

To derive (15), we have

$$\begin{aligned} & \mathbb{E}_P \left(\mathbb{E}_{w|\Theta^{(t)}} \pi(\tilde{\beta}_j, w, X) - \log q(\tilde{\beta}_j) \right) \\ &= C + \mathbb{E}_{\tilde{\mu}_j, \tilde{M}_j, z_j} \left[\frac{1}{2\sigma^2} \sum_{i=1}^n \left(-2X_{ij} \tilde{\beta}_j \tilde{\omega}_i + \tilde{\beta}_j H_i \tilde{\beta}'_j \right) \right. \\ & \quad + \mathbb{1}_{\{y_j=0\}} \log \frac{1-z_j}{1-\kappa_j^\circ} \\ & \quad \left. + \mathbb{1}_{\{y_j=1\}} \log \frac{z_j N(\tilde{\mu}_j, \sigma^2 M_j)}{\kappa_j^\circ g(\tilde{\beta}_j | \lambda_1)} \right] \\ &= C + (1-z_j) \log \frac{1-z_j}{1-\kappa_j^\circ} \\ & \quad + z_j \left\{ \frac{1}{2\sigma^2} \sum_{i=1}^n \left(\tilde{\mu}_j H_i \tilde{\mu}'_j + \sigma^2 \text{Tr}(\tilde{M}_j H_i) - 2X_{ij} \tilde{\mu}_j \tilde{\omega}_i \right) \right. \\ & \quad + r \log \left(\frac{\sqrt{2}}{\sqrt{\pi} \sigma \lambda_1} \right) - \frac{1}{2} \log \det(\tilde{M}_j) - \frac{1}{2} \\ & \quad \left. + \lambda_1 \sum_{k=1}^r f(\tilde{\mu}_{jk}, \sigma^2 \tilde{M}_{j,kk}) + \log \frac{z_j}{\kappa_j^\circ} \right\}. \end{aligned} \tag{37}$$

The solution of \hat{h}_j can be obtained by minimizing z_j from the last line of the above display. Similarly, (16) is obtained by minimizing z_j from the following expression

$$\begin{aligned} & C + z_j \left\{ \frac{1}{2\sigma^2} \sum_{i=1}^n \left(\tilde{\mu}_j H_i \tilde{\mu}'_j + \sigma^2 \text{Tr}(\tilde{M}_j H_i) - 2X_{ij} \tilde{\mu}_j \tilde{\omega}_i \right) \right. \\ & \quad - \frac{r \log \lambda_1 + 1}{2} - \frac{1}{2} \log \det(\tilde{M}_j) + \frac{\lambda_1}{2\sigma^2} (\tilde{u}_j \tilde{u}'_j + \text{Tr}(\sigma^2 \tilde{M}_j)) \\ & \quad \left. + \log \frac{z_j}{\kappa_j^\circ} \right\} + (1-z_j) \log \frac{1-z_j}{1-\kappa_j^\circ}. \end{aligned} \tag{38}$$

Last, to obtain (17), we first sum the expressions in (37) for all $j = 1, \dots, p$. Next, we write down the explicit expression of C which involves σ^2 , i.e.,

$$pC_{\sigma^2} = \frac{(np + 2\sigma_a + 2) \log \sigma^2}{2} + \frac{\text{Tr}(X'X) + 2\sigma_b}{2\sigma^2}.$$

Last, we plugging the above expression and solve σ^2 . The solution (18) can be obtained similarly using (38).

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11222-024-10430-8>.

Acknowledgements We would like to warmly thanks Drs. Ryan Martin and Botond Szabó for their helpful suggestions on an early version of this paper. Bo Ning gratefully acknowledges the funding support provided by NASA XRP 80NSSC18K0443. The authors would like to

thank two anonymous reviewers and the Editors for their very constructive comments and efforts on this lengthy work, which greatly improved the quality of this paper.

Funding The research of Ning was partially supported by NIH grant 1R21AI180492-01 and the Individual Research Grant at Texas A&M University.

Declarations

Conflict of interest Ning Ning serves as an Associate Editor for statistics and computing.

