

The Matrix Generalized Inverse Gaussian Distribution: Properties and Applications

Farideh Fazayeli^(✉) and Arindam Banerjee

Department of Computer Science and Engineering,
University of Minnesota, Twin Cities, USA
{farideh, banerjee}@cs.umn.edu

Abstract. While the Matrix Generalized Inverse Gaussian (*MGIG*) distribution arises naturally in some settings as a distribution over symmetric positive semi-definite matrices, certain key properties of the distribution and effective ways of sampling from the distribution have not been carefully studied. In this paper, we show that the *MGIG* is unimodal, and the mode can be obtained by solving an Algebraic Riccati Equation (ARE) equation [7]. Based on the property, we propose an importance sampling method for the *MGIG* where the mode of the proposal distribution matches that of the target. The proposed sampling method is more efficient than existing approaches [32, 33], which use proposal distributions that may have the mode far from the *MGIG*'s mode. Further, we illustrate that the the posterior distribution in latent factor models, such as probabilistic matrix factorization (PMF) [24], when marginalized over one latent factor has the *MGIG* distribution. The characterization leads to a novel Collapsed Monte Carlo (CMC) inference algorithm for such latent factor models. We illustrate that CMC has a lower log loss or perplexity than MCMC, and needs fewer samples.

1 Introduction

Matrix Generalized Inverse Gaussian (*MGIG*) distributions [3, 10] are a family of distributions over the space of symmetric positive definite matrices and has been recently applied as the prior for covariance matrix [20, 32, 33]. *MGIG* is a flexible prior since it contains Wishart, and Inverse Wishart distributions as special cases. We anticipate the usage of *MGIG* as prior for statistical machine learning models to grow with potential applications in Bayesian dimensionality reduction and Bayesian matrix completion (Sect. 4).

Some properties of the *MGIG* distribution and its connection with Wishart distribution has been studied in [10, 26, 27]. However, to best of our knowledge, it is not yet known if the distribution is unimodal and, if it is unimodal, how to obtain the mode of *MGIG*. Besides, it is difficult to analytically calculate mean of the distribution and sample from the *MGIG* distribution. Monte Carlo methods like the importance sampling can in principle be applied to infer the mean of *MGIG* but one needs to design a suitable proposal distribution [21, 23].

There is only one important sampling procedure for estimating the mean of *MGIG* [32, 33]. In this approach, *MGIG* is viewed as a product of the Wishart

and Inverse Wishart distributions and one of them is used as the proposal distribution. However, we illustrate that the mode of the proposal distribution in [32,33] may be far away from the $\mathcal{MGI\mathcal{G}}$'s mode. As a result, the proposal density is small in a region where the $\mathcal{MGI\mathcal{G}}$ density is large yielding to an ineffective sampler and drastically wrong estimate of the mean (Figs. 1 and 2).

In this paper, we first illustrate that the $\mathcal{MGI\mathcal{G}}$ distribution is unimodal where the mode can be obtained by solving an *Algebraic Riccati Equation (ARE)* [7]. This characterization leads to an effective importance sampler for the $\mathcal{MGI\mathcal{G}}$ distribution. More specifically, for estimating the expectation $\mathbb{E}_{X \sim \mathcal{MGI\mathcal{G}}}[g(X)]$, we select a proposal distribution over space of symmetric positive definite matrices like Wishart or Inverse Wishart distribution such that the mode of the proposal matches the mode of the $\mathcal{MGI\mathcal{G}}$. As a result, unlike the current sampler [32,33], by aligning the shape of the proposal and the $\mathcal{MGI\mathcal{G}}$, the density of the proposal gets higher values in the high density regions of the target, yielding to a good approximation of $\mathbb{E}_{X \sim \mathcal{MGI\mathcal{G}}}[g(X)]$.

Further, we discuss a new application of the $\mathcal{MGI\mathcal{G}}$ distribution in latent factor models such as probabilistic matrix factorization (PMF) [24] or Bayesian PCA (BPCA) [4]. In these settings, the given matrix $X \in \mathbb{R}^{N \times M}$ is approximated by a low-rank matrix $\hat{X} = UV^T$ where $U \in \mathbb{R}^{N \times D}$ and $V \in \mathbb{R}^{M \times D}$ with Gaussian priors over the latent matrices U and V . We show that after analytically marginalizing one of the latent matrices in PMF (or BPCA), the posterior over the other matrix has the $\mathcal{MGI\mathcal{G}}$ distribution. This illustration yields to a novel Collapsed Monte Carlo (CMC) inference algorithm for PMF. In particular, we marginalize one of the latent matrices, say V , and propose a direct Monte Carlo sampling from the posterior of the other matrix, say U . Through extensive experimental analysis on synthetic, SNP, gene expression, and MovieLens datasets, we show that CMC has lower log loss or perplexity with fewer samples than Markov Chain Monte Carlo (MCMC) inference approach for PMF [25].

The rest of the paper is organized as follows. In Sect. 2, we cover background materials. In Sect. 3, we show that $\mathcal{MGI\mathcal{G}}$ is unimodal and give a novel importance sampler for $\mathcal{MGI\mathcal{G}}$. We provide the connection of $\mathcal{MGI\mathcal{G}}$ with PMF in Sect. 4, present the results in Sect. 5, and conclude in Sect. 6.

2 Background and Preliminary

In this section, we provide some background on the relevant topics and tools that will be used in our analysis. We start by an introduction to importance sampling, $\mathcal{MGI\mathcal{G}}$ distribution, followed by a brief overview of the ARE.

Notations: Let \mathbb{S}_{++}^N and \mathbb{S}_+^N denote the space of symmetric ($N \times N$) positive definite and positive semi-definite matrix, respectively. Let $|\cdot|$ denote the determinant of matrix, $\text{Tr}(\cdot)$ be the matrix trace. A matrix $\Lambda \in \mathbb{S}_{++}^N$ has a Wishart distribution denoted as $\mathcal{W}_N(\Lambda|\Phi, \tau)$ where $\tau > N - 1$ and $\Phi \in \mathbb{S}_{++}^N$ [31]. A matrix $\Lambda \in \mathbb{S}_{++}^N$ has an Inverse Wishart distribution denoted as $\mathcal{IW}_N(\Lambda|\Psi, \alpha)$ where $\alpha > N - 1$ and $\Psi \in \mathbb{S}_{++}^N$ is the scale matrix. We denote $\mathbf{x}_{:m}$ as the m^{th} column of matrix $X \in \mathbb{R}^{N \times M}$ and \mathbf{x}_n as the n^{th} row of X .

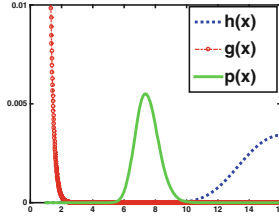


Fig. 1. An illustration of bad proposal distribution in importance sampling. Let $p(x) = h^*(x)g^*(x)/Z_p \propto h(x)g(x)$. Neither $h(x) = h^*(x)/Z_h$ nor $g(x) = g^*(x)/Z_g$ are a good candidate proposal distribution since their modes are far away from the one of $p(x)$.

