Noname manuscript No.
(will be inserted by the editor)

Bayesian inference via projections

Ricardo Silva · Alfredo Kalaitzis

Received: date / Accepted: date

Abstract Bayesian inference often poses difficult computational problems. Even when off-the-shelf Markov chain Monte Carlo (MCMC) methods are available to the problem at hand, mixing issues might compromise the quality of the results. We introduce a framework for situations where the model space can be naturally divided into two components: i. a baseline black-box probability distribution for the observed variables; ii. constraints enforced on functionals of this probability distribution. Inference is performed by sampling from the posterior implied by the first component, and finding projections on the space defined by the second component. We discuss the implications of this separation in terms of priors, model selection, and MCMC mixing in latent variable models. Case studies include probabilistic principal component analysis, models of marginal independence, and a interpretable class of structured ordinal probit models.

Keywords MCMC, optimization, latent variable models, structured covariance matrices.

This work was partially funded by a EPSRC grant EP/J013293/1.

R. Silva
Department of Statistical Science and CSML, University College London
E-mail: ricardo@stats.ucl.ac.uk

A. Kalaitzis
Department of Statistical Science and CSML, University College London
E-mail: a.kalaitzis@ucl.ac.uk

1 Contribution

Bayesian inference raises the computational problems of calculating posterior distributions and expectations of functionals. Markov chain Monte Carlo (MCMC) is a common tool in this case. In many classes of problems, however, the likelihood function is difficult to compute. Using an off-the-shelf method, poor mixing may follow, particularly if latent variables are sampled explicitly [16]. Alternatives include estimating the likelihood function within a MCMC step [1]; using only summary statistics implied by draws from the model [4]; using surrogate likelihood functions [6, 25]. While much of the motivation for the latter is to avoid a complete specification of a model, which calls for explicit assumptions on nuisance parameters, computational considerations are also invoked.

This paper presents a complement to the approaches above. Consider the following setup: modeling multivariate ordinal data with a structured multivariate probit model,

$$\begin{aligned} &\mathbf{Y}^\star \sim \mathcal{N}(0, \Sigma) \\ &Y_i \ = \textstyle\sum_{k=1}^{K_i} k \cdot I(\tau^i_{k-1} \leq Y^\star_i / \sigma_{ii}^{1/2} < \tau^i_k) \end{aligned} \tag{1}$$

for $i = 1, 2, \ldots, p$, where $\mathcal{N}(\mu, \Sigma)$ is the multivariate Gaussian distribution with mean μ and covariance Σ; $I(\cdot)$ is the indicator function; σ_{ij} is the corresponding entry of Σ; observable ordinal variable $Y_i \in \{1, 2, \ldots, K_i\}$ is the result of thresholding $Y^\star_i$ according to a set of thresholds τ^i, such that $\tau^i_0 \equiv -\infty$, $\tau^i_{K_i} \equiv \infty$.

In this setup, and all scenarios thereafter, we assume that the covariance matrix Σ is structured. The assumption is that latent variables $\mathbf{Z}$, representing hidden factors in the world, provide an explanation for the multivariate dependence structure of $\mathbf{Y}^\star$. The mapping from $\mathbf{Z}$ to $\mathbf{Y}^\star$ will impose constraints on Σ, which can then be represented as a function $\Sigma(\theta)$ of some parameter vector θ. Vector θ is composed of continuous and/or discrete random variables. MCMC on $\{\theta, \{\tau_i\}, \mathbf{Z}, \mathbf{Y}^\star\}$ can have considerably worse mixing than MCMC on the space of unconstrained matrices. This is due to the hard constraints mapping θ to Σ, which might involve the implicit definition of discrete variables that control the activation of constraints supported by the data. Also, even if mixing is fast, MCMC does not easily allow for the parallel generation of different structures.

Motivated by such computational problems, we consider an approach where inference is first performed on unconstrained covariance matrices. Randomly sampled matrices from the *saturated* posterior are then mapped via an optimization method to some θ lying on a constrained space. The end result is a posterior distribution over the *target parameter space*. We will illustrate attractive features of this principle with case studies.

In Section 2 we formally describe the general framework of Bayesian Projections. Section 3 provides a simple illustration in Bayesian principal component analysis. In Section 4, we apply Bayesian projections to model selection for marginal independence models. Section 5 demonstrates our main application in modeling questionnaire data with structured latent probit models. Finally, we conclude in Section 6 and discuss some connections to approximate Bayesian computation (ABC) and indirect inference (IL).

2 Inference via Bayesian projections

Let a model be defined by a tuple $(\pi, \Theta, \mathcal{A}, \pi_U)$, where π is the black-box probability model for observable variables $\mathbf{Y}$. Θ is the space of possible reparameterizations of (functionals of) π. $\mathcal{A}$ is a function mapping π and random samples from π_U onto Θ. More generally, we define $\mathcal{A}$ as an algorithm that returns a stationary point (e.g., a local minimum) of some function $d(\cdot)$, which depends on the random initialization variables U sampled from π_U.

Using our example from (1), let π be a probit model for ordinal variables $Y_1, Y_2, \ldots Y_p$; $R(\pi)$, the correlation matrix parameter of the probit model; and Θ, the product space of $p \times v$ matrices L and $v \times v$ diagonal matrices D. Given π, we can define $\theta^\star(\pi)$ as

$$\theta^\star(\pi) = \arg\min_{\{L,D\}\in\Theta} Frob(R(\pi), LDL^T) \qquad (2)$$

where $Frob(A, B)$ is the Frobenious distance,

$$Frob(A, B) \equiv \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{p}(a_{ij} - b_{ij})^2},$$

m_{ij} being the (i, j) entry of matrix M. Algorithm $\mathcal{A}$ solves the optimization problem above. The sign of the entries of L may be underdetermined, and may depend on the random choice U that initializes $\mathcal{A}$. That is,

$$\theta^\star(\pi, U) = \mathcal{A}(\pi, U). \qquad (3)$$

Algorithm $\mathcal{A}$ is meant to be general and cover the cases where the function to be optimized is non-convex, combinatorial, or unidentifiable, with the solution found depending on the random initialization mechanism. In what follows, we will use the notation $\theta^\star$ instead of $\theta^\star(\pi, U)$ when context is clear.

In the context of Bayesian inference, our problem statement is as follows: having observed data $\mathcal{D}$, find the posterior distribution given by the following:

$$\mathcal{P}(\theta^\star \mid \mathcal{D}) = \int \mathcal{P}_{\mathcal{A}}(\theta^\star \mid \pi, u)\mathcal{P}_{\mathcal{D}}(\pi \mid \mathcal{D})\pi_U(u)\ d\pi du \qquad (4)$$

where $\mathcal{P}_{\mathcal{A}}(\theta^\star \mid \pi, u)$ is the point-mass distribution concentrated at the projection (3). The posterior distribution

$$\mathcal{P}_{\mathcal{D}}(\pi \mid \mathcal{D}) \propto \mathcal{L}(\pi; \mathcal{D})\mathcal{P}_0(\pi) \qquad (5)$$

is defined given an appropriate likelihood function $\mathcal{L}(\cdot; \mathcal{D})$ and prior $\mathcal{P}_0(\cdot)$. The result is a posterior distribution over *parameters of interest* in Θ, starting from a *black-box* probability model π. Because $\mathcal{A}$ is defined here in terms of optimizing a (distance) function between functionals of π and a parameter space Θ, we call this inference procedure a *Bayesian projection.*

2.1 Properties

Algorithm 1 shows a high-level description of the Bayesian projections procedure. It can be interpreted as a model decomposition, comprised of a black-box stochastic component π (Step 1), and of a constrained optimization problem defined by $\mathcal{A}$ and Θ (Steps 2-5). Bayesian projections are motivated mainly by scenarios where the sampling procedure in Step 1 mixes well with a relatively simple MCMC algorithm, while a direct MCMC application to the constrained space implied by Θ shows bad mixing behaviour. If $\mathcal{A}$ in Step 4 is reasonably fast and easy to implement, this decomposition would be preferable to the time invested either in designing complicated proposals or using very expensive MCMC procedures in the constrained space.

Interpreting $\mathcal{A}$ as an optimization algorithm that might converge to different local optima depending on random factors U, it is clear that posterior (4) will not converge to a single point even as the size of $\mathcal{D}$ goes to infinite with the model being identifiable. This

```
input : Data matrix $\mathcal{D}$; prior $\mathcal{P}_0(\pi)$; likelihood
        function $\mathcal{L}(\pi; \mathcal{D})$; projection algorithm $\mathcal{A}(\cdot)$;
        distribution $\pi_U(\cdot)$
output: Samples $\theta^{\star(1)}, \ldots, \theta^{\star(M)}$
Use a sampling method to generate $M$ (nearly)
independent samples from $\mathcal{P}(\pi \mid \mathcal{D}) \propto \mathcal{L}(\pi; \mathcal{D})\mathcal{P}_0(\pi)$
for $m$ in $1, 2, \ldots, M$ do
    Generate $U \sim \pi_U(u)$
    $\theta^{\star(m)} \leftarrow \mathcal{A}(\pi^{(m)}, U)$
end
return $\theta^{\star(1)}, \ldots, \theta^{\star(M)}$
```

Algorithm 1: Outline of the Bayesian Projection procedure.

just reflects the difficulty of the problem at hand. With MCMC, this is manifested in the difficulty on exploring multiple modes. Although MCMC can asymptotically sample according to the desired target distribution, this may also require an infinite amount of time. In reality, one might want to restart the chain from multiple points, and accept that the resulting distribution given by any rule that merges the different outcomes is an algorithm that marginalizes over random starting points. That is, if we are solving hard problems, we have to accept that the best we can achieve is a distribution of solutions that depends on the choice of π_U. This optimization view takes Bayesian inference to a more abstract level – one involving a higher-level computational aspect – where π_U represents our prior knowledge on the distribution of reasonable random factors usable by $\mathcal{A}$ to construct the projection. The form of the marginalization in (4) encapsulates this fact. To otherwise claim that a single chain achieves a guaranteed exploration of all modes of the posterior in a reasonable amount of time is to claim one is solving intractable problems in a tractable manner [15].