References

Avalos-Pacheco, A., Rossell, D., Savage, R.S.: Heterogeneous large datasets integration using Bayesian factor regression. *Bayesian Anal.* **17**, 33–66 (2022)

Banerjee, S., Castillo, I., Ghosal, S.: Bayesian inference in high-dimensional models. Springer volume on Data Science (to Appear) (2021)

Belitser, E., Ghosal, S.: Empirical Bayes oracle uncertainty quantification for regression. *Ann. Stat.* **48**, 3113–3137 (2020)

Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* **518**, 859–877 (2017)

Bouveyron, C., Latouche, P., Mattei, P.-A.: Bayesian variable selection for globally sparse probabilistic PCA. *Electron. J. Stat.* **12**, 3036–3070 (2018)

Cai, T., Ma, Z., Wu, Y.: Optimal estimation and rank detection for sparse spiked covariance matrices. *Probab. Theory Relat. Fields* **161**(3), 781–815 (2015)

Carbonetto, P., Stephens, M.: Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Anal.* **7**(1), 73–108 (2012)

Castillo, I., Roquain, E.: On spike and slab empirical Bayes multiple testing. *Ann. Stat.* (to appear) (2020)

Castillo, I., Schmidt-Hieber, J., van der Vaart, A.: Bayesian linear regression with sparse priors. *Ann. Stat.* **43**, 1986–2018 (2015)

Castillo, I., Szabó, B.: Spike and slab empirical Bayes sparse credible sets. *Bernoulli* **26**, 127–158 (2020)

Castillo, I., van der Vaart, A.: Needles and straw in a haystack: Posterior concentration for possibly sparse sequences. *Ann. Stat.* **40**, 2069–2101 (2012)

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B* **39**, 1–22 (1977)

Erichson, N.B., Zheng, P., Manohar, K., Brunton, S.L., Kutz, J.N., Aravkin, A.Y.: Sparse principal component analysis via variable projection. *SIAM J. Appl. Math.* **80**, 977–1002 (2020)

Gao, C., Zhou, H.H.: Rate-optimal posterior contraction rate for sparse PCA. *Ann. Stat.* **43**, 785–818 (2015)

Ghahramani, Z., Beal, M.: Variational inference for Bayesian mixtures of factor analysers. In: *Advances in Neural Information Processing Systems*, vol. 12. MIT Press, Cambridge (1999)

Guan, Y., Dy, J.: Sparse probabilistic principal component analysis. *Proc. Twelfth Int. Conf. Artif. Intell. Stat.* **5**, 185–192 (2009)

Hansen, B., Avalos-Pacheco, A., Russo, M., Vito, R.D.: Fast variational inference for Bayesian factor analysis in single and multi-study settings. (2023). [arXiv:2305.13188](https://arxiv.org/abs/2305.13188)

Huang, X., Wang, J., Liang, F.: A variational algorithm for Bayesian variable selection. [arXiv:1602.07640](https://arxiv.org/abs/1602.07640) (2016)

