

Bayesian Low-Rank Interpolative Decomposition for Complex Datasets

Jun Lu ¹

¹Trexquant, New York

Correspondence: Jun Lu, jun.lu.locky@gmail.com.

Received: May 17, 2022 Accepted: June 24, 2022 Online Published: June 30, 2022

doi: [10.11114/set.v9i1.5624](https://doi.org/10.11114/set.v9i1.5624) URL: <https://doi.org/10.11114/set.v9i1.5624>

Abstract

In this paper, we introduce a probabilistic model for learning interpolative decomposition (ID), which is commonly used for feature selection, low-rank approximation, and identifying hidden patterns in data, where the matrix factors are latent variables associated with each data dimension. Prior densities with support on the specified subspace are used to address the constraint for the magnitude of the factored component of the observed matrix. Bayesian inference procedure based on Gibbs sampling is employed. We evaluate the model on a variety of real-world datasets including CCLE *EC50*, CCLE *IC50*, CTRP *EC50*, and MovieLens 100K datasets with different sizes, and dimensions, and show that the proposed Bayesian ID GBT and GBTN models lead to smaller reconstructive errors compared to existing randomized approaches.

Keywords: Interpolative decomposition, Low-rank approximation, Bayesian inference, Hierarchical model.

1. Introduction

Matrix factorization methods such as singular value decomposition (SVD), factor analysis, principal component analysis (PCA), and independent component analysis (ICA) have been used extensively over the years to reveal hidden structures of matrices in many fields of science and engineering such as deep learning (Liu et al., 2015), recommendation systems (Comon et al., 2009; Lu, 2021a; 2022), computer vision (Goel et al., 2020), clustering and classification (Li et al., 2009; Wang et al., 2013; Lu, 2021b), collaborative filtering (Marlin, 2003; Lim & Teh, 2007; Mnih & Salakhutdinov, 2007; Raiko et al., 2007; Chen et al., 2009; Brouwer & Lio, 2017; Lu, 2022) and machine learning in general (Lee & Seung, 1999).

Moreover, low-rank approximations are essential in modern data science. Low-rank matrix approximation with respect to the Frobenius norm - minimizing the sum squared differences to the target matrix - can be easily solved with singular value decomposition. For many applications, however, it is sometimes advantageous to work with a basis that consists of a subset of the columns of the observed matrix itself (Halko et al., 2011; Martinsson et al., 2011). The interpolative decomposition (ID) provides one such approximation. Its distinguishing feature is that it reuses columns from the original matrix. This enables it to preserve matrix properties such as sparsity and nonnegativity that also help save space in memory.

Interpolative decomposition is widely used as a feature selection tool that extracts the essence and allows dealing with big data which is originally too large to fit into the RAM. In addition, it removes the non-relevant parts of the data which consist of error and redundant information (Liberty et al., 2007; Halko et al., 2011; Martinsson et al., 2011; Ari et al., 2012; Lu, 2021a). In the meantime, finding the indices associated with the spanning columns is frequently valuable for the purpose of data interpretation and analysis, it can be very useful to identify a subset of the columns that distills the information in the matrix. If the columns of the observed matrix have some specific interpretations, e.g., they are transactions in a transaction dataset, then the columns of the factored matrix in ID will have the same meaning as well. The factored matrix obtained from interpolative decomposition is also numerically stable since its maximal magnitude is limited to a certain range.

On the other hand, matrix factorizations can also be thought of as statistical models in which we seek the factorization that offers the maximum marginal likelihood (MML) for the underlying data (Ari et al., 2012; Brouwer & Lio, 2017). Probabilistic interpretations are investigated for many popular matrix decompositions in the literature such as real-valued matrix factorization, nonnegative matrix factorization (NMF) (Brouwer & Lio, 2017; Lu & Ye, 2022), principal component analysis (Tipping & Bishop, 1999), and generalized to tensor factorizations (Schmidt & Mohamed, 2009). In the meantime, probabilistic models can easily accommodate constraints on the specific range of the factored matrix.

In this light, we focus on the Bayesian ID (BID) of underlying matrices. The ID problem of observed matrix \mathbf{A} can be stated as $\mathbf{A} = \mathbf{C}\mathbf{W} + \mathbf{E}$, where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N] \in \mathbb{R}^{M \times N}$ is approximately factorized into an $M \times K$ matrix $\mathbf{C} \in \mathbb{R}^{M \times K}$ containing K basis columns of \mathbf{A} and a $K \times N$ matrix $\mathbf{W} \in \mathbb{R}^{K \times N}$ with entries no larger than 1 in magnitude¹; the noise is captured by matrix $\mathbf{E} \in \mathbb{R}^{M \times N}$. Training such models amounts to finding the best rank- K approximation to the observed $M \times N$ target matrix \mathbf{A} under the given loss function. Let $\mathbf{r} \in \{0, 1\}^N$ be the state vector with each element indicating the type of the corresponding column, basis column or interpolated (remaining) column: if $r_n = 1$, then the n -th column \mathbf{a}_n is a basis column; if $r_n = 0$, then \mathbf{a}_n is interpolated using the basis columns plus some error term. Suppose further J is the set of the indices of the selected basis columns, I is the set of the indices of the interpolated columns such that $J \cup I = \{1, 2, \dots, N\}$, $J = J(\mathbf{r}) = \{n | r_n = 1\}_{n=1}^N$, and $I = I(\mathbf{r}) = \{n | r_n = 0\}_{n=1}^N$. Then \mathbf{C} can be described as $\mathbf{C} = \mathbf{A}[:, J]$ where the colon operator implies all indices. The approximation $\mathbf{A} \approx \mathbf{C}\mathbf{W}$ can be equivalently stated that $\mathbf{A} \approx \mathbf{C}\mathbf{W} = \mathbf{X}\mathbf{Y}$ where $\mathbf{X} \in \mathbb{R}^{M \times N}$ and $\mathbf{Y} \in \mathbb{R}^{N \times N}$ such that $\mathbf{X}[:, J] = \mathbf{C}$, $\mathbf{X}[:, I] = \mathbf{0} \in \mathbb{R}^{M \times (N-K)}$; $\mathbf{W} = \mathbf{Y}[J, :]$. We also notice that there exists an identity matrix $\mathbf{I} \in \mathbb{R}^{K \times K}$ in \mathbf{W} and \mathbf{Y} :

$$\mathbf{I} = \mathbf{W}[:, J] = \mathbf{Y}[J, J]. \quad (1)$$

To find the low-rank ID of $\mathbf{A} \approx \mathbf{C}\mathbf{W}$ then can be transformed into the problem of finding the $\mathbf{A} \approx \mathbf{X}\mathbf{Y}$ with state vector \mathbf{r} recovering the submatrix \mathbf{C} (see Figure 1). To evaluate the approximation, reconstruction error measured by mean squared error (MSE or Frobenius norm) is minimized (assume K is known):

$$\min_{\mathbf{W}, \mathbf{Z}} \frac{1}{MN} \sum_{n=1}^N \sum_{m=1}^M \left(a_{mn} - \mathbf{x}_m^\top \mathbf{y}_n \right)^2, \quad (2)$$

where \mathbf{x}_m , \mathbf{y}_n are the m -th row and n -th column of \mathbf{X} , \mathbf{Y} respectively. In this paper, we approach the magnitude constraint in \mathbf{W} and \mathbf{Y} by considering the Bayesian ID model as a latent factor model and we describe a fully specified graphical model for the problem and employ Bayesian learning methods to infer the latent factors. In this sense, explicit magnitude constraints are not required on the latent factors, since this is naturally taken care of by the appropriate choice of prior distribution; here we use general-truncated-normal prior.