2.2 Priors and Model Selection

If $\mathcal{A}$ provides a one-to-one mapping between π and $\theta^\star$, then there is an implicit Jacobian matrix for this mapping (with U playing no role in the solution found). In this case, Algorithm 1 can be seen as a way of avoiding an explicit form for the Jacobian. One disadvantage is that the implicit prior on Θ is not obvious, and one has to resort to simulations to understand how $\mathcal{P}_0(\cdot)$ translates into a prior in the Θ space[1]. We argue that any serious application of Bayesian inference in non-trivial problems should always start with simulations from the prior anyway, as priors on parameters are usually factorized, and their joint effect on testable observable conditions on the marginal distribution are hardly obvious.

In particular, it is desirable to set a prior so that the data is allowed to distinguish among models of different structure: for instance, models having covariance matrices represented by decompositions $A + B$, A being a low-rank matrix and B being a sparse matrix [7]. The rank of A and the sparsity level of B are two hard constraints. In our example used at the opening of this Section, the number v of columns of L and D would be such an index of complexity. One way of approaching this setup is by defining a set of projection spaces $\{\Theta^{[1]}, \ldots, \Theta^{[K]}\}$ over the different combinations of rank and sparsity levels. The projection algorithm $\mathcal{A}$ is applied elementwise to this set. Posed in a slightly different way, the output of $\mathcal{A}$ is now a K-dimensional vector. The question is how to decide on the appropriate level of complexity by assessing how well each entry of the output of $\mathcal{A}$ optimizes the target function. We call the *model index*, $M_\Theta \in \{1, \ldots, K\}$, the corresponding index of $\Theta^{[k]}$ one has to choose.

A prior distribution on the model index should be provided without affecting the decoupling introduced by the Bayesian projections framework: we want all inference for $\theta^\star$ to depend on $\mathcal{D}$ only through π. Therefore, $\mathcal{L}(\cdot; \mathcal{D})$ should not depend directly on M_Θ. Loosely following the spirit of [12], we rely on discrepancy measures between the predictive distribution $\mathcal{P}(\pi \mid \mathcal{D})$ and features of the data generating process we would like to represent. In our case, such features are the target functionals defined by $\mathcal{A}$ at different model spaces $\Theta^{[k]}$.

As an abuse of notation, let $d_k(\theta, \pi, U)$ be the value of the objective function at the point optimized by algorithm $\mathcal{A}$ for a fixed (π, U). Let $\theta^{\star[k]}$ be the corresponding projection onto $\Theta^{[k]}$. Random vector $\mathbf{d}$ is defined as $\mathbf{d} \equiv (d_1, d_2, \ldots, d_K)^T$. We define a conditional prior $\mathcal{P}_M(M_\Theta = k \mid \mathbf{d}, \alpha)$, calibrated by some hyperparameter vector α so that

$$\theta^\star = \theta^{\star[M_\Theta]} \tag{6}$$

is our projection of choice. Figure 1 shows a graphical model of the entire process. The algorithm in shown in Algorithm 2.

The definition of $\mathcal{P}_M(\cdot\ |\mathbf{d}, \alpha)$ is problem dependent. In our case study in Section 4, the model index space is organized so that $\Theta^{[k]}$ is nested within $\Theta^{[k+1]}$. Algorithm $\mathcal{A}$ is defined so that the entries of $\mathbf{d}$ do not increase with k. We define $\mathcal{P}_M(\cdot\ |\mathbf{d}, \alpha)$ in a way to tradeoff model complexity and the magnitude of the d_k *for a fixed* π.

Bayesian model selection is in general sensitive to the choice of priors. The Bayesian projections frame-

[1] In fact, this issue is endemic across the literature that considers Bayesian adaptions of frequentist estimators which depend on solving constrained optimization methods [13].

```
input : Data matrix $\mathcal{D}$; prior $\mathcal{P}_0(\pi)$; prior
        $\mathcal{P}_M(M_\Theta \mid \mathbf{d}, \alpha)$; likelihood function $\mathcal{L}(\pi; \mathcal{D})$;
        projection algorithm $\mathcal{A}(\cdot)$; distribution $\pi_U(\cdot)$
output: Samples $\theta^{\star(1)}, \dots, \theta^{\star(M)}$
Use a sampling method to generate $T$ (nearly)
independent samples from $\mathcal{P}(\pi \mid \mathcal{D}) \propto \mathcal{L}(\pi; \mathcal{D})\mathcal{P}_0(\pi)$
for $i$ in $1, 2, \dots, T$ do
    Generate $U \sim \pi_U(u)$
    for $k$ in $1, 2, \dots, K$ do
        $\theta^{\star[k]} \leftarrow \mathcal{A}_k(\pi^{(i)}, U)$
        Calculate $d_k$
    end
    Sample $M_\Theta \sim \mathcal{P}(\mathcal{M} \mid \mathbf{d}, \alpha)$
    $\theta^{\star(i)} \leftarrow \theta^{\star[M_\Theta]}$
end
return $\theta^{\star(1)}, \dots, \theta^{\star(T)}$
```

Algorithm 2: Bayesian Projection procedure with model selection, where $\mathcal{A}_k$ is the k-th entry of the output vector generated by $\mathcal{A}$.

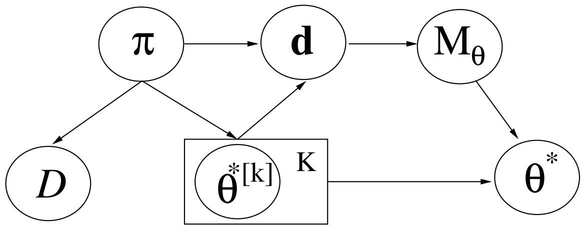


Fig. 1 The full directed acyclic graph representation of the Bayesian projections framework. The box is a plate model representation of a vector of random variables indexed as $1, 2, \dots, K$.

work copes with this sensitivity by funneling the decision process through a projection of interest, thus isolating the contribution of the prior over model (hard) constraints in a more explicit way. Forcing the separation between π and Θ is not only a computational device to alleviate mixing issues and parallelizing some inferential stages, but also a different way of encoding priors within a more emphasis on particular functionals of interest.

3 Case study I: Probabilistic PCA

Our first example will serve as a simple illustration and sanity check, as the likelihood function is easy to compute, and mixing with off-the-shelf MCMC algorithms is not a major issue. We consider the following probabilistic modeling view of principal component analysis (PCA) as a latent variable model [22]:

$$\begin{aligned} \mathbf{Z} &\sim \mathcal{N}(0, \mathbf{I}_{k \times k}) \\ \mathbf{Y} \mid \mathbf{Z} &\sim \mathcal{N}(\mathbf{Z}A, \sigma^2 \mathbf{I}_{p \times p}) \end{aligned} \tag{7}$$

where $\mathbf{I}_{k \times k}$ is a k dimensional identity matrix. The parameters of interest are A and σ^2. This is a simple generalization of PCA, in the sense that the non-trivial stationary points of the likelihood function (for a fixed k) given by the marginal $\mathbf{Y} \sim \mathcal{N}(0, AA^T + \sigma^2 \mathbf{I}_{p \times p})$ are

$$A^\star = V_k(\Lambda_k - {\sigma^2}^\star \mathbf{I}_q)^{1/2}, {\sigma^2}^\star = \frac{1}{p-k} \sum_{j=k+1}^{p} \lambda_j,$$

where the columns of V_k are given by eigenvectors associated with the k largest eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$ of the covariance matrix Σ of $\mathbf{Y}$. Such eigenvalues form the diagonal matrix Λ_k.

The maximum likelihood estimator can then be obtained in closed form by substituting Σ with its empirical estimator. Alternatively one can interpret the negative log-likelihood function as the function minimized by algorithm $\mathcal{A}$,

$$d_{lik}(\pi, \{A, \sigma^2\}) = \log |S(A, \sigma^2)| + \mathrm{Tr}\{S(A, \sigma^2)\Sigma(\pi)^{-1}\}, \tag{8}$$

with $S(A, \sigma^2) \equiv AA^T + \sigma^2 \mathbf{I}_{p \times p}$ and $\Sigma(\pi)$ the covariance matrix of distribution function π. U plays no explicit role here.

The standard Gibbs sampling procedure, augmenting the observed data with latent data $\mathbf{Z}$, can be used to infer posteriors over A and σ^2. It does not, however, make any use the nice analytical properties of (8). Consider instead the application of Algorithm 1 with π being a zero-mean Gaussian with covariance matrix Σ and $\mathcal{P}_0(\Sigma)$ being the prior over correlation matrices introduced by [2]. We compare this inference approach against standard Bayesian inference with independent standard Gaussian priors on each entry of A, an inverse-gamma $(2, 2)$ prior for σ^2, and posterior samples generated by Gibbs sampling.