2.1 Importance Sampling

Consider distribution $p(x) = \frac{1}{Z_p}p^*(x)$ where Z_p is the partition function which plays the role of a normalizing constant. Importance sampling is a general technique for estimating $\mathbb{E}_{x \sim p(x)}[g(x)]$ where sampling from $p(x)$ (the target distribution) is difficult but we can evaluate the value of $p^*(x)$ at any given x [21]. The idea is to draw S samples $\{x_i\}_{i=1}^S$ from a similar but easier distribution denoted by proposal distribution $q(x) = \frac{1}{Z_q}q^*(x)$. Define $w(x_i) = \frac{p^*(x_i)}{q^*(x_i)}$ as the weight of each sample i . Then, we calculate the expected value as follows

$$\mathbb{E}_{x \sim p}[g(x)] = \mathbb{E}_{x \sim q} \left[\frac{g(x)p(x)}{q(x)} \right] \approx \frac{\sum_{i=1}^S g(x_i)w(x_i)}{\sum_{i=1}^S w(x_i)},$$

The efficiency of importance sampling depends on how closely the proposal approximates the target in the shape. One way for monitoring the efficiency of importance sampling is the effective sample size $ESS = \frac{(\sum_{i=1}^S w(x_i))^2}{\sum_{i=1}^S w^2(x_i)}$ [15]. Very small value of ESS indicates a big discrepancy between the proposal and target (for example when the mode of the proposal distribution is far away from the target’s mode) leading to a drastically wrong estimate of $\mathbb{E}_{x \sim p}[g(x)]$ [21].

2.2 MGIG Distribution

MGIG distribution was first introduced in [3] as a distribution over the space of symmetric $(N \times N)$ positive definite matrices defined as follows.

Definition 21. A matrix-variate random variable $\Lambda \in \mathbb{S}_{++}^N$ is *MGIG* distributed [3, 10] and is denoted as $\Lambda \sim \text{MGIG}_N(\Psi, \Phi, \nu)$ if the density of Λ is

$$f(\Lambda) = \frac{|\Lambda|^{\nu-(N+1)/2}}{|\frac{\Psi}{2}|^\nu B_\nu(\frac{\Phi \Psi}{2})} \exp\{\text{Tr}(-\frac{1}{2}\Psi\Lambda^{-1} - \frac{1}{2}\Phi\Lambda)\},$$

where $B_\nu(\cdot)$ is the matrix Bessel function [13] defined as

$$B_\nu(\frac{\Phi \Psi}{2}) = |\frac{\Phi}{2}|^{-\nu} \int_{\mathbb{S}_{++}^N} |S|^{-\nu-\frac{N+1}{2}} \exp\{\text{Tr}(-\frac{1}{2}\Psi S^{-1} - \frac{1}{2}\Phi S)\} dS. \quad (1)$$

When $N = 1$, the $\mathcal{MGI\mathcal{G}}$ is the generalized inverse Gaussian distribution $\mathcal{GI\mathcal{G}}$ [14] which is often used as the prior in several domains [6, 12]. If $\Psi = 0$, the $\mathcal{MGI\mathcal{G}}$ distribution reduces to the Wishart, and if $\Phi = 0$, it becomes the Inverse Wishart distribution.

Proposition 1. [32, Proposition 2] *If matrix $\Lambda \sim \mathcal{MGI\mathcal{G}}_N(\Psi, \Phi, \nu)$, then $\Lambda^{-1} \sim \mathcal{MGI\mathcal{G}}_N(\Phi, \Psi, -\nu)$.*

Sampling Mean of $\mathcal{MGI\mathcal{G}}$: The sufficient statistics of $\mathcal{MGI\mathcal{G}}$ are $\log|\Lambda|$, Λ , and Λ^{-1} . It is, however, difficult to analytically calculate the expectations $\mathbb{E}_{\Lambda \sim \mathcal{MGI\mathcal{G}}}[A]$ and $\mathbb{E}_{\Lambda \sim \mathcal{MGI\mathcal{G}}}[\Lambda^{-1}]$. Importance sampling can be applied to approximate those quantities. Note that based on the result of Proposition 1, the importance sampling procedure for estimating mean of $\mathcal{MGI\mathcal{G}}$, i.e., $\mathbb{E}_{\Lambda \sim \mathcal{MGI\mathcal{G}}}[A]$, can also be applied to infer the reciprocal mean i.e. $\mathbb{E}_{\Lambda \sim \mathcal{MGI\mathcal{G}}}[\Lambda^{-1}]$.

An importance sampling procedure proposed in [32, 33], where the $\mathcal{MGI\mathcal{G}}$ is viewed as a product of Inverse Wishart and Wishart distributions and one of the multiplicands is used as the natural choice of the proposal distribution. In particular, in [32, 33], the $\mathcal{MGI\mathcal{G}}$ is viewed as

$$\mathcal{MGI\mathcal{G}}_N(\Lambda|\Psi, \Psi, \nu) \propto \underbrace{e^{\text{Tr}(-\frac{1}{2}\Phi\Lambda)}}_{T_1} \underbrace{\mathcal{IW}_N(\Lambda|\Psi, -2\nu_u)}_{T_2} \propto \underbrace{e^{\text{Tr}(-\frac{1}{2}\Psi\Lambda^{-1})}}_{T_3} \underbrace{\mathcal{W}_N(\Lambda|\Phi, 2\nu_u)}_{T_4}.$$

In [32, 33], authors advocate using T_2 (or T_4) as the proposal distribution which simplify the weight calculation to the evaluation of T_1 (or T_3). However, it is not studied how close T_2 (or T_4) are to the $\mathcal{MGI\mathcal{G}}$ distribution in shape. For example, consider the 1-dimensional $\mathcal{MGI\mathcal{G}}$ distribution

$$\mathcal{MGI\mathcal{G}}_1(\Lambda|35, 10, 10) \propto \underbrace{e^{\text{Tr}(-\frac{35}{2}\Lambda^{-1})}}_{T_3} \underbrace{\mathcal{W}_1(\Lambda|10, 20)}_{T_4}. \tag{2}$$

In [32, 33], $T_4 : \mathcal{W}_1(\Lambda|10, 20)$ is considered as the proposal distribution, but the mode of T_4 is far away from the mode of $\mathcal{MGI\mathcal{G}}_1(\Lambda|35, 10, 10)$ (Fig. 2(a)). As a result, samples drawn from T_4 will be on the tail of the $\mathcal{MGI\mathcal{G}}_1(\Lambda|10, 20)$ distribution, and will end up getting low weights from the $\mathcal{MGI\mathcal{G}}_1(\Lambda|10, 20)$ distribution. Such a sampling procedure will be wasteful, i.e., drawing samples from the tails of the target $\mathcal{MGI\mathcal{G}}_1$ distribution, leading to a very low ESS . Similar behavior is observed with several different choices of parameters for the $\mathcal{MGI\mathcal{G}}$, here we only show three of them in Fig. 2 due to the lack of space.