The main contribution of this paper is to propose a novel Bayesian ID method that is flexible and is called the GBT model. To further favor flexibility and insensitivity on the hyperparameter choices, we also propose the hierarchical model known as the GBTN algorithm, which has simple conditional density forms with little extra computation. Meanwhile, the method is easy to implement. We show that our method can be successfully applied to the large, sparse, and very imbalanced Movie-User dataset, containing 100,000 user/movie ratings.

2. Related Work

The most popular algorithm to compute the low-rank ID approximation is the randomized ID (RID) algorithm (Liberty et al., 2007). At a high level, the algorithm randomly samples $M > K$ columns from \mathbf{A} , uses column-pivoted QR (Lu, 2021a) to select K of those M columns for basis matrix \mathbf{C} , and then computes \mathbf{W} via least squares. M is usually set to $M = 1.2K$ to oversample the columns so as to capture a large portion of the range of matrix \mathbf{A} .

We propose a new Bayesian approach for interpolative decomposition, which is designed to have a strict constraint on the magnitude of the factored matrix \mathbf{W} . The proposed GBT and GBTN models introduce a general-truncated-normal density over the factored matrix \mathbf{W} that finds interpolated columns to reconstruct the observed matrix.

2.1 Probability Distributions

We introduce all notations and probability distribution in this section.

$\mathcal{N}(x | \mu, \tau^{-1}) = \sqrt{\frac{\tau}{2\pi}} \exp\{-\frac{\tau}{2}(x - \mu)^2\}$ is a Gaussian distribution with mean μ and precision τ (variance $\sigma^2 = \tau^{-1}$).

¹A weaker construction is to assume no entry of \mathbf{Y} has an absolute value greater than 2. See proof of the existence of the decomposition in Lu (2021a).

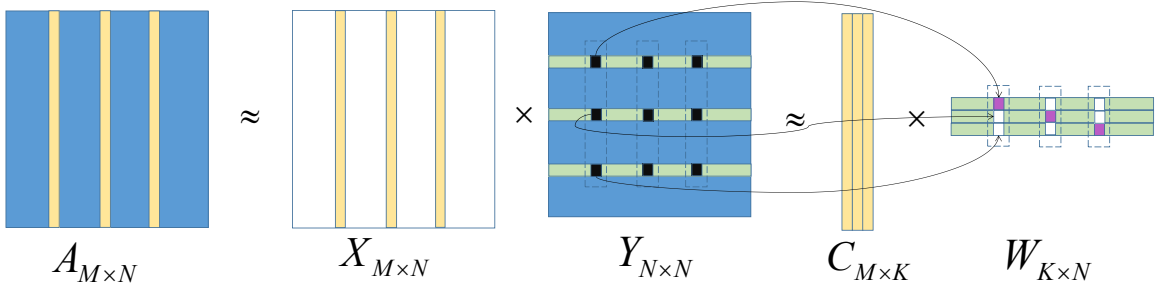


Figure 1. Demonstration of the interpolative decomposition of a matrix where the yellow vector denotes the basis columns of matrix \mathbf{A} , white entries denote zero, purple entries denote one, blue and black entries denote elements that are not necessarily zero. The Bayesian ID models find the approximation $\mathbf{A} \approx \mathbf{X}\mathbf{Y}$ and the post processing procedure finds the approximation $\mathbf{A} \approx \mathbf{C}\mathbf{W}$.

$\mathcal{G}(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp\{-\beta x\} u(x)$ is a Gamma distribution where $\Gamma(\cdot)$ is the gamma function and $u(x)$ is the unit step function that has a value of 1 when $x \geq 0$ and 0 otherwise.

$\mathcal{G}^{-1}(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\{-\frac{\beta}{x}\} u(x)$ is an inverse-Gamma distribution.

$\mathcal{TN}(x|\mu, \tau^{-1}) = \frac{\sqrt{\frac{\tau}{2\pi}} \exp\{-\frac{\tau}{2}(x-\mu)^2\}}{1 - \Phi(-\mu\sqrt{\tau})} u(x)$ is a truncated-normal (TN) with zero density below $x = 0$ and renormalized to integrate to one. μ and τ are known as the ‘‘parent mean’’ and ‘‘parent precision’’. $\Phi(\cdot)$ is the cumulative distribution function of standard normal density $\mathcal{N}(0, 1)$.

$\mathcal{GTN}(x|\mu, \frac{1}{\tau}, a, b) = \frac{\sqrt{\frac{\tau}{2\pi}} \exp\{-\frac{\tau}{2}(x-\mu)^2\}}{\Phi((b-\mu)\sqrt{\tau}) - \Phi((a-\mu)\sqrt{\tau})} u(x|a, b)$ is a general-truncated-normal (GTN) with zero density below $x = a$ or above $x = b$ and renormalized to integrate to one where $u(x|a, b)$ is a step function that has a value of 1 when $a \leq x \leq b$ and 0 otherwise. Similarly, μ and τ are known as the ‘‘parent mean’’ and ‘‘parent precision’’ of the normal distribution. When $a = 0$ and $b = \infty$, the GTN distribution reduces to a TN density.

3. Bayesian Interpolative Decomposition

3.1 Bayesian GBT and GBTN Models for ID

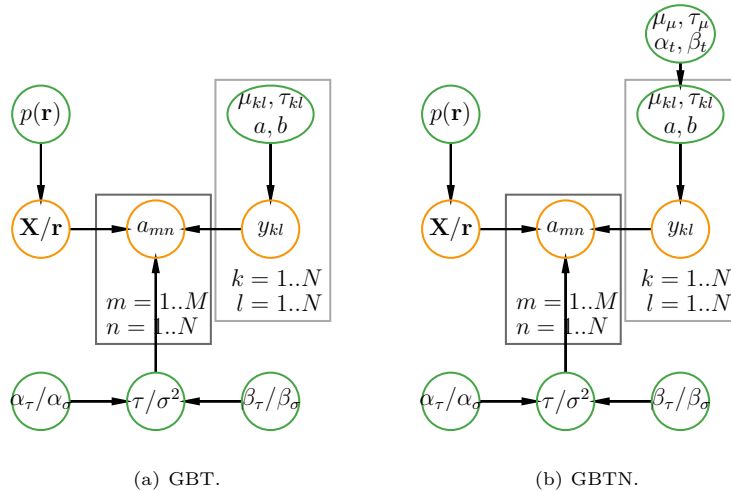


Figure 2. Graphical representation of GBT and GBTN models. Orange circles represent observed and latent variables, green circles denote prior variables, and plates represent repeated variables. ‘‘/’’ in the variable represents ‘‘or’’, and comma ‘‘,’’ in the variable represents ‘‘and’’. Parameters a, b are fixed with $a = -1, b = 1$ in our case; while a weaker construction can set them to $a = -2, b = 2$.

We view the data \mathbf{A} as being produced according to the probabilistic generative process shown in Figure 2. The observed (m, n) -th entry a_{mn} of matrix \mathbf{A} is modeled using a Gaussian likelihood function with variance σ^2 and mean given by the latent decomposition $\mathbf{x}_m^\top \mathbf{y}_n$ (Eq. (2)),

$$p(a_{mn}|\mathbf{x}_m^\top \mathbf{y}_n, \sigma^2) = \mathcal{N}(a_{mn}|\mathbf{x}_m^\top \mathbf{y}_n, \sigma^2). \quad (3)$$

We choose a conjugate prior over the data variance, an inverse-Gamma distribution with shape α_σ and scale β_σ ,

$$p(\sigma^2|\alpha_\sigma, \beta_\sigma) = \mathcal{G}^{-1}(\sigma^2|\alpha_\sigma, \beta_\sigma). \quad (4)$$

While it can also be equivalently given a conjugate Gamma prior over the precision and we shall not repeat the details.