We generate 100 synthetic datasets by generating a random matrix B of $p \times p$ independently standard Gaussians, setting $\Sigma_{true} = BB^T$ and sampling $2,000$ data points from a Gaussian with zero mean and covariance matrix Σ_{true}. The data is normalized before being given as input to the inference algorithms. We transform each posterior sample from the two methods to correlation matrices, and compare them to the ground truth correlation matrix.

We define the Frobenius error of a method as the Frobenius distance between the true correlation matrix and the posterior expected correlation matrix of the method. The average Frobenius errors over the 100 trials were 1.75 and 1.80 (standard deviations: 0.73, 0.08) for Bayesian projections and the standard Bayesian method, respectively. The KL divergence error is defined similarly, as the KL divergence of a zero-mean Gaussian

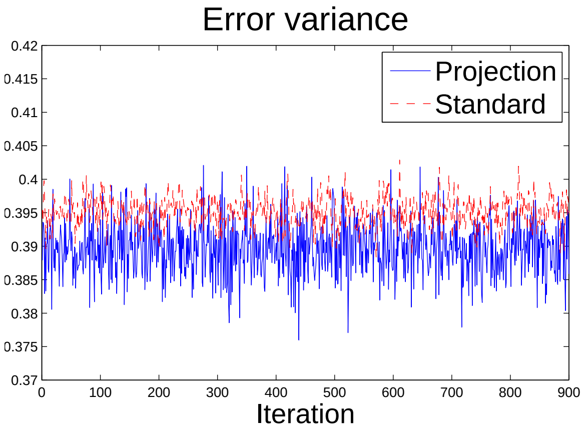


Fig. 2 Bayesian PCA: a run of the Bayesian projection algorithm with likelihood distance against the standard Bayesian inference procedure with Gibbs sampling. The parameter being plotted is σ^2. Discrepancy between methods is small but noticeable, due to the fact that implied priors are different.

model with the true correlation matrix with respect to one with the estimated correlation matrix. In this case, the errors were 9.71 and 9.93 (standard deviations of $1.49, 1.48$). The average computing time on a Xeon E5-1650 at 3.20 Ghz was 0.26 and 1.50, respectively, for Bayesian projections and the standard Bayesian approach. For the projections, this include the (non-trivial) overhead of sampling from the posterior of π. It may be surprising that a Bayesian projection can run faster than a standard Bayesian approach, but in this case we thinned the samples from π from 1000 samples to 450 based on skipping samples to achieve an average absolute one-step autocorrelation of less than 0.05. However, due to the linear cost in time complexity, even if we ran the projection function over all samples we would still achieve a non-trivial reduction in computing time. As a matter of fact, we are being conservative, as the average one-step autocorrelation over the entries of each posterior correlation matrix was around 0.01 for the Bayesian projection output without any thinning, and 0.20 for the standard Gibbs sampler, Figure 2 showing a visual example. On top of this, the loop at the core of Algorithm 1 is amenable to embarrassingly parallel implementations.

4 Case study II: Gaussian marginal independence models

The work in [10] describes the statistical problem of estimating sparse covariance matrices, where some entries σ_{ij} of a matrix Σ are not free parameters, but structurally zero. In Gaussian distributions, this corresponds to a model of marginal independence, where two variables $\{Y_i, Y_j\}$ are independent if and only if $\sigma_{ij} = 0$. A graphical model to represent such a family of constraints is described by [19], with the lack of an edge between corresponding vertices $\{Y_i, Y_j\}$ implying $\sigma_{ij} = 0$, and a *bi-directed* edge $Y_i \leftrightarrow Y_j$ indicating that σ_{ij} is unconstrained. A modification of the inverse Wishart distribution is introduced by [21], and used as a prior over p-dimensional sparse covariance matrices that obey the constraints encoded by a bi-directed graph $\mathcal{G}$:

$$p_{\mathcal{G}IW}(\Sigma; \delta, \mathbf{U}, \mathcal{G}) = \\ \frac{1}{I_{\mathcal{G}}(\delta, \mathbf{U})} |\Sigma|^{-(\delta+2p)/2} \exp\left\{-\frac{1}{2} tr(\{\Sigma\}^{-1}\mathbf{U})\right\}, \\ \Sigma \in M^+(\mathcal{G}), \tag{9}$$

with $\{\delta, \mathbf{U}\}$ playing a role analogous to the hyperparameters of a inverse Wishart. $M^+(\mathcal{G})$ is the cone of positive definite sparse matrices such that $\sigma_{ij} = 0$ if there is no corresponding bi-directed edge in $\mathcal{G}$. There is no analytical form for the normalizing constant $I_{\mathcal{G}}(\delta, \mathbf{U})$.

In [20], we introduce a Gibbs sampler for the graphical structure $\mathcal{G}$ in the Gaussian case. [23] introduces a new variation of the idea, where the graphical structure does not encode hard constraints: instead each edge represents a mixture indicator, with the lack of an edge representing a prior for σ_{ij} strongly concentrated around zero, and the presence of an edge as indicating a high variance prior. Although this prior puts zero probability on $\sigma_{ij} = 0$, it allows the mixture indicators to be sampled independently within a Gibbs sampling step, increasing its computational efficiency. In our experiments, we used a modified version of our Gibbs sampler [20], which allows for positive mass on sparsity patterns[2].

[2] Please notice that [23] correctly indicates that the $\mathcal{G}$-inverse Wishart prior with a δ independent of $\mathcal{G}$ may concentrate mass around a diagonal matrix, as the dimensionality p of the problem increases. However, the empirical problems reported by [23], where the algorithm in [20] basically returns empty graphs in problems of 150 variables and small sample sizes, were unfortunately caused by a bug in our code: once this was corrected, the standard $\mathcal{G}$-inverse Wishart prior had no issues in such problems. The point raised by [23] is still valid, and our procedure from [20] uses a hyperparameter δ that depends on $\mathcal{G}$ – given a baseline hyperparameter δ, we change δ according to $\mathcal{G}$ by subtracting from it the minimum number of non-adjacent nodes among all nodes in $\mathcal{G}$. However, in the experiments described in this paper, this made little difference. Moreover, we further add a small modification to the Gibbs sampler of [20] that is more scalable than the original version: unlike [20], which marginalizes a whole row/column of Σ every time each edge $Y_i \leftrightarrow Y_j$ is sampled, we only marginalize σ_{ij}.

4.1 A Bayesian projection approach

More dramatic computational savings can be achieved by a Bayesian projection approach. We start with a prior $\mathcal{P}_0$ over positive definite matrices. For simplicity, here we assume all matrices are correlation matrices and given the prior described by [2]. Given a sample Σ from a posterior over general correlation matrices, our projection algorithm is simple: if k is the maximum allowed number of edges in the bi-directed graph $\mathcal{G}$, we set to zero all off-diagonal entries of Σ that do not correspond to the top-k covariance entries, measured in terms of their absolute value. Notice this is equivalent to finding the matrix minimizing the Frobenius distance to Σ in the space of correlation matrices that have entries either equal to zero or equal to $(\Sigma)_{ij}$, with no more than k non-zero entries above the diagonal. Since the corresponding matrix may not be positive definite, we apply yet another projection that finds the closest positive semidefinite matrix to the sparsified matrix in terms of the Frobenius distance. This amounts to setting negative eigenvalues of its spectral decomposition to zero.

This idea is similar to several frequentist estimators for sparse covariance matrices which threshold the empirical covariance matrix [5], the thresholding level here being implied by the choice of k. Bayesian inference plays a role through the implicit prior on the sparse matrices, as a function of $\mathcal{P}_0$. What is left is a method to choose k, which has to be done in a non-standard way.

Let $\theta^{\star[k]}$ be the resulting correlation matrix obtained by sparsification at level k (followed by a projection into the positive semidefinite matrix space, where necessary), as a function of a given Σ. Let $d_k \equiv Frob(\theta^{\star[k]}, \Sigma)$ be the k-entry of the K dimensional score vector $\mathbf{d}$, where K is chosen a priori as the maximum number of edges that $\mathcal{G}$ can have[3]. We define the "gradient" and "curvature" vectors $\mathbf{g}$ and $\mathbf{h}$ as follows:

$$\begin{array}{l} g_k \equiv d_k - d_{k+1}, 1 \leq k < K \\ h_k \equiv g_k - g_{k+1}, 1 \leq k < K-1 \end{array} \tag{10}$$

Our model index M_Θ is deterministically chosen as

$$\begin{array}{l} M_\Theta = \min_{k \in 1,2,\ldots,K} k, \\ \text{subject to } |h_k| \leq \alpha \times \sum_{q=1}^{Q} |h_{K-q-1}|/Q \end{array} \tag{11}$$

where $\alpha \geq 0$ and $Q \geq 1$ are hyperparameters. Notice this rule cannot select $k = K-1$ nor $k = K$. Since K influences the thresholding of the model, it is also a hyperparameter for model selection.

The interpretation of this procedure is as follows, considering first the ideal situation where Σ is the population correlation matrix: as constraints on $\Theta^{[k]}$ are relaxed as k increases, the greedy nature of the projection will make g_k approach zero for increasing levels of k. The point where $g_k = 0$ for the first time will be the point where the least complex model fits Σ perfectly.

In practice, Σ is a sample from a distribution over correlation matrices, and g_k will never plateau at zero before reaching maximum complexity. However, when we reach the stage where g_k remains approximately the same as k increases to K, we reach a regime where the order by which edges becomes unimportant. We assume that, by this point in the algorithm, the posterior distribution over these entries is approximately exchangeable and reflects the structurally zero entries. This regime can be detected by $\mathbf{h}$, the second-order differences of the distance vector. The average of the final Q entries of the curvature vector provides a scale for the flat regime[4].