2.3 Algebraic Riccati Equation

An algebraic Riccati equation (ARE) is

$$A^T X + X A + X R X + Q = 0, \tag{3}$$

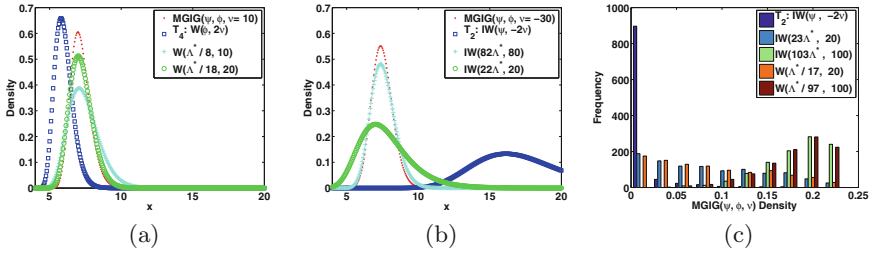


Fig. 2. (a,b) Comparison of different proposal distribution (a) Wishart (W) and (b) Inverse Wishart (IW) for sampling mean of $MGIG_1(\Psi, \Phi, \nu)$ where Λ^* is the mode of $MGIG$. The blue curves are the proposal distribution defined in [32, 33] which can not recover the mode of the $MGIG$ distribution. (c) Density of $MGIG_2(\Psi, \Phi, \nu)$ for 1000 samples generated by each proposal distribution is calculated. More than 90% of samples generated by the previous proposal distribution in [32, 33] ($IW(\psi, -2\nu)$) have zero $MGIG$ density leading to $ESS = 40$. Whereas, the new proposal distribution $IW(23\Lambda^*, 20)$ has the $ESS = 550$ which has a very similar shape to the target $MGIG$. (Color figure online)

where $A \in \mathbb{R}^{N \times N}$, $Q \in \mathbb{S}_+^N$, and $R \in \mathbb{S}_+^N$. We associate a $2N \times 2N$ matrix called the Hamiltonian matrix H with the ARE (3) as $H = \begin{bmatrix} A & R \\ -Q & -A^T \end{bmatrix}$. The ARE (3) has a unique positive definite solution if and only if the associated Hamiltonian matrix H has no imaginary eigenvalues (Section 5.6.3 of [7]).

There have been offered various numerical methods to solve the ARE which can be reviewed in [1]. The key of numerical technique to solve ARE (3) is to convert the problem to a stable invariant subspace problem of the Hamiltonian matrix i.e., finding the invariant subspace corresponding to the eigenvalues of H with negative real parts. The usual ARE solvers such as the Schur vector method [16], SR methods [9], the matrix sign function [2, 11] require in general $O(n^3)$ flops [19]. For special cases, faster algorithms such as [19] can be applied which solves such an ARE with 20k dimensions in seconds. In this paper, we use Matlab ARE solver (*care*) to find the solution of ARE.

3 $MGIG$ Properties

Some properties of the $MGIG$ and its connection with Wishart distribution has been studied in [10, 26, 27]. However, to best of our knowledge, it is not yet known if the distribution is unimodal and how to obtain the mode of $MGIG$. In the following Lemma we show that the $MGIG$ distribution is unimodal.

Lemma 1. Consider the $MGIG$ distribution $MGIG_N(\Lambda|\Psi, \Phi, \nu)$. The mode of $MGIG$ is the solution of the following Algebraic Riccati Equation (ARE)

$$-2\alpha\Lambda + \Lambda\Phi\Lambda - \Psi = 0, \tag{4}$$

where $\alpha = (\nu - \frac{N+1}{2})$. ARE in (4) has a unique positive definite solution, thus the $MGIG$ distribution is a unimodal distribution.

Proof. The log-density of $\mathcal{MGIG}_N(A|\Psi, \Phi, \nu)$ is

$$\log f(A) = \alpha \log |A| - \frac{1}{2} \text{Tr}(\Psi A^{-1} + \Phi A) + C, \tag{5}$$

where $\alpha = (\nu - \frac{N+1}{2})$, and C is a constant which does not depend on A . The mode of \mathcal{MGIG}_N is obtained by setting derivative of (5) to zero as follows

$$\nabla f(A) = -2\alpha A + A\Phi A - \Psi = 0, \tag{6}$$

which is a special case of ARE (3). The associated Hamiltonian matrix for (6) is $H = \begin{bmatrix} -\alpha \mathbb{I}_N & \Phi \\ \Psi & \alpha \mathbb{I}_N \end{bmatrix}$. Recall that ARE has a unique positive definite solution if and only if the associated Hamiltonian matrix H has no imaginary eigenvalues (Section 5.6.3 of [7]). Thus, to show the unimodality of \mathcal{MGIG} , it is enough to show that the corresponding characteristic polynomial $|H - \lambda \mathbb{I}_{2N}| = 0$ has no imaginary solution.

$$\begin{aligned} |H - \lambda \mathbb{I}_{2N}| &= |-(\alpha + \lambda) \mathbb{I}_N| |(\alpha - \lambda) \mathbb{I}_N + (\alpha + \lambda)^{-1} \Psi \Phi| \\ &= |(\alpha - \lambda) \mathbb{I}_N| |-(\alpha + \lambda) \mathbb{I}_N - (\alpha - \lambda)^{-1} \Phi \Psi| = \prod_{i=1}^N \{-(\alpha^2 - \lambda^2) - \tilde{\lambda}_i\} = 0, \end{aligned}$$

which yields to $\lambda^2 = \tilde{\lambda}_i + \alpha^2$ where $\tilde{\lambda}_i$ is the i^{th} eigenvalue of $\Phi \Psi$. Note $\tilde{\lambda}_i > 0$ since Φ and Ψ are positive definite and product of two positive definite matrix has positive eigenvalue. As a result, (7) has no imaginary solution and H does not have any imaginary eigenvalue. As a result, ARE in (6) has a unique positive definite solution. This completes the proof. \square

Importance Sampling for \mathcal{MGIG} : Since \mathcal{MGIG} is a unimodal distribution, we propose an efficient importance sampling procedure for \mathcal{MGIG} by mode matching. We select a proposal distribution over space of positive definite matrices by matching the proposal’s mode to the mode of \mathcal{MGIG} (mode matching) which aligns the proposal and \mathcal{MGIG} shapes. As a result, the proposal $q(x)$ is large in a region where the target distribution \mathcal{MGIG} is large leading to a good approximation of $\mathbb{E}_{A \sim \mathcal{MGIG}}[g(A)]$. An example of such proposal distribution is Inverse Wishart or Wishart distribution.

Let A^* be the mode of $\mathcal{MGIG}_N(A|\Psi, \Phi, \nu)$ which can be found by solving the ARE (6). The mode of $\mathcal{W}_N(A|\Sigma, \rho)$ distribution is $\Sigma^* = (\rho - N - 1)\Sigma$. To match the mode of $\mathcal{W}_N(A|\Sigma, \rho)$ with that of $\mathcal{MGIG}_N(A|\Psi, \Phi, \nu)$, we choose the scale parameter Σ of the Wishart distribution by setting $\Sigma^* = A^*$. In particular,

$$\Sigma^* = A^* = (\rho - N - 1)\Sigma \quad \Rightarrow \quad \Sigma = \frac{A^*}{\rho - N - 1}. \tag{7}$$

Thus, we suggest using $\mathcal{W}_N(\frac{A^*}{\rho - N - 1}, \rho)$ as the proposal distribution. At each iteration, we draw a sample $A_i \sim \mathcal{W}_N(\frac{A^*}{\rho - N - 1}, \rho)$, and calculate $w(A_i)$. More specifically, the density of Wishart distribution is

Algorithm 1. Random Generator of $\mathcal{W}_N(\Lambda|\Sigma, \rho, L)$ [28]