We treat the latent variables y_{kl} 's as random variables. And we need prior densities over these latent variables to express beliefs for their values, e.g., constraint with magnitude smaller than 1 in this context though there are many other constraints (nonnegativity in Lu & Ye (2022), semi-nonnegativity in Ding et al. (2008), discreteness in Gopalan et al. (2014; 2015)). Here we assume further that the latent variable y_{kl} 's are independently drawn from a general-truncated-normal prior

$$p(y_{kl}|\cdot) = \mathcal{GTN}(y_{kl}|\mu_{kl}, (\tau_{kl})^{-1}, a = -1, b = 1). \quad (5)$$

This prior serves to enforce the constraint on the components \mathbf{Y} with no entry of \mathbf{Y} having an absolute value greater than 1, and is conjugate to the Gaussian likelihood. The posterior density is also a general-truncated-normal distribution. While in a weaker construction of interpolative decomposition, the constraint on the magnitude can be loosened to 2; the prior is flexible in that the parameters can be then set to $a = -2, b = 2$ accordingly.

Hierarchical prior. To further favor flexibility, we choose a convenient joint hyperprior density over the parameters $\{\mu_{kl}, \tau_{kl}\}$ of GTN prior in Eq. (5), namely, the GTN-scaled-normal-Gamma (GTNSNG) prior,

$$\begin{aligned} p(\mu_{kl}, \tau_{kl}|\cdot) &= \mathcal{GTNSNG}(\mu_{kl}, \tau_{kl}|\mu_\mu, \frac{1}{\tau_\mu}, \alpha_t, \beta_t) \\ &= \{\Phi((b - \mu_\mu) \cdot \sqrt{\tau_\mu}) - \Phi((a - \mu_\mu) \cdot \sqrt{\tau_\mu})\} \cdot \mathcal{N}(\mu_{kl}|\mu_\mu, (\tau_\mu)^{-1}) \cdot \mathcal{G}(\tau_{kl}|a_t, b_t). \end{aligned} \quad (6)$$

This prior can decouple parameters μ_{kl}, τ_{kl} , and the posterior conditional densities of them are normal, and Gamma respectively due to this convenient scale.

Terminology. There are three types of choices we make that determine the specific type of matrix decomposition model we use, namely, the likelihood function, the priors we place over the factor matrices \mathbf{W} and \mathbf{Z} , and whether we use any further hierarchical priors. We will call the model by the density function in the order of the types of likelihood and priors. For example, if the likelihood function for the model is chosen to be a Gaussian density, and the two prior density functions are selected to be exponential density and Gaussian density functions respectively, then the model will be denoted as Gaussian Exponential-Gaussian (GEG) model. Sometimes, we will put a hyperprior over the parameters of the prior density functions, e.g., we put a Gamma prior over the Gaussian density, then it will further be termed as a Gaussian Exponential-Gaussian Gamma (GEGA) model. In this sense, the proposed hierarchical model is named the GBT and GBTN model where B stands for Beta-Bernoulli density intrinsically.

3.2 Gibbs Sampler

In this paper, we use Gibbs sampling since it tends to be very accurate at finding the true posterior. Other than this method, variational Bayesian inference can be an alternative way but we shall not go into the details. We shortly describe the posterior conditional density in this section. A detailed derivation can be found in Appendix Appendix A. The conditional density of y_{kl} is a general-truncated-normal density. Denote all elements of \mathbf{Y} except y_{kl} as \mathbf{Y}_{-kl} , we then have

$$\begin{aligned} &p(y_{kl}|\mathbf{A}, \mathbf{X}, \mathbf{Y}_{-kl}, \mu_{kl}, \tau_{kl}, \sigma^2) \\ &\propto p(\mathbf{A}|\mathbf{X}, \mathbf{Y}, \sigma^2) \cdot p(y_{kl}|\mu_{kl}, \tau_{kl}) = \prod_{i,j=1}^{M,N} \mathcal{N}(a_{ij}|\mathbf{x}_i^\top \mathbf{y}_j, \sigma^2) \cdot \mathcal{GTN}(y_{kl}|\mu_{kl}, \frac{1}{\tau_{kl}}, -1, 1) \\ &\propto \mathcal{GTN}(y_{kl}|\tilde{\mu}, (\tilde{\tau})^{-1}, a = -1, b = 1), \end{aligned} \quad (7)$$

where $\tilde{\tau} = \frac{\sum_i x_{ik}^2}{\sigma^2} + \tau_{kl}$ is the posterior ‘‘parent precision’’ of the GTN distribution, and

$$\tilde{\mu} = \left(\frac{1}{\sigma^2} \sum_i x_{ik} (a_{il} - \sum_{j \neq k} x_{ij} y_{jl}) + \tau_{kl} \mu_{kl} \right) / \tilde{\tau}$$

is the posterior ‘‘parent mean’’ of the GTN distribution.

Algorithm 1 Gibbs sampler for GBT and GBTN ID models. The procedure presented here may not be efficient but is explanatory. A more efficient one can be implemented in a vectorized manner. By default, uninformative priors are $a = -1, b = 1, \alpha_\sigma = 0.1, \beta_\sigma = 1, (\{\mu_{kl}\} = 0, \{\tau_{kl}\} = 1)$ for GBT, $(\mu_\mu = 0, \tau_\mu = 0.1, \alpha_t = \beta_t = 1)$ for GBTN.

```

1: for  $t = 1$  to  $T$  do
2:   Sample state vector  $\mathbf{r}$  from Eq. (9);
3:   Update matrix  $\mathbf{X}$  by  $\mathbf{A}[:, J]$  where index vector  $J$  is the index of  $\mathbf{r}$  with value 1 and set  $\mathbf{X}[:, I] = \mathbf{0}$  where index vector  $I$  is the index of  $\mathbf{r}$  with value 0;
4:   Sample  $\sigma^2$  from  $p(\sigma^2 | \mathbf{X}, \mathbf{Y}, \mathbf{A})$  in Eq. (10);
5:   for  $k = 1$  to  $N$  do
6:     for  $l = 1$  to  $N$  do
7:       Sample  $y_{kl}$  from Eq. (7);
8:       (GBTN only) Sample  $\mu_{kl}$  from Eq. (11);
9:       (GBTN only) Sample  $\tau_{kl}$  from Eq. (12);
10:    end for
11:  end for
12:  Report loss in Eq. (2), stop if it converges.
13: end for
14: Report mean loss in Eq. (2) after burn-in iterations.
```

Given the state vector $\mathbf{r} = [r_1, r_2, \dots, r_N]^T \in \mathbb{R}^N$, the relation between \mathbf{r} and the index set J is simple; $J = J(\mathbf{r}) = \{n | r_n = 1\}_{n=1}^N$ and $I = I(\mathbf{r}) = \{n | r_n = 0\}_{n=1}^N$. A new value of state vector \mathbf{r} is to select one index j from index set J and another index i from index sets I (we note that $r_j = 1$ and $r_i = 0$ for the old state vector \mathbf{r}) such that

$$\begin{aligned} o_j &= \frac{p(r_j = 0, r_i = 1 | \mathbf{A}, \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji})}{p(r_j = 1, r_i = 0 | \mathbf{A}, \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji})} \\ &= \frac{p(r_j = 0, r_i = 1)}{p(r_j = 1, r_i = 0)} \times \frac{p(\mathbf{A} | \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji}, r_j = 0, r_i = 1)}{p(\mathbf{A} | \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji}, r_j = 1, r_i = 0)}, \end{aligned} \quad (8)$$

where \mathbf{r}_{-ji} denotes all elements of \mathbf{r} except j -th and i -th entries. Trivially, we can set $p(r_j = 0, r_i = 1) = p(r_j = 1, r_i = 0)$. Then the full conditionally probability of $p(r_j = 0, r_i = 1 | \mathbf{A}, \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji})$ can be calculated by

$$p(r_j = 0, r_i = 1 | \mathbf{A}, \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji}) = \frac{o_j}{1 + o_j}. \quad (9)$$