If the black-box distribution is given by $\mathcal{P}_0$, we generate samples of $\mathcal{G}$ from the prior. If this distribution is $\mathcal{P}(\Sigma \mid \mathcal{D})$, we generate samples from the posterior. In principle, hyperpriors for α, Q and K could also be adopted, although we will not explore this idea here.

In our experiments in the next section, we illustrate the behaviour of this model selection procedure in practice. To emphasize again, our goal is not to improve on the Gibbs procedure regarding its statistical properties, but to show we are competitive using less computation.

4.2 Results

We generate 30×30 synthetic bi-directed graphs with corresponding sparse correlation matrices[5]. Three sam-

[3] Because of the positive definite projection, $\mathbf{d}$ is not necessarily monotonically decreasing in its entries, although in practice it will be approximately so.

[4] Other straightforward criteria can be added to this scheme, such as requiring that d_k falls below a minimum acceptable error level. Although this selection rule is loosely inspired by the posterior predictive checks of [12], notice that here we apply this check to each sample of the distribution of Σ instead of samples from the data space.

[5] First, a synthetic graph $\mathcal{G}$ is generated by adding each edge independently with probability 0.05. Observed variables $\mathbf{Y}$ are generated according to the model $\mathbf{Y} = B\mathbf{X} + \mathbf{e}$, where $\mathbf{X}$ is a set of independent standard Gaussian variables. Latent variables $\mathbf{X}$ are introduced such that for each pair $\{Y_i, Y_j\}$ linked by a bi-directed edge, we create a latent variable X_k, sampling the sign of $(B)_{ik}$ uniformly, and the magnitude of $(B)_{ik}$ from a truncated Gaussian in the positive axis with location parameter 0.25 and variance parameter 1. The same applies to $(B)_{jk}$. The entries of B not corresponding to this process are set to zero. Error vector $\mathbf{e}$ is jointly Gaussian with zero mean, and the off-diagonal entries of its covariance given by $BB^T/10$ (elements in the diagonal are set to 1). The corresponding covariance matrix is then rescaled into a correlation matrix

ple sizes are chosen: 100 data points, 1000 and 10000. For each sample size configuration, 100 synthetic data sets are generated. We apply Algorithm 2 to generate posteriors over sparse matrices, with details given in the previous section. For each run, data are standardized according to the empirical covariance matrix and mean. Prior $\mathcal{P}_0$ over full correlation matrices is the one discussed by [2]. For the Gibbs sampling procedure, we used a $\mathcal{G}$-inverse Wishart prior over covariance matrices with parameters $\delta = 3, \mathbf{U} = 3\mathbf{I}_{p\times p}$ and a prior probability 1/30 of an edge being independently added.

Our evaluation metrics, besides wallclock time, are: false positive rate (FPR) (number of edges included by a procedure that is not in the true synthetic graph, divided by the total number of pairs which are not linked in the true graph); false negative rate (FNR) (number of true edges not detected by the procedure, divided by the total number of true edges); absolute false negatives (AFN) (number of true edges not detected by the procedure); and Frobenius distance (FROB) between the implied correlation matrix given by a model and the known synthetic correlation matrix. As baselines, we calculate the FROB measure for the identity matrix $\mathbf{I}_{p\times p}$ and for the full correlation matrix models before any projection. Each metric is computed for each sample generated, where we then average over the samples. We sampled 5000 samples from the posterior of the full model and 5000 samples with the Gibbs sampling based sparse modeling approach, with a burn-in of 1000 steps. We set a maximum of $K = 3 \times 30 = 90$ edges for the Bayesian projections method. A inflating factor $\alpha = 1$ and a tail smoothing factor $Q = 1$ are adopted for model selection.

In the case of synthetic datasets with a sample size of 100, the average run-time for the Gibbs procedure was 117 seconds (standard deviation of 3), with 3.6 seconds for Algorithm 2 (s.d. 0.14), out of which about 60% was due to the initial sampling procedure. The FROB error was 0.89(0.20) for Gibbs and 1.87(0.12) for Algorithm 2. In comparison, the identity matrix model has a FROB metric of 3.19, while the full correlation matrix does worse than that, at 4.27. FPR is 0.01 for Gibbs and 0.03 for Bayesian projections, with a FNR of 0.26(0.12) for Gibbs and 0.47(0.12) for Bayesian projections. The absolute metrics, AFN, are 6.1(3.6) and 10.9(4.6), respectively.

At a sample size of 1000, the Gibbs procedure effectively gets zero FPR, with a FNR of 0.05(0.04), while Bayesian projections gets 0.03 FPR, but with a FNR of 0.13(0.06). The FROB errors are 0.22(0.05) and 0.62(0.05) for Gibbs and Bayesian projections. For comparison, the full correlation matrix models achieves 1.31(0.03), now substantially better than the fully independent model. AFN for Bayesian projections was 3.4 edges, as opposed to 1.07 of the standard Gibbs. The runtime for both methods remain comparable to the previous case. Finally, at a sample of of 10000 both methods achieve near-zero FNR, with Bayesian projections still getting 0.03 FPR. In the FROB metric, results are 0.06(0.01) and 0.19(0.02) for Gibbs and Bayesian projections, respectively. The full correlation model obtains an average FROB of 0.41.

Figures 3 and 4 provide some illustration of the behavior of Bayesian projections. In Figures 3, our implied prior over model structures provides more diffuse posteriors than the one adopted by the more traditional Bayesian approach introduced by [20]. Both posteriors still centered close to the right model for data sets of size 10000, but Bayesian projections does still have a somewhat broad posterior, which in some sense reflects the greater insensitivity of the Frobenius norm compared to the likelihood function of the Gibbs procedure. In Figure 4, our curvature-based model selection criterion can be visualized by posterior simulations: essentially, it slices the curve at the leftmost point where "most" of its mass is around the final entries of the curvature vector. Although we do not provide any formal proofs of consistency, it is clear that as sample size increases, the curvature vector will become flat with high posterior probability at the point where adding more complexity to the model does not decrease the projection error. In the case where the projection algorithm is guaranteed to generate the correct population sparse covariance matrix given the population matrix, the curvature vector will remain at zero starting from the correct model complexity until the end. Consistency still requires the assumption that the rate by which the projection error improves, before reaching the right complexity, will have enough variability so that the model selection criterion will not prematurely stop at a different plateau of the curvature vector – but notice that such a plateau cannot exist in the greedy procedure adopted here. Also, the choice of α for guaranteeing a particular rate of convergence will depend on assumptions on the minimum error decrease before reaching the right complexity, which we also leave for future work. For instance, if the true matrix is very dense, this will lead to situations where rate of change of the gradient vector is overall very small. The choice of α should reflect this knowledge.

5 Case study III: Partition-and-Patch models

In social sciences, latent variable models are commonly used to represent hidden traits of a population [3]. Simple models that explain observed data such as the pat-

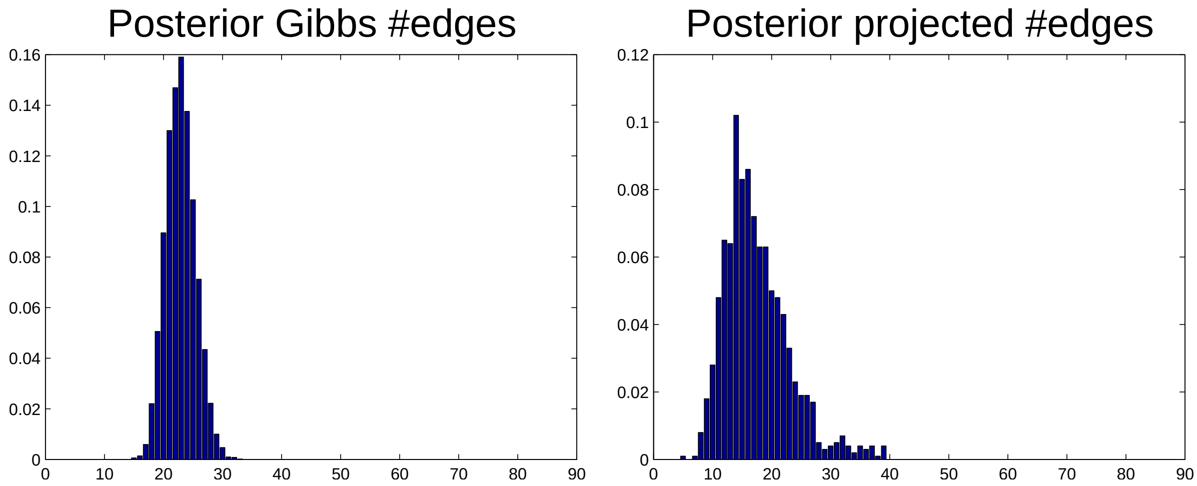


Fig. 3 Synthetic studies with Gaussian marginal independence models: the posterior distribution on the number of edges of the corresponding graphical model for a synthetic study based on a sample size of 100 data points. The left figure is the one obtained by traditional Bayesian modeling with Gibbs sampling, the right one the result of applying model selection by Bayesian projections.