- 1: Note $L^T L = \Sigma$ is the Cholesky factorization of Σ
 - 2: $P_{ii} \sim \sqrt{\chi^2(\rho - (i - 1))}$ for all $i = 1 \cdots N$.
 - 3: $P_{ij} \sim \mathcal{N}(0, 1)$ for $i < j$. ▷ P : upper triangular
 - 4: $R = PL$.
 - 5: Return $\Lambda = R^T R$, $\Lambda^{-1} = R^{-1} R^{-T}$, and $|\Lambda| = \prod_{i=1}^N R_{ii}^2$.
-

$$q(\Lambda) = \frac{q^*(\Lambda)}{Z_q}, \quad \text{where } q^*(\Lambda) = |\Lambda|^{\frac{\nu-N-1}{2}} \exp\left\{-\frac{1}{2} \text{Tr}(\Sigma^{-1} \Lambda)\right\}. \quad (8)$$

Then, the importance weight can be calculated as

$$w(\Lambda_i) = |\Lambda_i|^{\nu-\frac{\rho}{2}} \exp\left\{-\frac{1}{2} \text{Tr}(\Psi \Lambda_i^{-1} + [\Phi - \Sigma^{-1}] \Lambda_i)\right\}. \quad (9)$$

As a result, we approximate $\mathbb{E}_{\Lambda \sim \mathcal{MGI\mathcal{G}}}[g(\Lambda)] \approx \frac{\sum_{i=1}^S w(\Lambda_i) g(\Lambda_i)}{\sum_{j=1}^S w(\Lambda_j)}$. A similar argument holds when the proposal distribution is an Inverse Wishart distribution.

Note that the weight calculation requires to calculate the inverse and determinant of sampled matrix Λ_i . However, as illustrated in Algorithm 1, the random samples generator from \mathcal{W} [28] returns the upper triangular matrix R where $\Lambda = R^T R$. Hence the inverse and determinant of Λ can be calculated efficiently from the inverse and diagonal of the triangular matrix R , respectively. Therefore, the cost of weight calculation is reduced to the cost of solving a linear system and upper triangular matrix production at each iteration.

Figure 2 illustrates that the proposed importance sampling outperforms the one in [32,33] for three examples of $\mathcal{MGI\mathcal{G}}$. In particular, more than 90 % of samples drawn from the proposal distribution T_2 in [32,33] have zero weights leading to $ESS = 40$ (Fig. 2 (c)). Whereas, our proposal distribution achieved $ESS = 550$ leading to a better approximation of the mean of $\mathcal{MGI\mathcal{G}}$. Similar behavior is observed with several different choices of parameters for the $\mathcal{MGI\mathcal{G}}$.

4 Connection of $\mathcal{MGI\mathcal{G}}$ and Bayesian PCA

In this section, we illustrate that the mapping matrix V in Bayesian PCA can be marginalized or ‘collapsed’ yielding a Matrix Generalized Inverse Gaussian ($\mathcal{MGI\mathcal{G}}$) [3,10] posterior distribution over the latent matrix U denoting as the marginalized posterior distribution. Then, we explain the derivation of the marginalized posterior for data with missing values, followed by a collapsed Monte Carlo Inference for PMF.

4.1 PMF, PPCA, and Bayesian PCA

First, we give a review of PMF [24], Probabilistic PCA (PPCA) [29], and Bayesian PCA (BPCA) [4], to illustrate the similarity and differences between the existing ideas and our approach. A related discussion appears in [18]. All these models focus on an (partially) observed data matrix $X \in \mathbb{R}^{N \times M}$. Given latent factors $U \in \mathbb{R}^{N \times D}$ and $V \in \mathbb{R}^{M \times D}$, the rows of X are assumed to be generated according to $\mathbf{x}_{:,m} = U\mathbf{v}_m^T + \epsilon$, where $\epsilon \in \mathbb{R}^N$. The different models vary depending on how they handle distributions or estimates of the latent factors U, V . Without loss of generality, in this paper, we are considering a fat matrix X where $M > N$.

PMF and BPMF: In PMF [24], one assumes independent Gaussian priors for all latent vectors \mathbf{u}_n and \mathbf{v}_m , i.e., $\mathbf{u}_n \sim \mathcal{N}(0, \sigma_u^2 \mathbb{I}), [n]_1^N$ and $\mathbf{v}_m \sim \mathcal{N}(0, \sigma_v^2 \mathbb{I}), [m]_1^M$. Then, one obtains the following posterior over (U, V)

$$p(U, V|X, \theta) = \prod_{n,m} [\mathcal{N}(x_{nm} | \langle \mathbf{u}_n, \mathbf{v}_m \rangle, \sigma^2)]^{\delta_{nm}} \prod_n \mathcal{N}(\mathbf{u}_n | 0, \sigma_u^2 \mathbb{I}) \prod_m \mathcal{N}(\mathbf{v}_m | 0, \sigma_v^2 \mathbb{I}),$$

where $\delta_{nm} = 0$ if x_{nm} is missing and $\delta = \{\sigma^2, \sigma_u^2, \sigma_v^2\}$. PMF obtains point estimates (\hat{U}, \hat{V}) by maximizing the posterior (MAP), based on alternating optimization over U and V [24].

Bayesian PMF (BPMF) [25] considers independent Gaussian priors over latent factors with full covariance matrices, i.e., $\mathbf{u}_n \sim \mathcal{N}(0, \Sigma_u), [n]_1^N$ and $\mathbf{v}_m \sim \mathcal{N}(0, \Sigma_v), [m]_1^M$. Inference is done using Gibbs sampling to approximate the posterior $P(U, V|X)$. At each iteration t , $U^{(t)}$ is sampled from the conditional probability of $p(U|V^{(t-1)}, X)$, followed by sampling V from $p(V|U^{(t)}, X)$.

Probabilistic PCA: In PPCA [29], one assumes independent Gaussian prior over \mathbf{u}_n , i.e., $\mathbf{u}_n \sim \mathcal{N}(0, \sigma_u^2 \mathbb{I})$, but V is treated as a parameter to be estimated. In particular, V is chosen so as to maximize the marginalized likelihood of X

$$p(X|V) = \int_U p(X|U, V)p(U)dU = \prod_{n=1}^N \mathcal{N}(\mathbf{x}_n | 0, \sigma_u^2 VV^T + \sigma^2 \mathbb{I}). \tag{10}$$

Interestingly, as shown in [29], the estimate \hat{V} can be obtained in closed form. For such a fixed \hat{V} , the posterior distribution over $U|X, \hat{V}$ can be obtained as

$$p(U|X, \hat{V}) = \frac{p(X|U, \hat{V})p(U)}{p(X|\hat{V})} = \prod_{n=1}^N \mathcal{N}(\mathbf{u}_n | \Gamma^{-1} \hat{V}^T \mathbf{x}_n, \sigma^{-2} \Gamma), \tag{11}$$

where $\Gamma = \hat{V}^T \hat{V} + \sigma_u^{-2} \sigma^{-2} \mathbb{I}$. Note that the posterior of the latent factor U in (11) depends on both X and \hat{V} . For applications of PPCA in visualization, embedding, and data compression, any point x_n in the data space can be summarized by its posterior mean $E[\mathbf{u}_n | \mathbf{x}_n, \hat{V}]$ and covariance $Cov(\mathbf{u}_n | \hat{V})$ in the latent space.