Finally, the conditional density of σ^2 is an inverse-Gamma distribution by conjugacy,

$$p(\sigma^2 | \mathbf{X}, \mathbf{Y}, \mathbf{A}) = \mathcal{G}^{-1}(\sigma^2 | \widetilde{\alpha}_\sigma, \widetilde{\beta}_\sigma), \quad (10)$$

where $\widetilde{\alpha}_\sigma = \frac{MN}{2} + \alpha_\sigma$, $\widetilde{\beta}_\sigma = \frac{1}{2} \sum_{i,j=1}^{M,N} (a_{ij} - \mathbf{x}_i^\top \mathbf{y}_j)^2 + \beta_\sigma$.

Extra updates for the GBTN model. Following the graphical representation of the GBTN model in Figure 2, the conditional density of μ_{kl} can be derived,

$$\begin{aligned} & p(\mu_{kl} | \tau_{kl}, \mu_\mu, \tau_\mu, a_t, b_t, y_{kl}) \\ & \propto \mathcal{GTN}(y_{kl} | \mu_{kl}, (\tau_{kl})^{-1}, a = -1, b = 1) \times \mathcal{GTNSNG}(\mu_{kl}, \tau_{kl} | \mu_\mu, (\tau_\mu)^{-1}, \alpha_t, \beta_t) \\ & \propto \mathcal{N}(\mu_{kl} | \widetilde{m}, (\widetilde{t})^{-1}), \end{aligned} \quad (11)$$

where $\widetilde{t} = \tau_{kl} + \tau_\mu$, and $\widetilde{m} = (\tau_{kl} y_{kl} + \tau_\mu \mu_\mu) / \widetilde{t}$ are the posterior precision and mean of the normal density. Similarly, the conditional density of τ_{kl} is,

$$\begin{aligned} & p(\tau_{kl} | \mu_{kl}, \mu_\mu, \tau_\mu, a_t, b_t, y_{kl}) \\ & \propto \mathcal{GTN}(y_{kl} | \mu_{kl}, (\tau_{kl})^{-1}, a = -1, b = 1) \times \mathcal{GTNSNG}(\mu_{kl}, \tau_{kl} | \mu_\mu, (\tau_\mu)^{-1}, \alpha_t, \beta_t) \\ & \propto \mathcal{G}(\tau_{kl} | \widetilde{a}, \widetilde{b}), \end{aligned} \quad (12)$$

where $\widetilde{a} = a_t + 1/2$ and $\widetilde{b} = b_t + \frac{(y_{kl} - \mu_{kl})^2}{2}$ are the posterior parameters of the Gamma density. The full procedure is then formulated in Algorithm 1.

3.3 Aggressive Update

In Algorithm 1, we notice that we set $\mathbf{X}[:, I] = \mathbf{0}$ when the new state vector \mathbf{r} is sampled. However, in the next iteration step, the index set I may be updated in which case one entry i of I may be altered to have value of 1:

$$r_i = 0 \rightarrow r_i = 1.$$

This may cause problems in the update of y_{kl} in Eq. (7), a zero i -th column in \mathbf{X} cannot update \mathbf{Y} accordingly. One solution is to record a proposal state vector \mathbf{r}_2 and a proposal factor matrix \mathbf{X}_2 from the \mathbf{r}_2 vector. When the update in the next iteration selects the old state vector \mathbf{r} , the factor matrix \mathbf{X} is adopted to finish the updates; while the algorithm chooses the proposal state vector \mathbf{r}_2 , the proposal factor matrix \mathbf{X}_2 is applied to do the updates. We call this procedure the aggressive update. The aggressive sampler for the GBT model is formulated in Algorithm 2. For the sake of simplicity, we don't include the sampler for the GBTN model because it may be done in a similar way.

Algorithm 2 Aggressive Gibbs sampler for GBT ID model. The procedure presented here may not be efficient but is explanatory. A more efficient one can be implemented in a vectorized manner. By default, uninformative priors are $a = -1, b = 1, \alpha_\sigma = 0.1, \beta_\sigma = 1, (\{\mu_{kl}\} = 0, \{\tau_{kl}\} = 1)$ for GBT.

- 1: for $t = 1$ to T do
 - 2: Sample state vector \mathbf{r} from $\{\mathbf{r}_1, \mathbf{r}_2\}$ by Eq. (9);
 - 3: Decide \mathbf{Y} : $\mathbf{Y} = \mathbf{Y}_1$ if \mathbf{r} is \mathbf{r}_1 ; $\mathbf{Y} = \mathbf{Y}_2$ if \mathbf{r} is \mathbf{r}_2 ;
 - 4: Update state vector $\mathbf{r}_1 = \mathbf{r}$;
 - 5: Sample proposal state vector \mathbf{r}_2 based on \mathbf{r} ;
 - 6: Update matrix \mathbf{X} by $\mathbf{r} = \mathbf{r}_1$;
 - 7: Update proposal \mathbf{X}_2 by \mathbf{r}_2 ;
 - 8: Sample σ^2 from $p(\sigma^2 | \mathbf{X}, \mathbf{Y}, \mathbf{A})$ in Eq. (10);
 - 9: Sample $\mathbf{Y}_1 = \{y_{kl}\}$ using \mathbf{X} ;
 - 10: Sample $\mathbf{Y}_2 = \{y_{kl}\}$ using \mathbf{X}_2 ;
 - 11: Report loss in Eq. (2), stop if it converges.
 - 12: end for
 - 13: Report mean loss in Eq. (2) after burn-in iterations.
-

Table 1. Overview of the CCLE EC50, CCLE IC50, CTRP EC50, and MovieLens 100K datasets, giving the number of rows, columns (after copying twice for redundancy), and the fraction of entries that are observed. The MovieLens 100K dataset has a larger fraction of unobserved entries.

Dataset	Rows	Columns	Fraction of observations
CCLE EC50	502	48	0.632
CCLE IC50	504	48	0.965
CTRP EC50	887	1090	0.801
MovieLens 100K	943	2946	0.072

3.4 Post Processing

The Gibbs sampling algorithm finds the approximation $\mathbf{A} \approx \mathbf{X}\mathbf{Y}$ where $\mathbf{X} \in \mathbb{R}^{M \times N}$ and $\mathbf{Y} \in \mathbb{R}^{N \times N}$. As stated above, the redundant columns in \mathbf{X} and redundant rows in \mathbf{Y} can be removed by the index vector J :

$$\mathbf{C} = \mathbf{X}[:, J] = \mathbf{A}[:, J], \quad \mathbf{W} = \mathbf{Y}[J, :].$$

Since the submatrix $\mathbf{Y}[J, J] = \mathbf{W}[:, J]$ (Eq. (1)) from the Gibbs sampling procedure is not enforced to be an identity matrix (as required in the interpolative decomposition). We need to set it to be an identity matrix manually. This will basically reduce the reconstructive error further. The post processing procedure is shown in Figure 1.