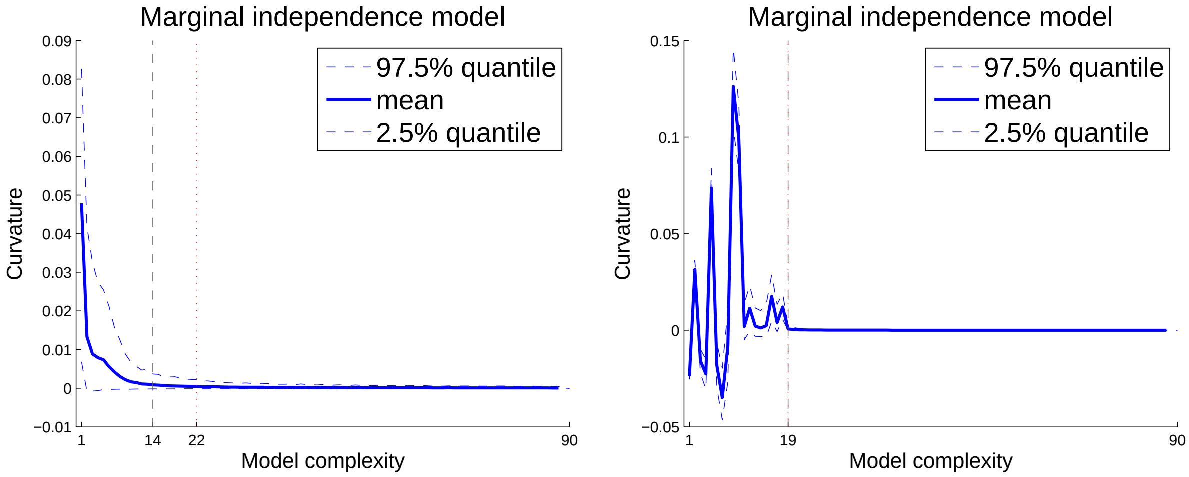


Fig. 4 Posterior distributions over the 90-dimensional curvature vector $\mathbf{h}$ from our synthetic case study. The left figure is an example from a synthetic study with a sample size of 100; in the right figure, the observed data has 10000 data points. The red dotted vertical bar indicates the true number of edges in the synthetic model; the black dashed vertical bar, the mode of the model selection posterior distribution, as defined by Equation (11). The two bars overlap in the rightmost figure.

tern of responses in questionnaires are particularly of interest.

One simple model applicable to this task is the variable clustering method of [17]: each observed variable is a rescaling of one latent variable with added noise,

$$Y_i^\star = \lambda_i Z_{[i]} + \epsilon_i, \tag{12}$$

with $i = 1, 2, \dots, p$ and $Z_{[i]} \in \{Z_1, Z_2, Z_3, \dots\}$, a countably infinite set of marginally independent latent standard Gaussians. Priors over $\{\lambda_1, \dots, \lambda_p\}$ and the variances of the error terms $\{\epsilon_i\}$ are provided, as well as a nonparametric combinatorial prior for the assignment indices $\{[i]\}$. While this model is simple to interpret, it will typically underfit the data as the covariance matrix of the observations is always block-diagonal.

In this section, we introduce a different compromise: starting with the unstructured ordinal model (1), we postulate variables $\mathbf{Y}^\star$ are clustered as in (12), each linked to a single element from a finite pool of latent variables $\mathbf{Z} \sim \mathcal{N}(0, \Sigma_Z)$. Matrix Σ_Z is a full correlation matrix of fixed dimensionality d. We denote as $C_i \in \{1, 2, \dots, d\}$ the assignment of $Y_i^\star$ to a particular latent variable, that is, $Z_{[i]} = Z_{C_i}$. The covariance matrix Σ_ϵ of the error terms is not diagonal, but a sparse covariance matrix as in Section 4. The inference problem is to generate posteriors over $\{\{\lambda_i\}, \{C_i\}, \Sigma_Z, \Sigma_\epsilon\}$ given the observed ordinal data $\mathbf{Y}^{1:n} \equiv \{\mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(n)}\}$. This model space can represent any covariance matrix for $\mathbf{Y}^\star$, avoiding the main reason for underfitting in [17]. Because the model is defined by partitioning the observed variables and matching residual covariances with a sparse error term covariance matrix, we call this the "partition-and-patch" model.

5.1 The Gibbs sampler

One inference procedure is to use independent Gaussian priors on each λ_i, an uniform prior for each C_i, the correlation matrix prior [2] for Σ_Z, a prior over bi-directed structures $\mathcal{G}$ representing the sparsity pattern of Σ_ϵ, and a $\mathcal{G}$-inverse Wishart prior for Σ_ϵ given $\mathcal{G}$. Gibbs sampling can then be performed, sampling $\mathcal{G}$ and Σ_ϵ given latent data $\mathbf{Z}^{1:n}$ and remaining parameters by a direct application of the Gibbs sampling algorithm of Section 4. C_i and λ_i can be sampled by first analytically marginalizing λ_i, sampling C_i from its conditional distribution, then sampling λ_i. Latent data $\mathbf{Y}^\star$ is sampled using standard truncated Gaussian samplers.

A major difficulty is poor mixing. This is partially explained by identifiability issues. Even if the true model is identifiable, sampled candidate models might not be. Parameter identification conditions for the case where Σ_Z is a diagonal matrix are provided by [14], although they would need to be adapted for the case where Σ_Z is arbitrary and the structure unknown. Also, restricting the sampler to generate only identifiable models would complicate it considerably: for instance, changing one C_i at a time is problematic if we forbid any cluster from having a single element only (a unidentifiable structure).

5.2 A Bayesian projection solution

Bayesian projections provide a much more straightforward approach that can tap on existing results commonly associated with frequentist estimation. The Robust PCA approach of [7] provides conditions in which a matrix Σ can be separated into a low rank component A and a sparse component B so that $\Sigma = A + B$. A and B are found as the solution of a optimization problem with a free parameter ω

$$d(\Sigma, A, B) = ||A||_\star + \omega ||B||_1 \tag{13}$$

which is minimized subject to $A + B = \Sigma$; here $||\cdot||_1$ is the L_1 norm and $||\cdot||_\star$ is the nuclear norm [24]. Notice that the definition of the decomposition and the resulting optimization problem are independent of any statistical estimation procedure. Providing the algorithm with samples from the posterior distribution of Σ as given by model (1) and data $\mathbf{Y}^{1:n}$ will generate posterior distributions of matrices A and B for a fixed ω.

The partition-and-patch model can be written so that for $\mathbf{Y}^\star \sim \mathcal{N}(0, \Sigma)$ we have $\Sigma = \Lambda \Sigma_X \Lambda^T + \Sigma_\epsilon$, where $(\Lambda)_{ij} \equiv \lambda_i$ if $C_i = j$, and 0 otherwise. Under the assumption that penalization ω is set such that $A = \Lambda \Sigma_X \Lambda^T$, we can in principle identify Λ and Σ_X. An extra assumption (besides non-degenerate Σ_X) is that all entries of Σ above the diagonal are different in magnitude and each non-empty variable cluster has at least 2 elements. First, since $\lambda_i^2 = (A)_{ii}$, we can cluster Y_i and Y_j as linear functions of the same Z if $|(A)_{ij}| = |\lambda_i \lambda_j|$, as by assumption this will only be true if Y_i and Y_j are indeed in the same cluster. The signs of the coefficients are then set arbitrarily by fixing one coefficient per cluster to be positive and setting the remaining ones according to the sign of the corresponding entries of the clusters. Finally, each entry $(\Sigma_X)_{kl}$ can be identified by finding some pair (i, j) where $C_i = k$, $C_j = l$, since $(A)_{ij} = \lambda_i \lambda_j (\Sigma_X)_{kl}$. Latent variables corresponding to empty clusters can just be ignored.

In practice, given some estimate of Σ (in our case, a sample from a saturated posterior), we can extract from it several candidate low rank matrices A_ω by solving for A via minimization of (13), under a variety of different levels of ω. Once we have a set $\{A_\omega\}$, we choose ω as

$$\omega^\star = \arg\min_\omega \{ \min_{\Lambda, \Sigma_X} Frob(A_\omega, \Lambda \Sigma_X \Lambda^T) \},$$

where the inner optimization could be solved by a variety of methods, including variants of a method of moments procedure. However, in our initial tests, the method of moments variants were not particularly robust to either small sample sizes or small deviations from a proper choice of ω. Instead, we use a iterative coordinate ascent method with a given initialization, which is shown as Algorithm 4 in the Appendix.

The "patching" stage of the procedure can be solved by the method in Section 4, as in synthetic studies we found the method from [24] unable to reliably provide a matrix B with a reasonable match to the sparsity pattern of the true matrix. Moreover, if the goal is only to find a variable clustering, the bi-directed component can be considered as a nuisance parameter and as such this step is completely ignored. This itself can be seen as an advantage over the standard Bayesian procedure, which is required to sample sparse error covariance matrices even if they are nuisance parameters.

Algorithm 3 describes the mapping procedure $\mathcal{A}$ used by Bayesian projections for Partition-and-Patch models.