Bayesian PCA: In Bayesian PCA [4], one assumes independent Gaussian priors for all \mathbf{u}_n and \mathbf{v}_m , i.e., $\mathbf{u}_n \sim \mathcal{N}(0, \sigma_u^2 \mathbb{I})$ and $\mathbf{v}_m \sim \mathcal{N}(0, \sigma_v^2 \mathbb{I}), [m]_1^M$. Bayesian posterior inference by Bayes rule considers $p(U, V|X) = p(X|U, V)p(U)p(V)/p(X)$, which includes the intractable partition function

$$p(X) = \int_U \int_V p(X|U, V)p(U)p(V)dUdV . \tag{12}$$

The literature has considered approximate inference methods, such as variational inference [5], gradient descent optimization [18], MCMC [25], or Laplace approximation [4, 22].

While PPCA and Bayesian PCA were originally considered in the context of embedding and dimensionality reduction, PMF and BPMF have been widely used in the context of matrix completion where the observed matrix X has many missing entries. Nevertheless, as seen from the above exposition, the structure of the models are closely related (also see [17, 18]).

4.2 Closed Form Posterior Distribution in Bayesian PCA

The key challenge in models such as Bayesian PCA or BPMF is that joint marginalization over both latent factors U, V is intractable. PPCA gets around the problem by considering one of the variables, say V , to be a constant. In this section, we show that one can marginalize or ‘collapse’ one of the latent factors, say V , and obtain the marginalized posterior $P(U|X)$ over the other variable denoted. In fact, we obtain the posterior with respect to the covariance structure $\Lambda_u = \beta_u \mathbb{I} + UU^T$, for a suitable constant β_u , which is sufficient to do Bayesian inference on new test points x_{test} . Note that

$$p(U|X) \propto p(U)P(X|U) = p(U) \int_V P(X|U, V)p(V)dV , \tag{13}$$

and, based on the posterior over U , one can obtain the probability on a new point as $p(x_{\text{test}}|X) = \int_U p(x_{\text{test}}|U)p(U|X)dU$.

Next, we show that the posterior over U as in (13), rather the distribution over $\Lambda_u = \beta_u \mathbb{I} + UU^T$, can be derived analytically in *closed form*. The distribution is the Matrix Generalized Inverse Gaussian (\mathcal{MGIG}) distribution.

Similar to (10), marginalizing V gives

$$p(X|U) = \int_V p(X|U, V)p(V)dV = \prod_{m=1}^M \mathcal{N}(\mathbf{x}_{:m} | 0, \sigma_v^2 \Lambda_u) , \tag{14}$$

where $\Lambda_u = \beta_u \mathbb{I} + UU^T$ and $\beta_u = \frac{\sigma_v^2}{\sigma_u^2}$. Then, the marginalized posterior of U is

$$p(U|X) \propto p(X|U) p(U) = |\Lambda_u|^{-M/2} \exp \left\{ \frac{-\text{Tr} \left(\Lambda_u^{-1} \sum_{m=1}^M \mathbf{x}_{:m} \mathbf{x}_{:m}^T \right)}{2\sigma_v^2} \right\} \\ \times \exp \left\{ \frac{-\text{Tr}(\Lambda_u)}{2\sigma_u^2} \right\} \times \exp \left\{ \frac{\text{Tr}(\beta_u \mathbb{I})}{2\sigma_u^2} \right\} \tag{15}$$

$$= |\Lambda_u|^{-M/2} \exp \left\{ \text{Tr} \left(-\frac{1}{2} \Lambda_u^{-1} \Psi_u - \frac{1}{2} \Lambda_u \Phi_u \right) \right\} \\ \sim \mathcal{MGIG}(\Lambda_u | \Psi_u, \Phi_u, \nu_u) , \tag{16}$$

where $\Psi_u = \frac{1}{\sigma_v^2} XX^T$, $\Phi_u = \frac{1}{\sigma_u^2} \mathbb{I}$, and $\nu_u = \frac{N-M+1}{2}$.

Therefore, by marginalizing or collapsing V , the posterior over $\Lambda_u = \beta_v \mathbb{I} + UU^T$ corresponding to the latent matrix U can be characterized exactly with a \mathcal{MGIG} distribution with parameters depending only on X . Note that this is in sharp contrast with (11) for PPCA, where the posterior covariance of \mathbf{u}_n is $\sigma^{-2}\Gamma$ which in turn depends on the point estimate for \hat{V} .

4.3 Posterior Distribution with Missing Data

In this section, we consider the matrix completion setting, when the observed matrix X has missing values. In presence of missing values, the likelihood of the observed sub-vector in any column of X is given as

$$p(\mathbf{x}_{n_m,m} | U, V) = \mathcal{N}(\mathbf{x}_{n_m,m} | \tilde{U}_m \mathbf{v}_m^T, \sigma^2 \mathbb{I}). \tag{17}$$

where n_m is a vector of size \tilde{N}_m containing indices of non-missing entries in column m of X , and \tilde{U}_m is a sub-matrix of U with size of $\tilde{N}_m \times D$ where each row correspond to a non-missing entry in the m^{th} column of X . The marginalized likelihood (14) is $p(X | U) = \prod_{m=1}^M \mathcal{N}(\mathbf{x}_{n_m,m} | 0, \sigma_v^2 \Lambda_{un})$, where $\Lambda_{un} = \beta_v \mathbb{I} + \tilde{U}_n \tilde{U}_n^T$ and $\beta_v = \frac{\sigma^2}{\sigma_v^2}$. The marginalized posterior is given by

$$p(U | X) \propto \exp\left\{-\frac{\text{Tr}(UU^T)}{2\sigma_u^2}\right\} \prod_{m=1}^M |\Lambda_{un}|^{-M/2} \exp\left\{-\frac{1}{2}\mathbf{x}_{n_m,m}^T \Lambda_{un}^{-1} \mathbf{x}_{n_m,m}\right\}.$$

Thus, in presence of missing values, the posterior cannot be factorized as in (15) because each column $\mathbf{x}_{:m}$ contributes to different blocks Λ_{un} of Λ .

We propose to address the missing value issue by gap-filling. In particular, if one can obtain a good estimate of the covariance structure in X , so that $\Psi_u = \frac{1}{\sigma_v^2} X X^T$ in (16) can be approximated well, one can use the \mathcal{MGIG} posterior to do approximate inference. We consider two simple approaches to approximate the covariance structure of X : (i) by zero-padding the missing value matrix X (assuming $E[X] = 0$ or centering the data in practice), and estimating the covariance structure based on the zero-padded matrix [30], and (ii) by using a suitable matrix completion method, such as PMF, to get point estimates of the missing entries in X , and estimating the covariance structure based on the completed matrix. We experiment with both approaches in Sect. 5, and the zero-padded version seems to work quite well.