4. Experiments

To evaluate the strategy and demonstrate the main advantages of the proposed Bayesian ID method, we conduct experiments with different analysis tasks; and different data sets including Cancer Cell Line Encyclopedia (CCLE EC50 and CCLE IC50 datasets, (Barretina et al., 2012)) and Cancer Therapeutics Response Portal (CTRP EC50, (Seashore-Ludlow et al., 2015)) from bioinformatics, and MovieLens 100K from movie ratings for different users

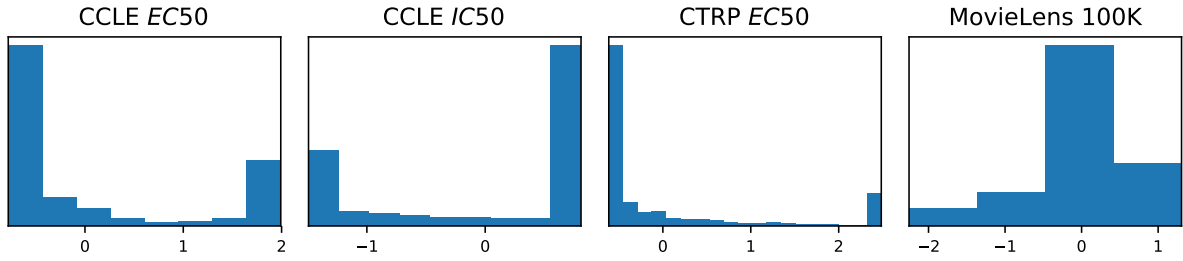
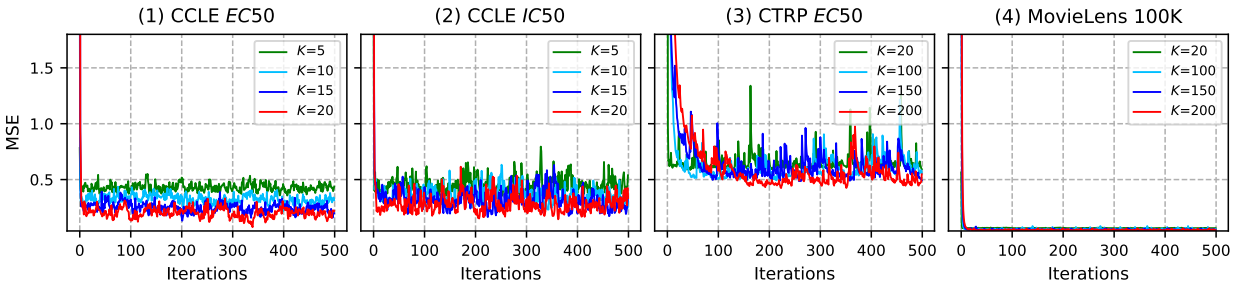
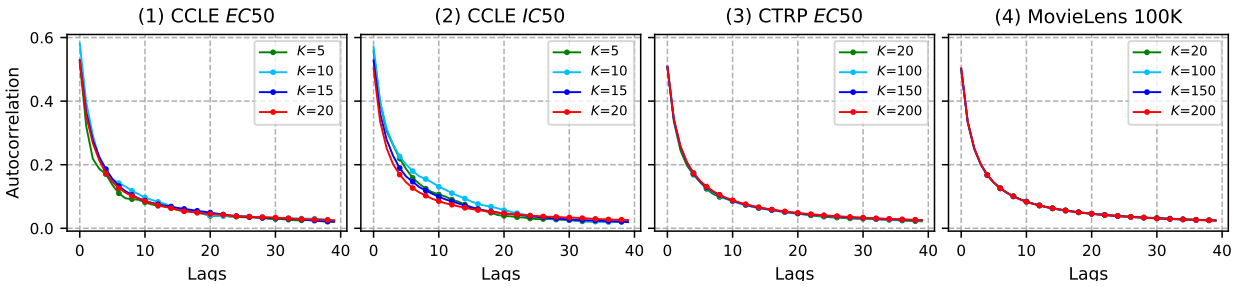


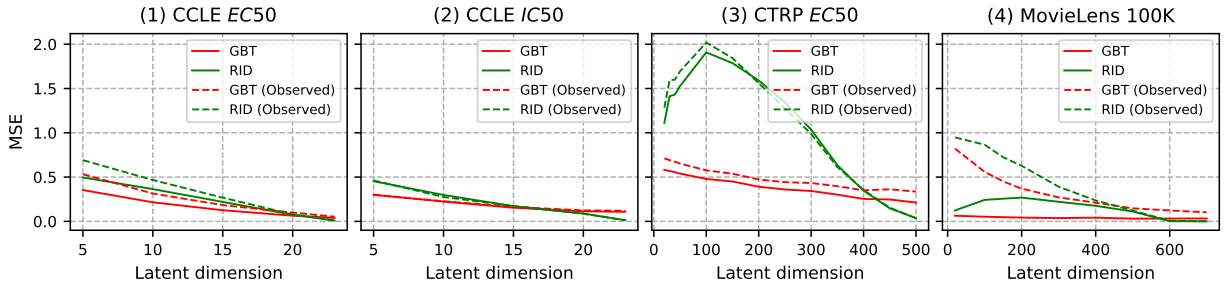
Figure 3. Data distribution of CCLE *EC50*, CCLE *IC50*, CTRP *EC50*, and MovieLens 100K datasets. The former three datasets have complex distributions and the last one has a Gaussian-like distribution.



(a) Convergence of the models on the CCLE *EC50*, CCLE *IC50*, CTRP *EC50*, and MovieLens 100K datasets, measuring the training data fit (mean squared error). The algorithm almost converges in less than 50 iterations.



(b) Averaged autocorrelation coefficients of samples of y_{kl} computed using Gibbs sampling on the CCLE *EC50*, CCLE *IC50*, CTRP *EC50*, and MovieLens 100K datasets.



(c) Reconstructive analysis on the CCLE *EC50*, CCLE *IC50*, CTRP *EC50*, and MovieLens 100K datasets with increasing latent dimension K .

Figure 4. Convergence results (upper), sampling mixing analysis (middle), and reconstructive results (lower) on the CCLE *EC50*, CCLE *IC50*, CTRP *EC50*, and MovieLens 100K datasets for various latent dimensions.

(Harper & Konstan, 2015). Following Brouwer & Lio (2017), we preprocess these datasets by capping high values to 100 and undoing the natural log transform for the former three datasets. Then we standardize to have zero mean and unit variance for all datasets and fill missing entries by 0. Moreover, the user vectors or movie vectors with less than 3 observed entries are cleaned in the MovieLens 100K dataset. Finally, we copy every column twice in order to increase redundancy in the matrix. A summary of the four datasets can be seen in Table 1 and their distributions are shown in Figure 3. The MovieLens 100K dataset is unbalanced in that it has a larger fraction of unobserved entries with only 7.2% being observed so that the algorithm may tend to simply predict all entries to be zero.

In all scenarios, the same parameter initialization is adopted when conducting different tasks. Experimental evidence reveals that post processing procedure can increase performance to a minor level, and that the outcomes of the GBT and GBTN models (aggressive and non-aggressive versions) are relatively similar. For clarification, we only

Table 2. Mean squared error measure with various latent dimension K parameters for CCLE $EC50$, CCLE $IC50$, CTRP $EC50$, and MovieLens 100K datasets.

Data	K	GBT	RID
CCLE $EC50$	5	0.35	0.50
	10	0.22	0.36
	15	0.13	0.22
	20	0.07	0.08
CCLE $IC50$	5	0.30	0.45
	10	0.23	0.30
	15	0.16	0.17
	20	0.13	0.09
CTRP $EC50$	20	0.58	1.11
	100	0.48	1.91
	150	0.47	1.79
MovieLens 100K	20	0.06	0.12
	100	0.05	0.24
	150	0.05	0.26

Table 3. Magnitude that exceeds 1 in randomized ID algorithm with various latent dimensions K for different datasets.