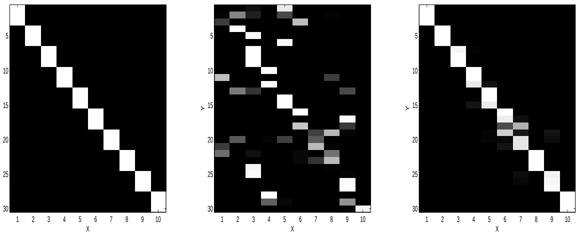
```
input : Matrix $\Sigma$; maximum number of latent
        variables $d$; initial clustering assignment
        $C_1, \ldots, C_p$; set $\Omega$ of candidate penalization
        factors $\omega$; maximum number $K$ of bi-directed
        edges
output: A decomposition $\{\{\lambda_i\}, \{C_i\}, \Sigma_X, \Sigma_\epsilon, \mathcal{G}\}$ of
        input matrix $\Sigma$
for $\omega$ in $\Omega$ do
    Find $A$ and $B$ solving (13) using the proximal
    gradient method of [24]
    Find $\{\{\lambda_i\}, \{C_i\}, \Sigma_X\}$ from $A$ using Algorithm 4
    Let $S \leftarrow Frob(A, \Lambda\Sigma_X\Lambda^T)$
    if $S$ is the smallest error so far then
        Let $\{\{\lambda_i^\star\}, \{C_i^\star\}, \Sigma_X^\star\} \leftarrow \{\{\lambda_i\}, \{C_i\}, \Sigma_X\}$
    end
end
Let $\{\Sigma_\epsilon^\star, \mathcal{G}^\star\}$ be the $K$ models of marginal
independence for $\Sigma - \Lambda^\star\Sigma_X^\star\Lambda^{T\star}$ obtained by greedy
search
return $\{\{\lambda_i^\star\}, \{C_i^\star\}, \Sigma_X^\star, \Sigma_\epsilon^\star, \mathcal{G}^\star\}$
```

Algorithm 3: Given a matrix Σ, generate a decomposition $\{\{\lambda_i\}, \{C_i\}, \Sigma_X\ \Sigma_\epsilon, \mathcal{G}\}$ corresponding to a partition of the variables followed by constructing a sparse correlation matrix of error terms.

5.3 Synthetic experiments

To show the difficulties with the standard Gibbs sampler discussed in Section 5.1, consider the following synthetic study. We generate synthetic models with 10 latent variables and three variables per cluster, and 5 ordinal levels for each observed variable (a total of 30 observed variables)[6].

We generate 100 synthetic examples with 2000 data points. We assume we know there are 10 latent variables in the model. We evaluate how well Gibbs sampling perform in terms of parent reconstruction and the Frobenius error in the reconstruction of $\Lambda\Sigma_X\Lambda^T$. Parent reconstruction is calculated by matching each $C_i^{(m)}$ to the ground truth $1, 2, \ldots, 10$ at each iteration m of the MCMC method, and counting how many of the assignments are incorrect[7]. Figure 5 (b) illustrates the behavior of the Gibbs sampler, where columns are aligned according to the matched clusters. Figure 5 (c) illustrates a typical output for the Bayesian projections procedure, which starts with 1000 samples from the saturated model, and thin them to approximately 200-300 samples by skipping samples until achieving no more than 0.05 units of autocorrelation in the individual entries of Σ. Space Ω is defined to be $\{0.1, 0.2, 0.3, \ldots, 0.9\}$. In principle, a parallel implementation can be used to select ω in Algorithm 3. However, as the model imposes strong constraints on A, we observed such a small variability in ω that we recommend finding it “off-line”: we estimate it once by running Algorithm 3 with the posterior expected value of Σ as given by the saturated model, and fix it when generating projections individually for each posterior sample of Σ.



(a) (b) (c)

Fig. 5 A demonstration of the posterior cluster assignment for each observed variable. True model is shown in (a). Each figure represents a probability of cluster assignment of 30 variables (vertical axis) to 10 clusters (horizontal axis), with black representing a probability of 0, and white a probability of 1. after 5000 iterations (burn-in of 1000) for the Gibbs sampling algorithm (b) and Bayesian projections (c) (approximately 200 samples after thinning 1000 samples from the saturated model).

The average computing time for the full MCMC procedure was 540 seconds, with 97 seconds for Bayesian projections (which includes the initial sampling from the saturated model and the choice of ω), where we did not include a bi-directed structured learning step (the cost of which is negligible compared to the other steps anyway). The average clustering error for the Gibbs sampler was 0.47, with only 0.07 for Bayesian projections. Both methods were initialized by running k-means with the raw ordinal data transposed, so variables are clustered based on their responses (correlation distance and $k = 10$ were chosen). The average clustering error for k-means was 0.38, therefore better than 5000 iterations of Gibbs sampling. Also, Gibbs achieved a Frobenius error of 0.11 between the true A and the estimated $\Lambda\Sigma_X\Lambda^X$, while Bayesian projections achieved 0.03. It should also be mentioned that if we start the Gibbs algorithm with the solution for Bayesian projec-

[6] Coefficients λ_i were generated by sampling its sign uniformly and its magnitude from a truncated Gaussian in the positive axis with location parameter 0.25 and variance parameter 1. Correlation matrix Σ_X was sampled by rescaling an inverse Wishart $(10, 10\mathbf{I})$. Σ_ϵ and $\mathcal{G}$ were sampled using the same scheme as in 4.2. Vector $\{\lambda_i\}$ and Σ_ϵ are re-scaled such that $\lambda_i^2 + (\Sigma_\epsilon)_{ii} = 1$ for all i. Marginal probabilities for each Y_i are generated by generating 5 uniform $(0, 1)$ variables, adding 0.01 to each, and renormalizing them. Thresholds $\{\tau_k^i\}$ are then set accordingly.

[7] Matching is performed by creating a bipartite graph between latent variables $\{Z_i^{(m)}\}$ in the candidate sample and the ground truth $\{Z_i\}$, where an edge $Z_i^{(m)} - Z_j$ is given as a weight the number of common observed variables assigned to $Z_i^{(m)}$ and the number assigned to Z_j in the true model. The resulting matching is given by the Hungarian algorithm.

tions (which can be done using only the posterior expected value of the saturated Σ), then Gibbs performed far better compared to the k-means initialization: it obtains the best clustering error of 0.05 and a Frobenius error of 0.02. However, the computational cost was of 138 seconds per trial for 1000 highly correlated samples (with an effective sample size no greater than the one obtained by Bayesian projections), and without taking into account the initialization costs. Therefore, even without exploiting any of the natural parallelization of Bayesian projections, we obtain comparable performance to a well-initialized Gibbs procedure at a lower computational cost. Also, from the synthetic examples, it is not clear whether the well-initialized Gibbs procedure is exploring the posterior in a sensible way. We will show some evidence to the contrary in the next sections.

5.4 Green consumer data example

Our first real data example is a survey of 330 university students in Greece. The study measures factors that regulate willingness to pay a premium for environmentally friendly ("green") products [3]. This will illustrate the behavior of our procedure with a relatively small sample size and dimensionality.

Each item in the questionnaire asks for an ordinal level of agreement with a different statement, four of which are exemplified below:

1. Batteries cause severe soil pollution.
2. I prefer to buy products in recyclable packaging.
3. I try to cut down on electrical consumption in my household.
4. I am willing to spend an extra 10 euro a week in order to buy less environmentally harmful products.

Items are a 5-point Likert scale response from weakly disagreement to strong agreement. The first question can be interpreted as measuring a level of awareness linking consumerism and pollution; the second question, an item on purchasing alternative products if they have some "green" properties; the third question, on consumption reduction aiming at sustainability; finally, the fourth question more explicitly addresses willingness to spend more in environmentally friendly products. Using the structure of the questionnaire, each of these four types would theoretically describe four clusters of questions, respectively AWARENESS (7 questions), PURCHASE (5 questions), CONSUMPTION (4 questions), WILLINGNESS (2 questions)[8], 18 variables in total.

Figure 6 summarizes our findings regarding variable clustering, given as input the existence of 4 latent variables to be explained by dependent shared factors. Figure 6(a) is a depiction of the grouping of the variables according the theoretical constructs described above, although in principle the WILLINGNESS cluster should not be identifiable from data alone as it contains only 2 elements. Three different runs of the Gibbs sampler are shown in Figures (b)-(d), with some agreements and disagreements with the theoretical clusterings, but with a large variability at 5000 iterations. Notice that Figure (b) reflects the fact that the AWARENESS clustering is sometimes split into two for that run. Figure (e) shows the result for Bayesian projections, where the main difference with respect to the theory is that PURCHASE and WILLIGNESS are hard to separate. This is not too unacceptable as one examines the corresponding questions in detail. In particular, the point estimator obtained by associating with each question its most common cluster gives as the smallest cluster the following two items:

- I switch products for ecological reasons. (*a theoretical PURCHASE variable*)
- I would pay 10% more for groceries that are processed and packaged in an environmentally friendly way. (*a theoretical WILLINGNESS variable*)

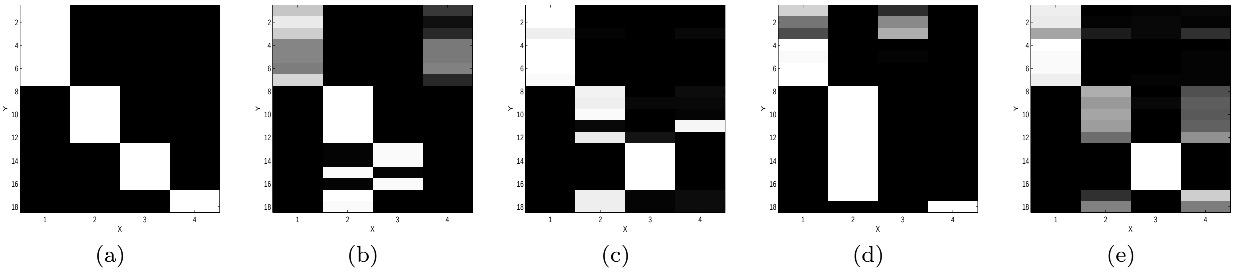
Finally, there remains a considerable amount of uncertainty on the parameters, justifying the generation of a fully Bayesian posterior, as shown in Figure 7. The lack of evident autocorrelation in this plot is the result of these being generated by thinning the posterior samples of the saturated model, as there is no point in consuming further computing time to generate projections on correlated samples. In total, Gibbs sampling consumed 150 seconds, while the saturated sampling consumed 50 seconds. Given the saturated samples, the optimizer consumer another 47 seconds. Effective sample sizes for the $\{\lambda_i\}$ parameters for the first run of Gibbs and Bayesian projections are depicted in Figure 8.