4.4 Collapsed Monte Carlo Inference for PMF

Given that $\Lambda_u \sim \mathcal{MGIG}_N$, we predict the missing values as follows. Let $\mathbf{x} = [\mathbf{x}^o, \mathbf{x}^*] \sim \mathcal{N}(0, \Lambda)$, where $\mathbf{x}^o \in \mathbb{R}^p$ is the observed partition of $\mathbf{x} \in \mathbb{R}^N$ and

$$\mathbf{x}^* \in \mathbb{R}^{N-p} \text{ is missing. Accordingly, partition } \Lambda \text{ as } \Lambda_u = \begin{pmatrix} p & N-p \\ \Lambda_{oo} & \Lambda_{o*} \\ \Lambda_{*o} & \Lambda_{**} \end{pmatrix} \begin{matrix} p \\ N-p \end{matrix}.$$

Algorithm 2. CMC Inference for PMF

- 1: Let $\Psi_u = \frac{ZZ^T}{\sigma_u^2}$, $\Phi_u = \frac{\mathbb{I}}{\sigma_u^2}$, and $\nu_u = \frac{N-M+1}{2}$.
 - 2: Solve (6) to find mode Λ^* of $\mathcal{MGIG}(\Psi_u, \Phi_u, \nu_u)$.
 - 3: Let $L^T L = \Lambda^*$ be the Cholesky factorization of Λ^* . Let $\tilde{L} = \frac{L}{\sqrt{\rho-M-1}}$.
 - 4: **for** $t = 1 \cdots T$ **do**
 - 5: Let $\Lambda^{(t)} \sim \mathcal{W}_N(\frac{\Lambda^*}{\rho-M-1}, \rho, \tilde{L})$ ▷ Algorithm 1
 - 6: Let $w^t = \frac{\mathcal{MGIG}_N(\Lambda^{(t)}|\Psi_u, \Phi_u, \nu_u)}{\mathcal{W}_N(\Lambda^{(t)}|\frac{\Lambda^*}{\rho-M-1}, \rho, \tilde{L})}$.
 - 7: Let $\mu^t = \Lambda_{*o}^{(t)} \Lambda_{oo}^{(t)-1} \mathbf{x}^o$. Let $\Sigma^t = \Lambda_{**}^{(t)} - \Lambda_{*o}^{(t)} \Lambda_{oo}^{(t)-1} \Lambda_{o*}^{(t)}$.
 - 8: Let $\bar{\mu} = \bar{\mu} + w^t \mu^t$.
Let $\bar{\Sigma} = \bar{\Sigma} + w^t \Sigma^t$.
 - 9: Report the distribution of $\mathbf{x}^* \sim \mathcal{N}(\tilde{\mu}^*, \tilde{\Sigma}^*)$ where $\tilde{\mu}^* = \frac{\bar{\mu}}{\sum_{t=1}^T w^t}$ and $\tilde{\Sigma}^* = \frac{\bar{\Sigma}}{\sum_{t=1}^T w^t}$.
-

Then the conditional probability of \mathbf{x}^* given \mathbf{x}^o and Λ is

$$p(\mathbf{x}^* | \mathbf{x}^o, \Lambda) \sim \mathcal{N}\left(\underbrace{\Lambda_{*o} \Lambda_{oo}^{-1} \mathbf{x}^o}_{\mu^*}, \underbrace{\Lambda_{**} - \Lambda_{*o} \Lambda_{oo}^{-1} \Lambda_{o*}}_{\Sigma^*}\right) \tag{18}$$

where $\mathbf{y} = \Lambda_{*o} \Lambda_{oo}^{-1}$ is the solution of the linear system $\Lambda_{oo} \mathbf{y} = \Lambda_{*o}^T$ and can be calculated efficiently. Since sampling from \mathcal{MGIG} is difficult, we propose to use importance sampling to infer the missing values as

$$p(x_n^* | x_n^o) = \mathbb{E}_{\Lambda \sim \mathcal{MGIG}} [p(x_n^* | x_n^o, \Lambda)] = \mathbb{E}_{\Lambda \sim q} \left[\frac{p(x_n^* | x_n^o, \Lambda) \mathcal{MGIG}_N(\Lambda | \Psi_u, \Phi_u, \nu_u)}{q(\Lambda)} \right],$$

where q is the proposal distribution as discussed above and sampling $\Lambda^{(t)}$ from q yields to the estimate of

$$\tilde{\mu}^* = \frac{\sum_{t=1}^T \Lambda_{*o}^{(t)} \Lambda_{oo}^{(t)-1} \mathbf{x}^o w(\Lambda^{(t)})}{\sum_{t=1}^T w(\Lambda^{(t)})}, \quad \tilde{\Sigma}^* = \frac{\sum_{t=1}^T [\Lambda_{**}^{(t)} - \Lambda_{*o}^{(t)} \Lambda_{oo}^{(t)-1} \Lambda_{o*}^{(t)}] w(\Lambda^{(t)})}{\sum_{t=1}^T w(\Lambda^{(t)})}.$$

Algorithm 2 illustrates the summary of the collapsed Monte Carlo (CMC) inference for predicting the missing values. A practical approximation to avoid the calculations in Lines 7–8 of Algorithm 2 at each iteration is to simply estimate the mean of the posterior $\bar{\Lambda} = \frac{\sum_{t=1}^T \Lambda^{(t)} w^t}{\sum_{t=1}^T w^t}$ with samples drawn from the proposal distribution (line 6), then do the inference based on $\bar{\Lambda}$.

5 Experimental Results

We compared the performance of MCMC and CMC on both log loss and running times. We evaluated the models on 4 datasets: (1) **SNP**: single nucleotide polymorphism (SNP) is important for identifying gene-disease associations where the data usually has 5 to 20% of genotypes missing [8]. We used phased SNP dataset for chromosome 13 of the CEU population¹. We randomly dropped 20%

¹ <http://hapmap.ncbi.nlm.nih.gov/downloads/phasing/>.

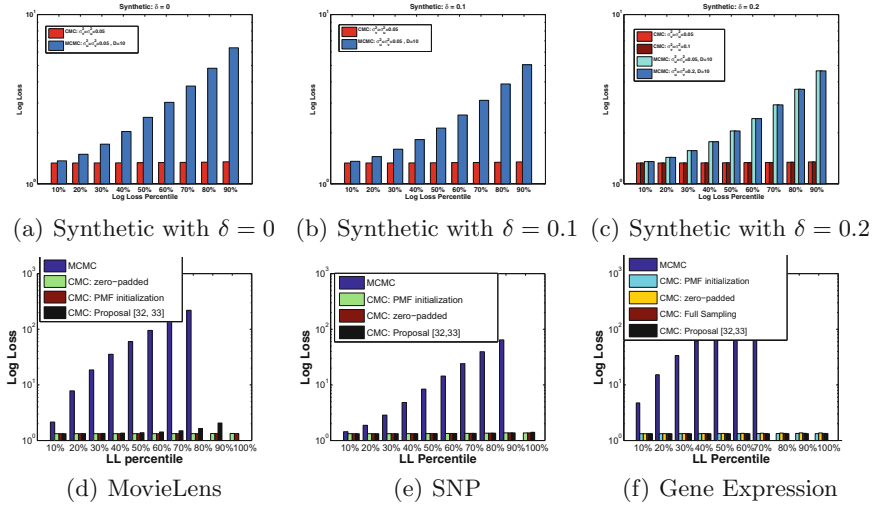


Fig. 3. Log loss (LL) of CMC and MCMC for different log loss percentile on different datasets presented in the log scale (δ denotes the missing proportion). CMC consistently achieves lower LL compared to MCMC. LL of MCMC increases exponentially (linearly in log scale) by adding data points with higher log loss. Proposal in [30,31] achieved infinity LL for MovieLens. Empty bar represents infinity LL (e.g. 90% and 100% percentile in (d))