Dataset	$K = 5$	$K = 10$	$K = 15$
CCLE $EC50$	0.0	4.44e-16	4.44e-16
CCLE $IC50$	1.99e-2	4.44e-16	6.66e-16
Dataset	$K = 20$	$K = 100$	$K = 150$
CTRP $EC50$	4.44e-16	2.69e-2	1.99e-15
MovieLens 100K	2.22e-16	6.66e-16	8.88e-16

provide the findings of the GBT model after an aggressive update and post processing. We compare the results with randomized ID (RID) algorithm (Liberty et al., 2007). In a wide range of scenarios across various datasets, the GBT model improves reconstructive error and leads to performances that are as good or better than the randomized ID method in low-rank ID approximation. We also notice that all the four datasets contain missing entries that are filled by zero values. When the fraction of unobserved data is large, the model can simply predict the entries with zero. So we also report the mean squared error for observed entries, termed as GBT (Observed) for the proposed GBT model, and RID (Observed) for the randomized ID model.

In order to measure overall decomposition performance, we use mean square error (MSE, Eq. (2)), which measures the similarity between the true and reconstructive; the smaller the better performance.

4.1 Hyperparameters

In this experiments, we use $a = -1, b = 1, \alpha_\sigma = 0.1, \beta_\sigma = 1, (\{\mu_{kl}\} = 0, \{\tau_{kl}\} = 1)$ for GBT, $(\mu_\mu = 0, \tau_\mu = 0.1, \alpha_t = \beta_t = 1)$ for GBTN. These are very weak prior choices and the models are not sensitive to them. As long as the hyperparameters are set, the observed or unobserved variables are initialized from random draws as this initialization procedure provides a better initial guess of the right patterns in the matrices. In all experiments, we run the Gibbs sampler 500 iterations with a burn-in of 100 iterations and a thinning of 5 iterations as the convergence analysis shows the algorithm can converge in less than 50 iterations.

4.2 Convergence Analysis

Firstly we show the convergence in terms of iterations on the CCLE $EC50$, CCLE $IC50$, CTRP $EC50$ and MovieLens 100K datasets. We run each model with $K = 5, 10, 15, 20$ for the CCLE $EC50$ and CCLE $IC50$ datasets, and $K = 20, 100, 150, 200$ for the CTRP $EC50$ and MovieLens 100K datasets; and the loss is measured by mean squared error (MSE). Figure 4(a) shows the average convergence results. Figure 4(b) shows autocorrelation coefficients of samples computed using Gibbs sampling. The coefficients are less than 0.1 when the lags are more than 10 showing the mixing of the Gibbs sampler is good. In all experiments, the algorithm converges in less than 50 iterations. On the CCLE $EC50$ and MovieLens 100K datasets, the sampling are less noisy; while on the CCLE $IC50$ and CTRP $EC50$ datasets, the sampling seems to be noisy.

4.3 Comparison with Randomized ID

Comparative results for the proposed GBT model and the RID model on the four datasets are given in Figure 4(c) and Table 2. We also observe that, in all experiments, the MSEs of the observed entries are worse than those of all the entries. In the results of CCLE $EC50$ and CCLE $IC50$ datasets, the proposed GBT model recovers smaller MSE when the latent dimension is lower than 15 (for both the observed ones and all entries); However, when K tends to be closer to the rank (24 in this case), the RID becomes better. Similar results can also be observed on the CTRP $EC50$ dataset whose rank is 545, and on the MovieLens 100K dataset whose rank is 943. This demonstrates that

our proposed models are more competitive when we need a low-rank ID approximation that can distill data to a great extent.

For the proposed Bayesian GBT and GBTN models, in all circumstances, the maximal magnitude of the factored matrix \mathbf{W} is no greater than 1 because the prior densities in Bayesian ID models guarantee the magnitude constraint. The randomized ID, on the other hand, finds a weaker interpolative decomposition with a maximal magnitude of no more than 2. Table 3 shows the magnitudes of the randomized ID model for different datasets and different latent dimensions. We may notice that when $K = 100$ on the CTRP *EC50* dataset, the maximal magnitude is about 1.0269. Though this is not a big issue in many applications, it has potential problems for other applications that have a high requirement for numerical stability. Advani & O’Hagan (2021) also reports that the randomized ID may have a magnitude larger than 167 making it less numerical stable.

5. Conclusion

The aim of this paper is to solve the numerical stability issue of the randomized algorithms in computing the low-rank ID approximation. We propose a simple and computationally efficient algorithm that requires little extra computation and is easy to implement for interpolative decomposition. Overall, we show that the proposed GBT and GBTN models are versatile algorithms that have good convergence results and better reconstructive performances on both sparse and dense datasets, especially for low-rank approximation ID. GBT and GBTN are able to force the magnitude of the factored matrix to be no greater than 1 such that numerical stability is guaranteed.

References

- Advani, R., & O’Hagan, S. (2021). Efficient algorithms for constructing an interpolative decomposition. arXiv preprint arXiv:2105.07076.
- Ari, I., Cemgil, A. T., & Akarun, L. (2012). Probabilistic interpolative decomposition. In 2012 IEEE International Workshop on Machine Learning for Signal Processing, pages 1–6. IEEE.
- Barretina, J., Caponigro, G., Stransky, N., Venkatesan, K., Margolin, A. A., Kim, S., Wilson, C. J., Lehár, J., Kryukov, G. V., Sonkin, D., et al. (2012). The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391), 603–607.
- Brouwer, T., & Lio, P. (2017). Prior and likelihood choices for Bayesian matrix factorisation on small datasets. arXiv preprint arXiv:1712.00288.
- Chen, G., Wang, F., & Zhang, C. (2009). Collaborative filtering using orthogonal nonnegative matrix tri-factorization. *Information Processing & Management*, 45(3), 368–379.
- Comon, P., Luciani, X., & De Almeida, A. L. (2009). Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 23(7-8), 393–405.
- Ding, C. H., Li, T., & Jordan, M. I. (2008). Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1), 45–55.
- Goel, A., Tung, C., Lu, Y.-H., & Thiruvathukal, G. K. (2020). A survey of methods for low-power deep learning and computer vision. In 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), pages 1–6. IEEE.
- Gopalan, P., Ruiz, F. J., Ranganath, R., & Blei, D. (2014). Bayesian nonparametric poisson factorization for recommendation systems. In *Artificial Intelligence and Statistics*, pages 275–283. PMLR.
- Gopalan, P., Hofman, J. M., & Blei, D. M. (2015). Scalable recommendation with hierarchical poisson factorization. In *UAI*, pages 326–335.
- Halko, N., Martinsson, P.-G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2), 217–288.
- Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4), 1–19.
- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788–791.
- Li, T., Zhang, Y., & Sindhwani, V. (2009). A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 244–252.
- Liberty, E., Woolfe, F., Martinsson, P.-G., Rokhlin, V., & Tygert, M. (2007). Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51), 20167–20172.
- Lim, Y. J., & Teh, Y. W. (2007). Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD cup and workshop*, volume 7, pages 15–21. Citeseer.