5.5 NHS survey data

The NHS, National Health System, is the public health system of the United Kingdom. The 2009 National NHS Survey [8] collected questionnaires from 156,951 staff members nationwide, asking questions on different aspects of job satisfaction and professional development. We selected 100 questions of the questionnaire[9]. This

[8] We ignore here a non-ordinal count of products student recycle.

[9] The criteria were: questions should either be binary or ordinal, with no "I don't know" items; questions should be



(a) (b) (c) (d) (e)

Fig. 6 The theoretical clustering of the green consumer data is shown in (a). Figures (b), (c) and (d) are different runs of the Gibbs sampler with the same initial cluster structure (but different initial parameters) after 5000 iterations. Figure (e) is the outcome of 1000 iterations of Bayesian projections, using a thinned sample of correlation matrices obtained from a run of 5000 iterations of the saturated model.

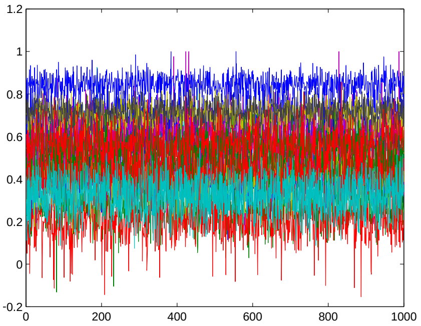


Fig. 7 Evolution of the 18 λ_i coefficients for Bayesian projections.

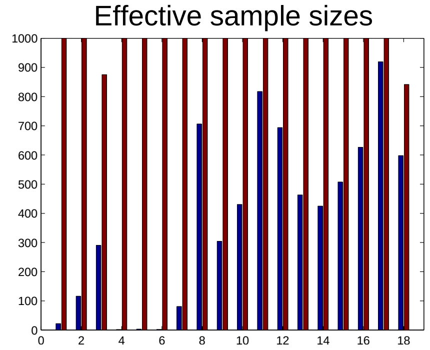


Fig. 8 Effective sample sizes for the 18 coefficients, red bars are results for Bayesian projections while blue are the results for the Gibbs sampler.

lead to a mix of binary and ordinal variables, with ordinal variables framed in a 5-point Likert scale (varying from "Strongly disagree" to "Strongly agree"). We selected a subsample of 100,000 respondents drawn randomly from the population as a training set. Questions are grouped into subsections in the questionnaire, from which we made the choice of fixing the number of latent variables to 20.

Our implementation takes an average of 13 seconds to perform a projection using Algorithm 3 (again, selecting first a single value for ω based on the posterior expected value of Σ in the black-box model and not including the bi-directed selection at this stage), which is approximately 4 times the amount of time taken by a Gibbs step (3.3 seconds) – in both cases, we are not taking into account the time taken to sample the $1,000,000$ underlying variables $\mathbf{Y}^\star$ using a truncated Gaussian sampler, which takes around 4.5 seconds in our naïve implementation. Unlike the Gibbs sampler, though, we once again emphasize that Bayesian projections is easily parallelizable once samples from the black-box model are provided. Moreover, even though a single Gibbs iteration is cheaper, autocorrelation only gets worse as the dimensionality of the problem increases, and the 4-fold speed-up advantage over a Bayesian projection step disappears once effective sample sizes are considered. More importantly, the exploration of the posterior is poor with Gibbs given the combinatorial nature of the partition-and-patch model. As a matter of fact, in a single trial of 1000 iterations, we noticed that the Gibbs sampler, initialized with the output of Bayesian projections given the posterior mean saturated correlation matrix, does not move at all away from the initial structure. The resulting clustering structure inferred by Bayesian projections is shown in Figure 9.

To assess how the bi-directed structure selection works in this case, we perform model selection using Step 9 of Algorithm 3. We perform the model selection procedure for bi-directed structure for $K \leq 300$ (three times the number of observed variables, 100). We set the tail smoothing factor as $Q = 10$. To decide on the prior distribution hyperparameter α, we generate samples from the implied prior for $\alpha = 0.5$ and $\alpha = 0.1$. We visualize the results in Figure 10 and choose $\alpha = 0.1$ as a better choice in our context, as $\alpha = 0.5$ is too sparse according to our expectations.

aimed at all employees and should not lead to follow-up questions such that only a subset of staff are asked to respond; questions should not have more than 50% of missing data.

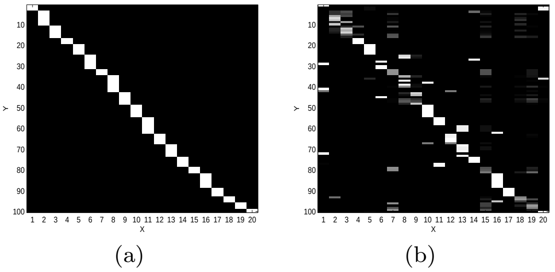


Fig. 9 (a) Theoretical variable clustering structure, as given by the structure of the 2009 NHS questionnaire. (b) Posterior expected clustering assignment given by the Bayesian projections algorithm.

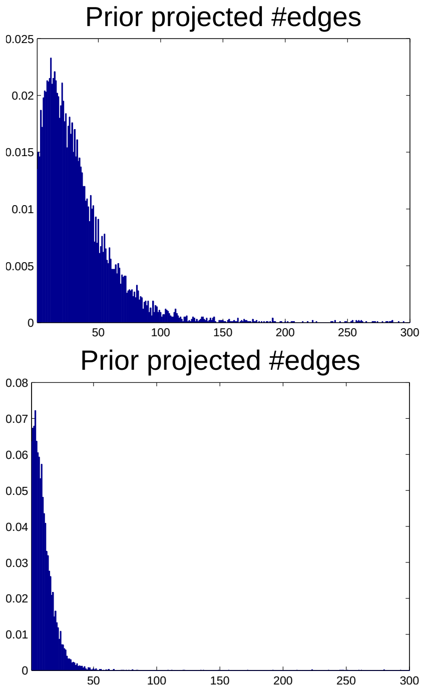


Fig. 10 Samples from the prior over bi-directed graphs implied by generating from Algorithm 2 for a 100-dimensional problem, with tail smoothing factor $Q = 10$ and $K = 300$. The top graph was generated using $\alpha = 0.1$, and the bottom one using $\alpha = 0.5$.

The corresponding curvature plot resulting from our model selection procedure is show in in Figure 11. Finally, to test how adequate this bi-directed structure is, we partially asses its fit by comparing the implied bivariate marginals of the model against the test set constructed from the $\sim 50,000$ data points not used to fit the model. Defining $\theta_{ij}^{kl} \equiv P_\theta(Y_i = k, Y_j = l)$ for some model θ, and $\hat{p}_{ij}^{kl}$ as the corresponding bivariate empirical distribution of the test set, we calculate the χ^2 distance,

$$\chi^2_{ijkl}(\theta, \hat{p}) = \frac{(\hat{p}_{ij}^{kl} - \theta_{ij}^{kl})^2}{\hat{p}_{ij}^{kl}}$$

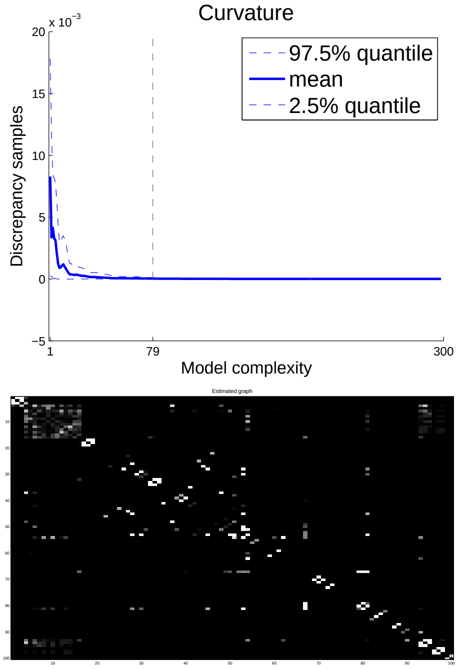


Fig. 11 The curvature plot of the bi-directed model selection procedure for the NHS dataset, and the corresponding posterior expected bi-directed structure. The posterior expected number of edges is 79.

The average $\chi^2_{ijkl}(\theta, \hat{p})$ for our model, using its posterior expected value as a point estimate, was 0.0386. The same result (four digits of precision) was obtained using a fully connected bi-directed graph, implying the amount of information lost by enforcing sparsity is not detectable in this pairwise sense. At the other extreme, using the same estimated clustering and an empty bi-directed graph gives a χ^2 measure of 0.0395. The Gibbs procedure, fixing the clustering structure to be the theoretical clustering while learning the bi-directed structure, gives a χ^2 error of 0.0406.

6 Related Work and Conclusion

Indirect likelihood (IL) [11,9] is an alternative approximate inference framework for cases where the likelihood function is hard to compute but easy to sample from. Here, the prior of the parameters is explicit and used in the generative model to sample an intermediate parameter vector, which can also be in the form of data simulated from the intractable likelihood. IL draws a conceptual decomposition reciprocal to that of Bayesian projections, in the sense that it circumvents the intractable likelihood by isolating the choice of auxiliary likelihood parameters and the mapping from the original to the auxiliary parameters [18].