of the entries. (2) **Gene Expression:** DNA microarrays provides measurement of thousands of genes under a certain experimental condition where suspicious values are usually regarded as missing values. Here we used gene expression dataset for Breast Cancer (BRCA)². We randomly dropped 20% of the entries. (3) **MovieLens:** we used MovieLens³ dataset with 1M rating represented as a fat matrix $X \in \mathbb{R}^{N \times M}$ where $M = 3900$ movies and $N = 6040$ users. (4) **Synthetic:** first the latent matrices U and V are generated by randomly choosing each $\{\mathbf{u}_n\}_{n=1}^N$ and $\{\mathbf{v}_m\}_{m=1}^M$ from $\mathcal{N}(0, \sigma_u^2 \mathbb{I})$ and $\mathcal{N}(0, \sigma_v^2 \mathbb{I})$, respectively. Then, matrix X is built by sampling each x_{nm} from $\mathcal{N}(\langle \mathbf{u}_n, \mathbf{v}_m \rangle, \sigma^2)$. The parameters are set to $N = 100$, $M = 6000$, $\sigma_u^2 = \sigma_v^2 = 0.05$, and $\sigma^2 = 0.01$. We dropped random entries using Bernoulli distributions with $\delta = 0.1, 0.2$.

5.1 Methodology

We compared CMC with MCMC inference for PMF. Gibbs sampling with diagonal covariance prior over the latent matrices is used for MCMC. For the model evaluation, average of log loss (LL) is reported over 5-fold cross-validation. LL measures how well a probabilistic model q predicts the test sample defined as $LL = -\frac{1}{T} \sum_{i=1}^N \sum_{j=1}^M \delta_{ij} \log q(x_{ij})$ where $q(x_{ij})$ is the inferred probability and T

² <http://cancergenome.nih.gov/>.

³ www.movielens.umn.edu.

is the total number of observed values. A better model q assign higher probability $q(x_{ij})$ to observed test data, and have a smaller value of LL.

LL Percentile: For any posterior model $q(x)$, a test data point x_{test} with low $q(x_{\text{test}})$ has large log loss, and high $q(x_{\text{test}})$ has low log loss. To comparatively evaluate the posteriors obtained from MCMC and CMC, we consider their log loss percentile plots. For any posterior, we sort all the test data points in ascending order of their log loss, and plot the mean log loss in 10 percentile batches. More specifically, the first batch corresponds to the top 10 % of data points with the lowest log loss, the second batch corresponds to the top 20 % of data points with the lowest log loss (including the first 10 % percentile), and so on.

5.2 Results

We summarize the results from different aspects:

Log loss: CMC has a small log loss across all percentile batches, whereas log loss of MCMC increases exponentially (linear increase in the log scale) for percentile batches with higher log loss i.e., smaller predicting probability, (Fig. 3). Thus, MCMC assigned extremely low probability to several test points as compared to CMC. Figure 4(a) illustrates that log loss of MCMC continues to decrease with growing sample size up to 2000 samples, implying that MCMC has not yet converged to the equilibrium distribution. Note that log loss of CMC with 200 samples (Fig. 4(b)) is 10 times less than log loss of MCMC with 2000 samples. We also compared the results with the previous proposal [32,33], and observed that for MovieLens the results are worse than our proposed result as they achieved Inf LL on the last batch.

Effective number of samples: For the synthetic, SNP, and gene expression datasets, we generated 10,000 samples using MCMC. The burn-in period is set to 500 with a lag of 10 yielding to 1000 effective samples. For the MovieLens, we generated 5,000 samples using MCMC with the burn-in period of 1000 and a lag of 2 yielding to 2000 effective samples. We initialized the latent matrices

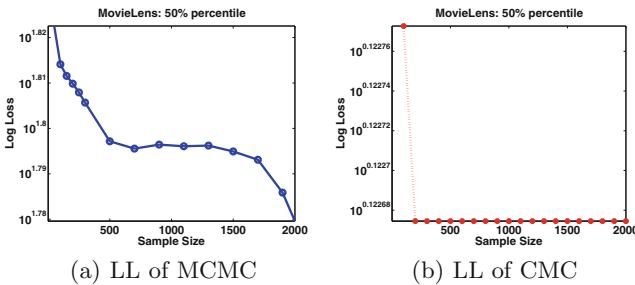


Fig. 4. LL of CMC and MCMC for different sample size of MovieLens data in the log scale. LL of both CMC and MCMC is decreasing by adding more samples. LL of MCMC is in magnitude 10 times more than CMC.

U and V with the factors estimated by PMF, to help the convergence of MCMC. Sample size in CMC procedure is set to 1,000 for all datasets. Note that MCMC alternately sample both latent matrices U and V from a Markov chain and the quality of the posterior improves with increasing number of samples. For the proposed CMC procedure, the bigger matrix V is marginalized and only samples from the smaller U matrix is drawn directly from the true posterior distribution. Hence, CMC has considerably improved sample utilization.

Initialization: As discussed in Sect. 4.3, in order to use the \mathcal{MGTG} posterior for inference, the covariance structure of matrix X should be estimated. Here we evaluate two approaches to approximate the covariance structure of X : (i) by zero-padding the missing value matrix X , and (ii) by computing the point estimates of the missing entries in X with PMF. CMC with zero-padded initialization has a similar log loss behavior as point estimate initialization with PMF (Figs. 3 (d-f)).

Full sampler vs Mean sampler: Figure 3(f) shows the result of the full sampler (Algorithm 2), and the mean sampler (approximating the inference by estimating $\bar{A} = \mathbb{E}_{A \sim \mathcal{MGTG}}[A]$ as discussed in Sect. 4.4) on gene expression data. Since the log losses are similar with both samplers, and the behavior is typical, we presented log loss results on the other datasets only based on the mean sampler, which is around 100 times faster.

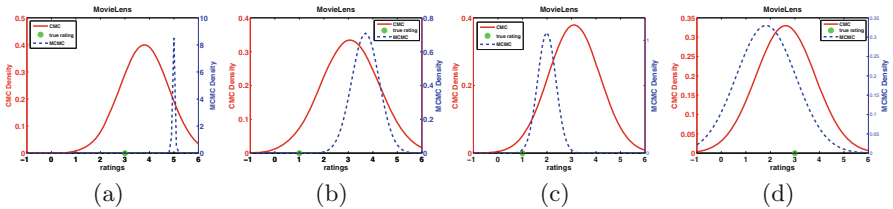


Fig. 5. Density of CMC and MCMC for several data input on MovieLens data with LL of (a) CMC:-1.78, MCMC:-Inf, (b) CMC:-3, MCMC:-17, (c) CMC:-4.2, MCMC:-6.4, (d) CMC:-1.4, MCMC:-2.04. CMC achieves lower LLs compared to MCMC. (Color figure online)

Comparison of inferred posterior distributions: To emphasize the importance of choosing the right measure for comparison, e.g., log loss vs RMSE, we illustrate the inferred posterior distributions over several missing entries/ratings in MovieLens obtained from MCMC and CMC in Fig. 5. Note that the scales for CMC (red) and MCMC (blue) are different. Overall, the posterior from CMC tends to be more conservative (not highly peaked), and obtains lower log loss across a range of test points. Interestingly, as shown in Fig. 5(a), MCMC can make mistakes with high confidence, i.e., predicts 5 stars with a peaked posterior whereas the true rating is 3 stars. Such troublesome behavior is correctly assessed with log loss, but not by RMSE since it does not consider the confidence in the prediction. As shown in Fig. 5(d), for some test points, both MCMC and

Table 1. Time Comparison of CMC and MCMC on different datasets. At each step of MCMC, rows of U and V can be sampled in parallel denoted by MCMC parallel. The running time is reported over 1000 steps for both methods. Note that the effective number of samples of MCMC is less than 1000 and more steps is required to obtain enough samples. The number of iterations for convergence of CMC is less than 1000.