- Liu, B., Wang, M., Foroosh, H., Tappen, M., & Pensky, M. (2015). Sparse convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 806–814.
- Lu, J. (2021)a. Numerical matrix decomposition and its modern applications: A rigorous first course. arXiv preprint arXiv:2107.02579.
- Lu, J. (2021)b. A survey on Bayesian inference for Gaussian mixture model. arXiv preprint arXiv:2108.11753.
- Lu, J. (2022). Matrix decomposition and applications. arXiv preprint arXiv:2201.00145.
- Lu, J., & Ye, X. (2022). Flexible and hierarchical prior for Bayesian nonnegative matrix factorization. arXiv preprint arXiv:2205.11025.
- Marlin, B. M. (2003). Modeling user rating profiles for collaborative filtering. Advances in neural information processing systems, 16.
- Martinsson, P.-G., Rokhlin, V., & Tygert, M. (2011). A randomized algorithm for the decomposition of matrices. Applied and Computational Harmonic Analysis, 30(1), 47–68.
- Mnih, A., & Salakhutdinov, R. R. (2007). Probabilistic matrix factorization. Advances in neural information processing systems, 20.
- Raiko, T., Ilin, A., & Karhunen, J. (2007). Principal component analysis for large scale problems with lots of missing values. In European Conference on Machine Learning, pages 691–698. Springer.
- Schmidt, M. N., & Mohamed, S. (2009). Probabilistic non-negative tensor factorization using Markov chain Monte Carlo. In 2009 17th European Signal Processing Conference, pages 1918–1922. IEEE.
- Seashore-Ludlow, B., Rees, M. G., Cheah, J. H., Cokol, M., Price, E. V., Coletti, M. E., Jones, V., Bodycombe, N. E., Soule, C. K., Gould, J., et al. (2015). Harnessing connectivity in a large-scale small-molecule sensitivity dataset. Cancer discovery, 5(11), 1210–1223.
- Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61(3), 611–622.
- Wang, J. J.-Y., Wang, X., & Gao, X. (2013). Non-negative matrix factorization by maximizing correntropy for cancer clustering. BMC bioinformatics, 14(1), 1–11.

Appendix A. Derivation of Gibbs Sampler for Bayesian Interpolative Decomposition

In this section, we give the derivation of the Gibbs sampler for the proposed Bayesian ID models. As shown in the main paper, the observed data (m, n) -th entry a_{mn} of matrix \mathbf{A} is modeled using a Gaussian likelihood with variance σ^2 and mean given by the latent decomposition $\mathbf{x}_m^\top \mathbf{y}_n$ (Eq. (2)),

$$p(a_{mn} | \mathbf{x}_m^\top \mathbf{y}_n, \sigma^2) = \mathcal{N}(a_{mn} | \mathbf{x}_m^\top \mathbf{y}_n, \sigma^2);$$

$$p(\mathbf{A} | \boldsymbol{\theta}) = \prod_{m,n=1}^{M,N} \mathcal{N}(a_{mn} | (\mathbf{X}\mathbf{Y})_{mn}, \sigma^2) = \prod_{m,n=1}^{M,N} \mathcal{N}(a_{mn} | (\mathbf{X}\mathbf{Y})_{mn}, \tau^{-1}),$$

where $\boldsymbol{\theta} = \{\mathbf{X}, \mathbf{Y}, \sigma^2\}$ denotes all parameters in the model, σ^2 is the variance, $\tau^{-1} = \sigma^2$ is the precision,

Prior. We choose a conjugate prior on the data variance, an inverse-Gamma distribution with shape α_σ and scale β_σ ,

$$p(\sigma^2 | \alpha_\sigma, \beta_\sigma) = \mathcal{G}^{-1}(\sigma^2 | \alpha_\sigma, \beta_\sigma).$$

We assume further that the latent variable y_{kl} 's are drawn from a general-truncated-normal prior,

$$p(y_{kl} | \cdot) = \mathcal{GTN}(y_{kl} | \mu_{kl}, \tau_{kl}^{-1}, a = -1, b = 1) = \frac{\sqrt{\frac{\tau_{kl}}{2\pi}} \exp\{-\frac{\tau_{kl}}{2}(y_{kl} - \mu_{kl})^2\}}{\Phi((b - \mu_{kl}) \cdot \sqrt{\tau_{kl}}) - \Phi((a - \mu_{kl}) \cdot \sqrt{\tau_{kl}})} u(y_{kl} | a, b), \quad (\text{A.1})$$

where $u(x | a, b)$ is a step function with a value of 1 when $a \leq x \leq b$ and 0 otherwise.

Let $\mathbf{r} \in \{0, 1\}^N$ be the state vector with each element indicating the type of the corresponding column. If $r_n = 1$, then \mathbf{a}_n is a basis column; otherwise, \mathbf{a}_n is interpolated using the basis columns plus some error term.

Hyperprior. To further favor flexibility, we choose a convenient joint hyperprior density over the parameters $\{\mu_{kl}, \tau_{kl}\}$ of GTN prior in Eq. (A.1), namely, the GTN-scaled-normal-Gamma (GTNSNG) prior,

$$p(\mu_{kl}, \tau_{kl} | \cdot) = \mathcal{GTNSNG}(\mu_{kl}, \tau_{kl} | \mu_\mu, (\tau_\mu)^{-1}, \alpha_t, \beta_t)$$

$$= \{\Phi((b - \mu_\mu) \cdot \sqrt{\tau_\mu}) - \Phi((a - \mu_\mu) \cdot \sqrt{\tau_\mu})\} \cdot \mathcal{N}(\mu_{kl} | \mu_\mu, (\tau_\mu)^{-1}) \cdot \mathcal{G}(\tau_{kl} | a_t, b_t). \quad (\text{A.2})$$

This prior can decouple parameters μ_{kl}, τ_{kl} , and the posterior conditional densities of them are normal and Gamma respectively due to this convenient scale.

Posterior. Following the graphical representation of the GBT (or the GBTN) model in Figure 2, the conditional density of y_{kl}^W can be derived,

$$p(y_{kl} | \mathbf{A}, \mathbf{X}, \mathbf{Y}_{-kl}, \mu_{kl}, \tau_{kl}, \sigma^2) \propto p(\mathbf{A} | \mathbf{X}, \mathbf{Y}, \sigma^2) \cdot p(y_{kl} | \mu_{kl}, \tau_{kl})$$

$$= \prod_{i,j=1}^{M,N} \mathcal{N}(a_{ij} | \mathbf{x}_i^\top \mathbf{y}_j, \sigma^2) \times \mathcal{GTN}(y_{kl} | \mu_{kl}, (\tau_{kl})^{-1}, a = -1, b = 1)$$

$$\propto \exp\left\{-\frac{1}{2\sigma^2} \sum_{i,j=1}^{M,N} (a_{ij} - \mathbf{x}_i^\top \mathbf{y}_j)^2\right\} \exp\left\{-\frac{\tau_{kl}}{2}(y_{kl} - \mu_{kl})^2\right\} u(y_{kl} | a, b)$$

$$\propto \exp\left\{-\frac{1}{2\sigma^2} \sum_i^M (a_{il} - \mathbf{x}_i^\top \mathbf{y}_l)^2\right\} \exp\left\{-\frac{\tau}{2}(y_{kl} - \mu_{kl})^2\right\} u(y_{kl} | a, b)$$

$$\propto \exp\left\{-\frac{1}{2\sigma^2} \sum_i^M \left(x_{ik}^2 y_{kl}^2 + 2x_{ik} y_{kl} \left(\sum_{j \neq k}^N x_{ij} y_{jl} - a_{il}\right)\right)\right\} \exp\left\{-\frac{\tau_{kl}}{2}(y_{kl} - \mu_{kl})^2\right\} u(y_{kl} | a, b)$$