In the so called *parametric* variants of IL, this mapping is defined via an optimization problem (e.g. maximizing an auxiliary likelihood) as an intermediate step to sampling from the target posterior. *ABC* can be seen as the reduction of IL where the intermediate parameter vector consists of simulated data, and the suitability of each sample depends on its *proximity* to the observed data. That said, Bayesian projections and IL are complementary approaches: in a doubly-intractable scenario where both the likelihood is hard to compute, and sampling from the constrained posterior is challenging, Bayesian projections and IL can be used in unison. We aim to explore this direction in future work.

Bayesian inference via projections is a simple idea that immediately taps in the work developed on optimization methods for frequentist inference. There are open questions on the consistency of the corresponding model selection procedure, including for example potential problems that might affect the method when the number of data points is substantially smaller than the dimensionality of the problem.

Bayesian projections are not meant to be a substitute for other approaches for intractable likelihood problems, as it is not obvious at this stage how it would deal with the variety of problems tackled (in principle) by methods such as ABC. In the same way, strongly informative priors for more complicated models might be harder to encode in the Bayesian projections formulation (although the projection algorithm itself can be defined to incorporate prior information). We only claim we are offering an alternative that might be easily suitable in some scenarios: here, we chose to focus on structured covariance problems within a Gaussian/probit model, but the framework is much more general than that. Although we chose to base our inferences on saturated models, there is nothing in the framework against dealing with some constraints within the MCMC method, leaving the remaining constraints to be dealt with by the projection step. Also, the functionals used by our projection method were covariance matrices in the black-box model: in other cases, such functionals might need to be computed by a Monte Carlo approach, simulating data from the model. The link to pseudo-marginal approaches is somewhat closer in this case, although the analysis of Bayesian projections might be simpler and there is no subsequent MCMC step where such simulations will play a role. Finally, in some problems, the computational cost of a projection step might not be worthwhile if there is already a powerful MCMC method that is both efficient per iteration and of low autocorrelation. For problems with both continuous and discrete random variables, such as those studied in this paper, this might be a challenge, and Bayesian projections provide a simple template that might be immediately applicable to the problem at hand.

Acknowledgements We thank Irini Moustaki for the green consumer data.

References

1. Andrieu, C., Roberts, G.: The pseudo-marginal approach for efficient Monte Carlo computations. The Annals of Statistics **37**, 697–725 (2009)
2. Barnard, J., McCulloch, R., Meng, X.: Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. Statistica Sinica **10**, 1281–1311 (2000)
3. Bartholomew, D., Steele, F., Moustaki, I., Galbraith, J.: Analysis of Multivariate Social Science Data, 2nd edition. Chapman & Hall (2008)
4. Beaumont, M., Zhang, W., Balding, D.: Approximate Bayesian computation in population genetics. Genetics **162**, 2025–2035 (2002)
5. Bickel, P., Levina, E.: Covariance regularization by thresholding. Annals of Statistics **36**, 2577–2604 (2008)
6. Bissiri, P., Holmes, C., Walker, S.: A general framework for updating belief distributions. arXiv:1306.6430 (2013)
7. Candès, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of the ACM **58**(3) (2011)
8. Commission, C.Q., University, A.: Aston Business School, National Health Service National Staff Survey, 2009 [computer file]. Colchester, Essex: UK Data Archive [distributor], October 2010. Available at HTTPS://WWW.ESDS.AC.UK, SN: 6570 (2010)
9. Drovandi, C.C., Pettitt, A.N., Lee, A.: Bayesian indirect inference using a parametric auxiliary model. Statistical Science (2014)
10. Drton, M., Richardson, T.: A new algorithm for maximum likelihood estimation in Gaussian models for marginal independence. Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (2003)
11. Gallant, A.R., McCulloch, R.E.: On the determination of general scientific models with application to asset pricing. Journal of the American Statistical Association **104**(485), 117–131 (2009)
12. Gelman, A., Meng, X., Stern, H.: Posterior predictive assessment of model fitness via realized discrepancies. Statistica Sinica **6**, 733–807 (1996)
13. Gribonval, R., Machart, P.: Reconciling "priors" & "priors" without prejudice? Advances in Neural Information Processing Systems 26 pp. 2193–2201 (2013)
14. Grzebyk, M., Wild, P., Chouaniere, D.: On identification of multi-factor models with correlated residuals. Biometrika **91**, 141–151 (2004)
15. Jerrum, M., Sinclair, A.: The Markov chain Monte Carlo method: an approach to approximate counting and integration. In: D.S. Hochbaum (ed.) Approximation Algorithms for NP-hard Problems, pp. 482–520. PWS Publishing Company (1996)
16. Neal, R.: Probabilistic inference using Markov chain monte carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto (1993)
17. Palla, K., Knowles, D.A., Ghahramani, Z.: A nonparametric variable clustering model. Advances in Neural Information Processing Systems **25**, 2987–2995 (2012)

18. Reeves, R., Pettitt, A.: A theoretical framework for approximate bayesian computation. 20th International Workshop on Statistical Modelling pp. 393–396 (2005)
19. Richardson, T., Spirtes, P.: Ancestral graph Markov models. Annals of Statistics **30**, 962–1030 (2002)
20. Silva, R.: A MCMC approach for learning the structure of Gaussian acyclic directed mixed graphs. In: P. Giudici, S. Ingrassia, M. Vichi (eds.) Statistical Models for Data Analysis, pp. 343–352. Springer (2013)
21. Silva, R., Ghahramani, Z.: The hidden life of latent variables: Bayesian learning with mixed graph models. Journal of Machine Learning Research **10**, 1187–1238 (2009)
22. Tipping, M., Bishop, C.: Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B **61**(3), 611–622 (1999)
23. Wang, H.: Scaling it up: Stochastic search structure learning in graphical models. Bayesian Analysis (To appear)
24. Wright, J., Ganesh, A., Rao, S., Peng, Y., Ma, Y.: Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. Advances in Neural Information Processing Systems 22 pp. 2080–2088 (2009)
25. Yin, G.: Bayesian generalized method of moments. Bayesian Analysis **4**, 191–207 (2009)

A Algorithm for clustering variables in a partition-and-patch model

The algorithm below has three main stages. The first main stage adjusts the cluster assigment and parameters by changing one cluster assignment at a time. The second stage merges clusters with a single element with some other cluster, keeping records of past assignments so that the algorithm does not get stuck in an infinite loop. The third stage splits large clusters in two, again keeping track of which splits happened before.

Line 1 of the algorithm corresponds to setting $|\lambda_i| = \sqrt{(A)_{ii}}$ and setting the signs of each coefficient according to the identifiability conditions discussed in Section 5.2. Line 6 of the algorithm can be efficiently solved in closed form by varying $C_i \in \{1, 2, \dots, p\}$ and taking the derivative with respect to λ_i. In this algorithm, each optimization should be interpreted as keeping all other arguments fixed, optimizing only with respect to the variables on the left-hand side. Entries of Σ_X and $\{\lambda_i\}$ are constrained to the $[-1, 1]$ interval, with no enforcement of a global positive definitiness constraint.

```
   input : Association matrix $A$; initial $\{C_i\}$
   output: A decomposition $\{\{\lambda_i\}, \{C_i\}, \Sigma_X\}$ of $A$
1  Initialize each $\lambda_i$ according to the moment conditions
   of each cluster
2  $\Sigma_X \leftarrow \arg\min Frob(A, \Lambda\Sigma_X\Lambda^T)$
3  while true do
4      while true do
5          for $i = 1, 2, \dots, p$ do
6              $(C_i, \lambda_i) \leftarrow \arg\min Frob(A, \Lambda\Sigma_X\Lambda^T)$
7          end
8          $\Sigma_X \leftarrow \arg\min Frob(A, \Lambda\Sigma_X\Lambda^T)$
9      end
10     if no $C_i$ has changed then
11         break
12     end
13     for all $i$ such that $C_i \neq C_j$ for all $j \neq i$ do
14         $(C_i^{'}, \lambda_i^{'}) \leftarrow \arg\min Frob(A, \Lambda\Sigma_X\Lambda^T)$, among
           those clusters never assigned to $C_i$ before
15     end
16     Change the $(C_i, \lambda_i)$ to $(C_i^{'}, \lambda_i^{'})$ for the $i$ that
       minimizes the $Frob(A, \Lambda\Sigma_X\Lambda^T)$, if any
17     for all $j = 1, 2, \dots, d$ do
18         if there is no empty cluster then
19             break
20         end
21         Let $\mathcal{C}_j$ the variables assigned to cluster $j$
22         if $|\mathcal{C}_j| < 5$ then
23             break
24         end
25         Find the subset $\mathcal{S}_j$ of three elements of $\mathcal{C}_j$
           that minimizes $Frob(A, \Lambda\Sigma_X\Lambda^T)$ by assigning
           them to a previously empty cluster followed by
           the optimization of $\{\{\lambda_i\}, \Sigma_X\}$, such that no
           element of $\mathcal{S}_j$ has been previously split from
           any element of $\mathcal{C}_j \backslash \mathcal{S}_j$
26     end
27     Update $\{\lambda_i\}, \{C_i\}, \Sigma_X\}$ according to the best
       choice of the above loop, if any
28     if no $C_i$ has changed since the beginning of the
       main loop then
29         break
30     end
31 end
32 return $\{\{\lambda_i\}, \{C_i\}, \Sigma_X\}$
```

Algorithm 4: The algorithm for assigning observed variables to single-latent factor clusters, and the corresponding parameters that fit A in a Frobenius sense.