Dataset	Size	MCMC(200 samples)		CMC(1000 samples)	
		Serial	Parallel	Serial	Parallel
Synthetic	100 × 6,000	728s	404s	6s	4s
SNP	120 × 104,868	12,862s	5,859s	75s	22s
Gene Expression	591 × 17,814	3,478s	2,278s	140s	90s
MovieLens	3,233 × 6,040	2,350s	2,100s	5,387s	2,058s

CMC inferred similar posterior distributions with a bias difference where the mean of CMC is closer to the true value.

Time Comparison: We have compared running time in both serial and in parallel over 1000 steps yielding to 200 and 1000 samples for MCMC and CMC, respectively. We implement the algorithms in Matlab. The computation time is estimated on a PC with a 3.40 GHz Quad core CPU and 16.0 G memory. The average run time results are reported in Table 1. For Synthetic, SNP, and gene expression datasets, MCMC converges very slowly. For MovieLens dataset, the running time of both are very close but note that MCMC requires more number of samples for convergence than CMC (Fig. 4).

6 Conclusion

We studied the $MGI\mathcal{G}$ distribution and provided certain key properties with a novel sampling technique from the distribution and its connection with the latent factor models such as PMF or BPCA. With showing that the $MGI\mathcal{G}$ distribution is unimodal and the mode can be obtained by solving an ARE, we proposed a new importance sampling approach to infer the mean of $MGI\mathcal{G}$. The new sampler, unlike the existing sampler [32, 33], chooses the proposal distribution to have the same mode as the $MGI\mathcal{G}$. This characterization leads to a far more effective sampler than [32, 33] since the new sampler align the shape of the proposal to the target distribution. Although, the $MGI\mathcal{G}$ distribution has been recently applied to Bayesian models as the prior for the covariance matrix, here, we introduced a novel application of the $MGI\mathcal{G}$ in PMF or BPCA. We showed that the posterior distribution in PMF or BPCA has the $MGI\mathcal{G}$ distribution. This illustration, yields to a new CMC inference algorithm for PMF.

Acknowledgements. The research was supported by NSF grants IIS-1447566, IIS-1447574, IIS-1422557, CCF-1451986, CNS-1314560, IIS-0953274, IIS-1029711, NASA grant NNX12AQ39A, and gifts from Adobe, IBM, and Yahoo. F.F. acknowledges the support of DDF (2015–2016) from the University of Minnesota.

References

1. Anderson, B., Moore, J.: Optimal control: linear quadratic methods (2007)
2. Bai, Z., Demmel, J.: Using the matrix sign function to compute invariant subspaces. *SIAM J. Matrix Anal. Appl.* **19**(1), 205–225 (1998)
3. Barndorff-Nielsen, O., Blæsild, P., et al.: Exponential transformation models. *Proc. Roy. Soc. London Ser. A* **379**(1776), 41–65 (1982)
4. Bishop, C.M.: Bayesian PCA. *NIPS* **11**, 382–388 (1999)
5. Bishop, C.M.: Variational principal components. In: *ICANN* (1999)
6. Blei, D., Cook, P., Hoffman, M.: Bayesian nonparametric matrix factorization for recorded music. In: *ICML*, pp. 439–446 (2010)
7. Boyd, S., Barratt, C.: Linear controller design: limits of performance (1991)
8. Brevern, A.D., Hazout, S., Malpertuy, A.: Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering. *BMC Bioinform.* **5**(1), 114 (2004)
9. Bunse-Gerstner, A., Mehrmann, V.: A symplectic QR like algorithm for the solution of the real algebraic Riccati equation. *IEEE Trans. Autom. Control* **31**, 1104–1113 (1986)
10. Butler, R.W.: Generalized inverse Gaussian distributions and their Wishart connections. *Scand. J. Statist.* **25**(1), 69–75 (1998)
11. Byers, R.: Solving the algebraic Riccati equation with the matrix sign function. *Linear Algebra Appl.* **85**, 267–279 (1987)
12. Eberlein, E., Keller, U.: Hyperbolic distributions in finance. *Bernoulli* **1**, 281–299 (1995)
13. Herz, C.: Bessel functions of matrix argument. *Ann. Math.* **23**, 77–87 (1955)
14. Jørgensen, B.: Statistical properties of the generalized inverse Gaussian distribution
15. Kong, A., Liu, J., Wong, W.: Sequential imputations and bayesian missing data problems. *JASA* **89**(425), 278–288 (1994)
16. Laub, A.: A Schur method for solving algebraic Riccati equations. *IEEE Trans. Autom. Control* **24**(6), 913–921 (1979)
17. Lawrence, N.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *JMLR* **6**, 1783–1816 (2005)
18. Lawrence, N., Urtasun, R.: Non-linear matrix factorization with gaussian processes. In: *ICML* (2009)
19. Li, T., Chu, E., et al.: Solving large-scale continuous-time algebraic Riccati equations by doubling. *J. Comput. Appl. Math.* **237**(1), 373–383 (2013)
20. Li, Y., Yang, M., Qi, Z., Zhang, Z.: Bayesian multi-task relationship learning with link structure. In: *ICDM* (2013)
21. MacKay, D.: Information theory, inference, and learning algorithms (2003)
22. Minka, T.P.: Automatic choice of dimensionality for PCA. In: *NIPS* (2000)
23. Owen, A.: Monte Carlo theory, methods and examples (2013)
24. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: *NIPS* (2007)
25. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using markov chain monte carlo. In: *ICML* (2008)
26. Seshadri, V.: Some properties of the matrix generalized inverse Gaussian distribution. *Stat. Methods Pract.* **69**, 47–56 (2003)
27. Seshadri, V., Wesolowski, J.: More on connections between Wishart and matrix GIG distributions. *Metrika* **68**(2), 219–232 (2008)
28. Smith, W., Hocking, R.: Algorithm as 53: Wishart variate generator. *Appl. Statist.* **21**, 341–345 (1972)

29. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. *J. Royal Statist. Soc. Seri. B* **61**(3), 611–622 (1999)
30. Wang, H., Fazayeli, F., et al.: Gaussian copula precision estimation with missing values. In: *AISTATS* (2014)
31. Wishart, J.: The generalised product moment distribution in samples from a normal multivariate population. *Biometrika* **20A**, 32–52 (1928)
32. Yang, M., Li, Y., Zhang, Z.: Multi-task learning with Gaussian matrix generalized inverse Gaussian model. In: *ICML* (2013)
33. Yoshii, K., Tomioka, R.: Infinite positive semidefinite tensor factorization for source separation of mixture signals. In: *ICML* (2013)