- Jammalamadaka, S.R., Qiu, J., Ning, N.: Predicting a stock portfolio with the multivariate Bayesian structural time series model: Do news or emotions matter? *Int. J. Artif. Intell.* **17**(2), 81–104 (2019)
- Jeong, S., Ghosal, S.: Unified Bayesian asymptotic theory for sparse linear regression. [arXiv:2008.10230](https://arxiv.org/abs/2008.10230) (2020)
- Johnson, V.E., Rossell, D.: On the use of non-local prior densities in Bayesian hypothesis tests. *J. R. Stat. Soc. Ser. B* **72**, 143–170 (2010)
- Johnstone, I.M., Lu, A.Y.: On consistency and sparsity for principal components analysis in high dimensions. *J. Am. Stat. Assoc.* **104**, 682–693 (2009)
- Johnstone, I.M., Silverman, B.W.: Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences. *Ann. Stat.* **32**(4), 1594–1649 (2004)
- Li, Z., Safo, S.E., Long, Q.: Incorporating biological information in sparse principal component analysis with application to genomic data. *BMC Bioinformatics*, 12 pages (2017)
- Liu, C., Rubin, D.B., Wu, Y.N.: Parameter expansion to accelerate EM: The PX-EM algorithm. *Biometrika* **85**(4), 755–770 (1998)
- Martin, R., Mess, R., Walker, S.G.: Empirical Bayes posterior concentration in sparse high-dimensional linear models. *Bernoulli* **23**, 1822–1857 (2017)
- Martin, R., Ning, B.: Empirical priors and coverage of posterior credible sets in a sparse normal mean model. *Sankhya A* **82**, 477–498 (2020)
- Ning, B., Ghosal, S., Thomas, J.: Bayesian method for causal inference in spatially-correlated multivariate time series. *Bayesian Anal.* **14**(1), 1–28 (2019)
- Ning, B., Jeong, S., Ghosal, S.: Bayesian linear regression for multivariate responses under group sparsity. *Bernoulli* **26**, 2353–2382 (2020)
- Ning, B.Y.-C.: Empirical Bayes large-scale multiple testing for high-dimensional sparse binary sequences. [arXiv:2307.05943](https://arxiv.org/abs/2307.05943), 80 pages (2023a)
- Ning, N.: Bayesian feature selection in joint quantile time series analysis. *Bayesian Anal.* **1**(1), 1–27 (2023b)
- Ohn, I., Lin, L., Kim, Y.: A Bayesian sparse factor model with adaptive posterior concentration. *Bayesian Anal.* (to Appear), 1–25 (2023)
- Pati, D., Bhattacharya, A., Pillai, N.S., Dunson, D.: Posterior contraction in sparse Bayesian factor models for massive covariance matrices. *Ann. Stat.* **42**(3), 1102–1130 (2014)
- Paul, D.: Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Stat. Sin.* **17**(4), 1617–1642 (2007)
- Qiu, J., Jammalamadaka, S.R., Ning, N.: Multivariate Bayesian structural time series model. *J. Mach. Learn. Res.* **19**(1), 2744–2776 (2018)
- Qiu, J., Jammalamadaka, S.R., Ning, N.: Multivariate time series analysis from a Bayesian machine learning perspective. *Ann. Math. Artif. Intell.* **88**(10), 1061–1082 (2020)
- Rapach, D., Zhou, G.: Sparse macro factors. Available at SSRN: <https://ssrn.com/abstract=3259447> (2019)
- Ray, K., Szabo, B.: Variational Bayes for high-dimensional linear regression with sparse priors. *J. Am. Stat. Assoc.* **117**, 1270–1281 (2022)
- Ročková, V.: Bayesian estimation of sparse signals with a continuous spike-and-slab prior. *Ann. Stat.* **46**(1), 401–437 (2018)
- Ročková, V., George, E.I.: EMVS: The EM approach to Bayesian variable selection. *J. Am. Stat. Assoc.* **109**, 828–846 (2014)
- Ročková, V., George, E.I.: Fast Bayesian factor analysis via automatic rotations to sparsity. *J. Am. Stat. Assoc.* **111**, 1608–1622 (2016)
- Ročková, V., George, E.I.: The spike-and-slab lasso. *J. Am. Stat. Assoc.* **113**, 431–444 (2018)
- Ročková, V., Lesaffre, E.: Incorporating grouping information in Bayesian variable selection with applications in genomics. *Bayesian Anal.* **9**(1), 221–258 (2014)
- She, Y.: Selective factor extraction in high dimensions. *Biometrika* **104**, 97–110 (2017)
- van der Pas, S., Szabó, B., van der Vaart, A.: Uncertainty quantification for the horseshoe (with discussion). *Bayesian Anal.* **12**(4), 1221–1274 (2017)
- Varmuza, K., Filzmoser, P.: Introduction to Multivariate Statistical Analysis in Chemometrics. CRC Press, Boca Raton, FL (2009)
- Wang, Y., Blei, D.M.: Frequentist consistency of variational Bayes. *J. Am. Stat. Assoc.* **114**, 1147–1161 (2019)
- Wang, Z., Gu, Y., Lan, A., Baraniuk, R.: VarFA: A variational factor analysis framework for efficient Bayesian learning analytics. [arXiv:2005.13107](https://arxiv.org/abs/2005.13107), 12 pages (2020)
- Xie, F., Cape, J., Priebe, C.E., Xu, Y.: Bayesian sparse spiked covariance model with a continuous matrix shrinkage prior. *Bayesian Anal.* **17**(4), 1193–1217 (2022)
- Yang, Y., Pati, D., Bhattacharya, A.: α -variational inference with statistical guarantees. *Ann. Stat.* **48**, 886–905 (2020)
- Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B* **68**, 49–67 (2006)
- Zhang, F., Gao, C.: Convergence rates of variational posterior distributions. *Ann. Stat.* **48**, 2180–2207 (2020)
- Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. *J. Comput. Gr. Stat.* 265–286 (2006)
- Zou, H., Xue, L.: A selective overview of sparse principal component analysis. *Proc. IEEE* **106**(8), 1311–1320 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.