$$\propto \exp\left\{-y_{kl}^2 \left(\frac{\sum_i^M x_{ik}^2}{2\sigma^2} + \frac{\tau_{kl}}{2}\right) + y_{kl} \underbrace{\left(\frac{1}{\sigma^2} \sum_i^M x_{ik} (a_{il} - \sum_{j \neq k}^N x_{ij} y_{jl}) + \tau_{kl} \mu_{kl}\right)}_{\tilde{\mu}}\right\} u(y_{kl} | a, b)$$

$$\propto \mathcal{N}(y_{kl} | \tilde{\mu}, (\tilde{\tau})^{-1}) u(y_{kl} | a, b) \propto \mathcal{GTN}(y_{kl} | \tilde{\mu}, (\tilde{\tau})^{-1}, a = -1, b = 1),$$

where again, for simplicity, we assume the rows of \mathbf{X} are denoted by \mathbf{x}_i 's and columns of \mathbf{Y} are denoted by \mathbf{y}_j 's, $\tilde{\tau} = \frac{\sum_i^M x_{ik}^2}{\sigma^2} + \tau_{kl}$ is the posterior ‘‘parent precision’’ of the general-truncated-normal distribution, and

$$\tilde{\mu} = \left(\frac{1}{\sigma^2} \sum_i^M x_{ik} (a_{il} - \sum_{j \neq k}^N x_{ij} y_{jl}) + \tau_{kl} \mu_{kl} \right) / \tilde{\tau}$$

is the posterior ‘‘parent mean’’ of the general-truncated-normal distribution.

Given the state vector $\mathbf{r} = [r_1, r_2, \dots, r_N]^\top \in \mathbb{R}^N$, the relation between \mathbf{r} and the index sets J is simple; $J = J(\mathbf{r}) = \{n | r_n = 1\}_{n=1}^N$ and $I = I(\mathbf{r}) = \{n | r_n = 0\}_{n=1}^N$. A new value of state vector \mathbf{r} is to select one index j from index sets J and another index i from index sets I (we note that $r_j = 1$ and $r_i = 0$ for the old state vector \mathbf{r}) such that

$$o_j = \frac{p(r_j = 0, r_i = 1 | \mathbf{A}, \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji})}{p(r_j = 1, r_i = 0 | \mathbf{A}, \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji})} = \frac{p(r_j = 0, r_i = 1)}{p(r_j = 1, r_i = 0)} \times \frac{p(\mathbf{A} | \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji}, r_j = 0, r_i = 1)}{p(\mathbf{A} | \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji}, r_j = 1, r_i = 0)},$$

where \mathbf{r}_{-ji} denotes all elements of \mathbf{r} except j -th and i -th entries. Trivially, we can set $p(r_j = 0, r_i = 1) = p(r_j = 1, r_i = 0)$. Then the full conditionally probability of $p(r_j = 0, r_i = 1 | \mathbf{A}, \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji})$ can be calculated by

$$p(r_j = 0, r_i = 1 | \mathbf{A}, \sigma^2, \mathbf{Y}, \mathbf{r}_{-ji}) = \frac{o_j}{1 + o_j}.$$

Finally, the conditional density of σ^2 is an inverse-Gamma distribution by conjugacy,

$$p(\sigma^2 | \mathbf{X}, \mathbf{Y}, \mathbf{A}) = \mathcal{G}^{-1}(\sigma^2 | \tilde{\alpha}_\sigma, \tilde{\beta}_\sigma),$$

where $\tilde{\alpha}_\sigma = \frac{MN}{2} + \alpha_\sigma$, $\tilde{\beta}_\sigma = \frac{1}{2} \sum_{i,j=1}^{M,N} (a_{ij} - \mathbf{x}_i^\top \mathbf{y}_j)^2 + \beta_\sigma$.

Extra update for GBTN model. Following the graphical representation of the GBTN model in Figure 2, the conditional density of μ_{kl} can be derived,

$$\begin{aligned} & p(\mu_{kl} | \tau_{kl}, \mu_\mu, \tau_\mu, a_t, b_t, y_{kl}) \propto \mathcal{GTN}(y_{kl} | \mu_{kl}, (\tau_{kl})^{-1}, a = -1, b = 1) \cdot \mathcal{GTNSNG}(\mu_{kl}, \tau_{kl} | \mu_\mu, (\tau_\mu)^{-1}, \alpha_t, \beta_t) \\ & \propto \mathcal{GTN}(y_{kl} | \mu_{kl}, (\tau_{kl})^{-1}, a = -1, b = 1) \cdot \{ \Phi((b - \mu_\mu) \cdot \sqrt{\tau_\mu}) - \Phi((a - \mu_\mu) \cdot \sqrt{\tau_\mu}) \} \cdot \mathcal{N}(\mu_{kl} | \mu_\mu, (\tau_\mu)^{-1}) \cdot \mathcal{G}(\tau_{kl} | a_t, b_t) \\ & \propto \sqrt{\tau_{kl}} \cdot \exp \left\{ -\frac{\tau_{kl}}{2} (y_{kl} - \mu_{kl})^2 \right\} \cdot \exp \left\{ -\frac{\tau_\mu}{2} (\mu_\mu - \mu_{kl})^2 \right\} \\ & \propto \exp \left\{ -\frac{\tau_{kl} + \tau_\mu}{2} \mu_{kl}^2 + \underbrace{\mu_{kl} (\tau_{kl} y_{kl} + \tau_\mu \mu_\mu)}_{\tilde{m} \cdot \tilde{t}} \right\} \propto \mathcal{N}(\mu_{kl} | \tilde{m}, (\tilde{t})^{-1}), \end{aligned}$$

where $\tilde{t} = \tau_{kl} + \tau_\mu$, and $\tilde{m} = (\tau_{kl} y_{kl} + \tau_\mu \mu_\mu) / \tilde{t}$ are the posterior precision and mean of the normal density. Similarly, the conditional density of τ_{kl} is,

$$\begin{aligned} & p(\tau_{kl} | \mu_{kl}, \mu_\mu, \tau_\mu, a_t, b_t, y_{kl}) \propto \mathcal{GTN}(y_{kl} | \mu_{kl}, (\tau_{kl})^{-1}, a = -1, b = 1) \cdot \mathcal{GTNSNG}(\mu_{kl}, \tau_{kl} | \mu_\mu, (\tau_\mu)^{-1}, \alpha_t, \beta_t) \\ & \propto \mathcal{GTN}(y_{kl} | \mu_{kl}, (\tau_{kl})^{-1}, a = -1, b = 1) \cdot \{ \Phi((b - \mu_\mu) \cdot \sqrt{\tau_\mu}) - \Phi((a - \mu_\mu) \cdot \sqrt{\tau_\mu}) \} \cdot \mathcal{N}(\mu_{kl} | \mu_\mu, (\tau_\mu)^{-1}) \cdot \mathcal{G}(\tau_{kl} | a_t, b_t) \\ & \propto \exp \left\{ -\tau_{kl} \frac{(y_{kl} - \mu_{kl})^2}{2} \right\} \tau_{kl}^{1/2} \tau_{kl}^{a_t-1} \exp \{ -b_t \tau_{kl} \} \propto \exp \left\{ -\tau_{kl} \left[b_t + \frac{(y_{kl} - \mu_{kl})^2}{2} \right] \right\} \cdot \tau_{kl}^{-(a_t+1/2)-1} \\ & \propto \mathcal{G}(\tau_{kl} | \tilde{a}, \tilde{b}), \end{aligned}$$

where $\tilde{a} = a_t + 1/2$ and $\tilde{b} = b_t + \frac{(y_{kl} - \mu_{kl})^2}{2}$ are the posterior parameters of the Gamma density.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the [Creative Commons Attribution license](https://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.